

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE.

**MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA
RECHERCHE SCIENTIFIQUE.
UNIVERSITE MENTOURI, CONSTANTINE.
FACULTE DES SCIENCES DE L'INGENIEUR.
DEPARTEMENT D'INFORMATIQUE.**

N° d'ordre:

Série:

THESE

Présentée pour obtenir le grade de

DOCTORAT EN SCIENCES EN INFORMATIQUE

Intégration d'ontologies dans le cadre du web sémantique : une détection des relations sémantiques basée sur le RÀPC

Présentée par:

Kolli Manel

Dirigée par:

Pr. Boufaida Zizette. Université Mentouri Constantine.

Devant le jury

Président : Pr. Mahmoud Boufaida, professeur à l'université Mentouri, Constantine.
Rapporteur : Pr. Zizette Boufaida, professeur, Université Mentouri Constantine.
Examineurs : Pr. Nadir Farah, professeur à l'université Badji Mokhtar, Annaba.
Dr. Faiza Belala, MCA à l'université Mentouri, Constantine.
Dr. Hassina Seridi, MCA à l'université Badji Mokhtar, Annaba.

*Merci mon Dieu de m'avoir donné la force, la patience et la
volonté d'arriver au terme de travail.*

*A Mes parents
Mon mari
Mes enfants: May, Maha et Mouhamed*

Remerciements

Je tiens à exprimer ma gratitude envers mon encadreur, Professeur Zizette BOUFAIDA pour la qualité de ses conseils, sa disponibilité ainsi que le degré de responsabilisation de son encadrement qui m'a permis de progresser ce travail. Je la remercie d'avoir consacré beaucoup de son temps pour les nombreuses relectures de mon document et les articles publiés.

Mes sincères remerciements aux membres du jury qui ont accepté de juger mon travail et dont les remarques et les suggestions auront le plus grand impact pour l'amélioration et le raffinement de ce travail. Ainsi, je remercie :

Pr. Mahmoud Boufaida, professeur à l'université Mentouri, Constantine.

Pr. Nadir Farah, professeur à l'université Badji Mokhtar, Annaba.

Dr. Faiza Belala, MCA à l'université Mentouri, Constantine.

Dr. Hassina Seridi, MCA à l'université Badji Mokhtar, Annaba.

Je salue également tous les membres du laboratoire LIRE, spécialement de l'équipe SI & BC.

Enfin, je remercie les membres de ma famille et mes amis pour leur soutien et leur compréhension.

Sommaire

Introduction générale	I
1. Contexte de la thèse.....	i
2. Problématique	ii
3. Objectifs	iv
4. Contributions de la thèse	v
5. Organisation de la thèse	vi
Généralités sur les ontologies et le Web sémantique	1
1. Introduction	2
2. Le Web sémantique	2
3. Présentation de la notion d'ontologie	3
3.1 Origine de l'ontologie	3
3.2 Définitions	4
3.3 Composantes d'une ontologie	5
3.3.1. Les concepts	5
a. Les propriétés portant sur un concept	6
b. Les propriétés portant sur deux concepts	6
3.3.2. Les relations	7
a. Les propriétés intrinsèques à une relation	7
b. Les propriétés liant deux relations	7
c. Les propriétés liant une relation et des concepts	7
3.3.3. Les fonctions	8
3.3.4. Les axiomes (ou Règles)	8
3.3.5. Les instances (ou individus)	8
3.4 Rôles de l'ontologie	9
3.5 Classification des ontologies	10
3.5.1 Dimensions de classification des ontologies	10
a. Typologie selon l'objet de conceptualisation	10
b. Typologie selon le niveau de détail de l'ontologie	13
c. Typologie selon le niveau de complétude	13
d. Typologie selon le niveau du formalisme	14
3.6 Formalismes de représentation des ontologies	15
3.6.1 Les frames	15

3.6.2 Graphe conceptuel	16
- Le Modèle des Graphes Conceptuels	16
3.6.3 Logiques de description	18
a. Les constructeurs des LDs	19
- Syntaxe de la logique de description	19
- Sémantique	21
3.7 Le cycle de vie des ontologies	23
4. Langages pour le web sémantique	24
4.1 RDF et RDF(S) fondement du Web Sémantique	25
4.1.1 Le noyau du modèle	25
4.2 RDFS	26
4.3 DAML + OIL	28
4.4 OWL	28
4.4.1 OWL et les autres langages	29
4.1.2 Sous langages de OWL.....	29
4.4.3 Syntaxe et sémantique du langage OWL	30
5. Le Web sémantique et le Web 2.....	32
5.1 Apports du Web 2 pour le Web Sémantique	32
5.2 Apports du Web Sémantique pour le Web 2	33
6. Conclusion	34
Intégration des ontologies	36
1. Introduction	37
2. Intégration des ontologies	37
2.1 Hétérogénéité des ontologies	39
2.1.1 Niveau syntaxique	39
2.1.2. Niveau terminologique	39
2.1.3. Niveau conceptuel	40
2.1.4 Niveau pragmatique	40
2.2 Différents types d'intégration des ontologies	40
2.2.1 La mise en correspondance (mapping) des ontologies	42
a. Formalisation de la notion de mapping	42
b. Mesures de similarité	45
c. Méthodologies du mapping	45
- MAFRA	45
- IF-Map	47
- RDFT	48
- C-OWL	50
- OntoMap	51
2.2.2 L'alignement des ontologies	51
- Anchor-PROMPT	52
- GLUE	54
- S-Match	54
- QOM	55
- ASCO	55
2.2.3 La fusion (merging) des ontologies	55
- PROMPT	56
- CHIMAERA	58
- FCA-Merge	58
- OntoMerge	58
3. Discussion	59

4. Conclusion	60
Proposition d'un système de détection des relations sémantiques dans les ontologies basée sur le RÀPC	61
1. Introduction	62
2. Le raisonnement à partir de cas (RÀPC)	63
3. Détection des relations sémantiques entre les ontologies en utilisant le raisonnement à partir de cas (RÀPC)	64
3.1. La phase de conception	64
3.1.1 Distance sémantique	68
- Calcul de la distance sémantique	69
3.2. Phase de production	73
3.2.1. Elaboration	73
3.2.2. Remémoration	74
3.2.3. La réutilisation	76
a. Préparation des données	76
b. Fouille de données	78
- Appariement des ontologies	78
- Composition	79
- Agrégation de similarité	80
3.2.4. La révision	82
3.2.5. La mémorisation	83
4. Conclusion	84
Aspects techniques pour l'implantation et étude de cas	85
1. Introduction	86
2. Aspects techniques pour l'implantation	86
2.1. Choix de l'éditeur d'ontologies	86
2.2. Langage de programmation et frameworks spécialisées	87
3. Etude de cas	89
3.1. Présentation de l'étude de cas	89
4. Prototype	93
5. Test et vérification	98
6. Conclusion	101
Conclusion générale	102
1. Contributions	102
2. Perspectives	104
Bibliographie	105

INTRODUCTION GENERALE

1. Contexte de la thèse

La quantité d'information disponible sur le web croit sans cesse. En effet, le web classique nécessite de nouveaux outils pour la gestion, l'accès et la manipulation de ces informations. L'évolution du web actuel au web sémantique permet à des agents logiciels d'exploiter, de composer et de raisonner sur les contenus des ressources web via les mécanismes de l'intelligence artificielle.

Le Web sémantique a pour objectif de transformer le World Wide Web actuel, entièrement tourné vers la présentation des documents, vers un Web dont le contenu serait compréhensible par les machines. La vision s'appuie sur l'utilisation d'ontologies, qui sont des conceptualisations communes et partagées entre différents agents, et d'un langage de description, permettant d'exprimer à la fois les définitions des concepts et des rôles dans les ontologies et d'exprimer des annotations utilisant le vocabulaire de ces ontologies. Ce langage et ces ontologies permettraient aux agents informatiques de comprendre les diverses annotations et de communiquer entre eux, en effectuant des raisonnements sur les concepts. Les problématiques de la représentation de concepts en Intelligence Artificielle sont donc fondamentales pour le Web sémantique (Delteil 2002).

Il est clair que les recherches sur les ontologies sont essentielles pour la réalisation du Web sémantique. En effet, d'une part, une fois construite et acceptée par une communauté particulière, une ontologie doit traduire un certain *consensus explicite* et un certain niveau de *partage* qui sont essentiels pour permettre l'exploitation des ressources du Web par différentes applications ou agents logiciels.

D'autre part, la *formalisation*, autre facette des ontologies, est nécessaire pour que ces outils puissent être munis de capacités de raisonnement permettant de décharger les différents utilisateurs d'une partie de leur tâche d'exploitation et de combinaison des ressources du Web.

Du point de vue des ontologies, seront cruciales pour le Web sémantique les méthodes et les outils contribuant à (Delteil 2002):

- Construire les ontologies, que ce soit à partir de sources primaires, particulièrement les corpus textuels, ou en recherchant une certaine réutilisabilité. La construction d'ontologies à partir de l'analyse de corpus textuels est un domaine en forte évolution où un certain nombre de méthodologies et d'outils sont testés par une communauté très active. La

question de la réutilisabilité qui a suscité de longs débats dans la communauté Ingénierie des connaissances a permis de progresser vers la recherche d'une certaine généricité mais reste un enjeu majeur pour le Web sémantique.

- Gérer l'accès aux ontologies, leur évolution, avec gestion des versions, et leur intégration. Les ontologies sont souvent riches de plusieurs milliers de concepts et ne restent alors directement appréhendables que par leur concepteur. Leur accès par des utilisateurs, même professionnels, nécessite de gérer le lien entre les concepts des ontologies et les termes du langage naturel, que ce soit pour une simple compréhension ou pour l'indexation et la construction de requêtes destinées à des tâches de recherche d'information. Les solutions mises en œuvre à ce jour passent par des méthodologies séparant explicitement les termes et les concepts d'un domaine et des outils de visualisation et de navigation recherchant des proximités conceptuelles dans les termes d'un domaine et permettant d'appréhender intuitivement la complexité de ce domaine.
- Assurer l'interopérabilité des ontologies en gérant les hétérogénéités de représentation et les hétérogénéités sémantiques. Ces dernières sont les plus dures à gérer (Laublet et al. 2002).

L'ontologie est un facteur clé qui facilite l'interopérabilité dans le web sémantique (ACM 2009). Les ontologies sont le noyau du Web sémantique parce qu'elles permettent aux applications de communiquer en utilisant des termes partagés. L'ontologie facilite donc la communication en fournissant des notions précises qui peuvent être employées pour composer et échanger des messages (questions, réponses, etc.).

Cependant, il n'existe pas d'ontologie universelle partagée, adoptée par tous les utilisateurs d'un domaine donné. Les problématiques et les tentatives d'amélioration de l'interopérabilité du système comptent donc sur la réconciliation des différentes ontologies utilisées dans un domaine par les différents systèmes. Cette réconciliation est souvent réalisée par l'intégration manuelle ou semi-automatisée des ontologies. Elle consiste à identifier les liens de correspondance entre les ontologies, on parle alors de mapping d'ontologies.

2. Problématique

Le problème d'intégration des ontologies (connu aussi comme médiation des ontologies, correspondance d'ontologies, ... etc.) joue un rôle central dans le développement des systèmes à base de connaissances. Les nouvelles technologies augmentent l'utilisation des ontologies dans le Web sémantique pour la représentation des connaissances. Cette tendance a conduit au développement de nouvelles ontologies, d'où le nombre élevé des ontologies disponibles sur le Web.

Le Web sémantique exploite les ontologies dans la représentation des connaissances (Charlet et al. 2005). Partant du fait que plusieurs connaissances peuvent prendre des représentations différentes, nous trouvons de nos jours plusieurs ontologies de domaine pour un même champ d'application. Il est alors nécessaire de disposer d'outils permettant de faire le lien entre les connaissances exprimées dans chacune des ontologies.

La mise en correspondance (mapping) des ontologies permet les échanges de manière sémantique des connaissances. Dans ce cas la mise en correspondance est essentielle dans la réutilisation des ontologies dans des systèmes permettant leur manipulation. Cette manipulation est réalisée par l'intégration de ces ontologies.

Ce vaste thème de recherche a donné lieu à de très nombreux travaux dont La majorité se concentre sur l'alignement des ontologies. Dans ce type d'intégration, la mise en correspondance permet d'établir un lien entre deux sources d'informations pour échanger les informations d'une source à l'autre. Ainsi, l'alignement d'ontologies permet aussi de réconcilier, d'un point de vue sémantique, les opinions de plusieurs experts (Bach et al. 2004).

Les autres travaux se concentrent sur la fusion dont la mise en correspondance permet de construire une ontologie ou un schéma global unifiant toutes les informations des sources (Noy et Mussem 2002).

Les techniques d'intégration jouent un rôle crucial dans la construction d'un lien sémantique entre les ontologies d'un même domaine. Quelques approches d'intégration (Aleksovski et al, 2006, Stuckensschmidt et al, 2004, Hage et al. 2005) considèrent que l'utilisation d'une connaissance sur le domaine est une manière assurant la correspondance sémantique entre la dissimilarité syntaxique des ontologies. L'obtention de la bonne connaissance sur le domaine est primordiale.

D'autres approches n'exploitent pas une connaissance sur le domaine et ne réalisent pas un modèle sémantique formel pour l'alignement des structures produites. Dans ce cas, la structure obtenue est difficile à exploiter, *e.g.*, pour répondre aux requêtes interrogeant les ontologies (Lopez et al. 2006).

En outre, les approches courantes d'intégration des ontologies (Noy 2004, Shvaiko et Euzenat 2005, Bach 2006) sont basées sur les mesures de similarité entre chaînes de caractères et des structures composites. Les ontologies à aligner peuvent être représentées avec différents formalismes ce qui soulève le problème de l'interopérabilité des applications.

Dans le cadre de l'intégration, les techniques d'intégration doivent soulever d'autres défis comme :

- (i) la robustesse, car des erreurs mineures ne doivent pas avoir des conséquences importantes,
- (ii) l'évolutivité, car elles doivent travailler en un temps raisonnable avec un nombre considérable de données distribuées présentes sur le Web et avec des ontologies qui peuvent contenir des centaines de concepts sémantiques, même lorsqu'elles concernent des domaines spécialisés.

Afin de réaliser l'intégration des ontologies, il est nécessaire d'impliquer aussi bien la syntaxe que la sémantique des ontologies. Pour résumer, le processus de mapping est un des éléments fondamentaux du processus d'intégration d'ontologies. Il permet d'analyser et de comparer des ontologies pour déterminer les correspondances entre leurs concepts et pour détecter des éventuels conflits. Le résultat du processus de mapping est un ensemble de liens de correspondances (Elbyed 2009).

3. Objectifs

Le but de notre travail est de proposer un système capable d'identifier les relations sémantiques entre les ontologies avec l'intention de soutenir l'interopérabilité des systèmes d'informations.

L'objectif principal de cette thèse est décomposé en objectifs intermédiaires :

- **Objectif 1 :** La couverture des ontologies est très rarement la même et n'est pas complète (Elbyed 2009). D'où la nécessité d'offrir aux utilisateurs un système de mapping capable d'identifier les correspondances de tout ou partie des éléments de l'ontologie. En d'autres termes, être capable de faire de l'intégration totale ou partielle des ontologies;
- **Objectif 2 :** Beaucoup de travaux portent sur l'appariement des ontologies et un grand nombre de techniques ont vu le jour, mais ils sont concentrés sur le calcul de similarité entre les termes des concepts d'ontologies ou sur la façon de combiner ces techniques. En effet, les algorithmes existants parviennent dans le meilleur des cas à proposer 70% de mappings corrects et d'identifier 80% des correspondances existantes. De plus, ces valeurs changent selon la structure et la richesse sémantiques des ontologies à faire correspondre. Il y a donc encore un effort à faire pour avoir une fiabilité plus importante du mapping indépendamment de la conceptualisation des ontologies (Elbyed 2009). Notre but est d'analyser et exploiter les définitions des entités sémantiques (concepts et rôles) de ces ontologies afin d'améliorer les résultats du processus de mapping.
- **Objectif 3 :** Les études d'évaluation ont montré que le mapping résultant des approches d'appariement existantes soit il contient un montant équitable des erreurs ou bien il ne couvre qu'une petite partie des ontologies concernées

(Euzenat et al. 2006) (Euzenat et al. 2007) (Caracciolo et al 2008). Afin de pallier ces insuffisances, nous proposons de réduire l'application des mesures autant que possible en utilisant des outils plus fiables comme le raisonnement à partir de cas (RÀPC) basé sur des mécanismes formels tels que la subsomption.

- **Objectif 4 :** La phase de validation et d'interaction avec l'utilisateur est une phase clé dans le processus de mapping, mais elle est très compliquée. Pour l'ensemble des systèmes existants, les interactions avec l'utilisateur portent directement sur la validation des liens de correspondance entre les concepts des ontologies, ce qui rend la phase de validation des mappings un processus lourd et moins sûr (Elbyed 2009). Partant de ce constat, il faudrait trouver un autre moyen de valider les résultats, qui soit plus automatique.

4. Contributions de la thèse

Compte tenu des objectifs cités auparavant, les principales contributions de notre thèse peuvent être récapitulées dans les points suivants :

1. Nous proposons un système basé sur un mécanisme du Raisonnement à partir de cas (RÀPC) pour la détection des relations sémantiques entre les entités sémantiques des ontologies en utilisant les techniques d'extraction de connaissances dans des bases de données (ECBD) et la distance sémantique et nous procédons à des tests de vérification formelle pour évaluer notre approche.
2. Problème de l'automatisation du processus de mapping: La plupart des solutions de mapping existantes ne sont pas complètement automatisées. En effet, ces approches automatisent seulement une partie du processus (Elbyed 2009). Nous proposons d'aller plus loin dans l'automatisation du mapping;
3. Problème de la précision des résultats: Afin de minimiser le têt d'erreurs, nous proposons de réduire l'application des mesures de similarité autant que possible en exploitant tous les liens (liens hiérarchiques, sémantiques) existants entre les concepts de chaque ontologie dans le processus de filtrage pour détecter de nouveaux liens sémantiques parmi les résultats de mapping déjà obtenus.
4. Enfin, Problème de la vérification: Afin d'alléger le processus de vérification fait à travers une interaction avec un expert sur tous les liens de correspondance trouvés par le système, nous proposons d'utiliser un outil formel.

5. Organisation de la thèse

Le manuscrit est structuré en quatre chapitres et une conclusion générale. Les deux premiers sont dédiés à l'état de l'art important pour la bonne compréhension du contexte de travail. Les deux derniers chapitres sont consacrés à la présentation de notre contribution.

Dans le premier chapitre nous présentons le contexte scientifique de nos travaux en introduisant les concepts de base des ontologies et du web sémantique ; plusieurs langages et outils de représentation des ontologies sont présentés.

Le deuxième chapitre présente une étude structurée des techniques de mise en correspondance d'ontologies. Tout d'abord, nous commençons par la présentation des hétérogénéités qui peuvent exister entre les différentes ontologies et leurs impacts sur le processus de l'intégration. Puis nous effectuons une classification des modes d'intégration des ontologies, avant de terminer par une définition du calcul des mesures de similarité entre les entités sémantiques.

Le chapitre trois est consacré à la présentation de notre système. Nous commençons par la présentation de cycle de vie du mécanisme du RÀPC. Puis, nous détaillons notre système qui distingue deux phases: la phase de conception et la phase de production. Au cours du processus, nous utilisons un exemple pour bien expliquer le fonctionnement du système proposé.

Le chapitre quatre regroupe les aspects techniques de l'implantation de notre système, ainsi que l'application de ce dernier dans une étude de cas adéquate. Nous commençons par donner les outils et les technologies liés à la mise en œuvre de chaque composant. Par la suite, nous proposons une étude de cas, afin de motiver l'intérêt de notre approche d'un point de vue pratique.

Enfin, le manuscrit se termine par une conclusion générale qui récapitule les travaux réalisés et propose quelques visions pour les travaux futurs.

Chapitre I : Généralités sur le web sémantique et les ontologies

1. Introduction

Selon Tim Berners Lee (Berners-Lee 1999) et (Berners-Lee et al. 2001), le Web Sémantique doit permettre de porter le Web actuel à son plein potentiel. Le Web actuel est principalement tourné vers les êtres humains. L'objectif du Web Sémantique est de rendre le contenu du Web compréhensible à des machines. Le procédé consiste à exploiter:

- Des ontologies, une ontologie est un vocabulaire constitué de concepts, de relations et des axiomes liés à un certain domaine. (Gruber 1992) définit une ontologie comme étant « la spécification d'une conceptualisation ».
- Un langage commun pour exprimer les ontologies et décrire des annotations utilisant les termes de ces ontologies.
- Des moteurs de raisonnement permettant d'inférer sur les annotations d'après les axiomes déclarés dans les ontologies.

La suite de ce chapitre est organisée comme suit : dans la deuxième section, nous définissons le Web sémantique. Dans la troisième section, nous tentons de revenir sur les origines et les caractéristiques de l'ontologie en général. La quatrième section est consacrée aux langages utilisés pour la représentation des ontologies au sein du Web sémantique, à savoir, le langage *OWL*. Le chapitre se termine par une conclusion.

2. Le Web sémantique

Pour bien comprendre ce qu'est le Web sémantique, nous commençons par une définition. *Le grand dictionnaire terminologique* définit le terme sémantique comme « l'ensemble des relations entre les caractères, ou groupes de caractères, et leur significations, indépendamment de la façon de les employer ou de les interpréter. » Il précise par la suite que « si, en linguistique, la sémantique porte sur l'étude du sens à partir de la combinaison des mots, en intelligence artificielle, elle porte sur la capacité d'un réseau [le Web] à représenter de la manière la plus humaine possible des relations entre des objets, des idées ou des situations. »

Le terme sémantique implique donc que la machine ne se contentera plus de présenter visuellement les données du Web, mais, en les reliant, elle pourra conserver les significations qui leur sont attribuables. Or, en transformant le contenu du Web pour qu'il soit « compréhensible » par la machine et non seulement présentable, nous permettons à cette même machine d'être plus efficace dans le traitement de l'information. Ainsi, le dialogue avec les moteurs de recherche devient possible. Nous sommes alors en mesure de nous exprimer dans des termes que nos ordinateurs peuvent aussi interpréter et échanger. Il est également possible d'automatiser, d'intégrer et de réutiliser l'information entre diverses applications

Le Web sémantique est décrit généralement comme un Web destiné aux machines. Disposer d'un Web dont le contenu est abordable par les machines peut apporter de grands bénéfices:

- L'automatisation de nombreuses tâches fondées sur le contenu comme la recherche de ressources ayant un contenu particulier, la comparaison du contenu de ressources (pages, bases de données, ontologies, etc...). Le Web sémantique permettrait de résoudre la relative difficulté de trouver de l'information sur le web.
- L'automatisation de tâches liées à la mise en relation de ce contenu comme la composition de réponses fournies par plusieurs ressources.
- La description de ressources informatiques (services) par leurs conditions d'activations, résultat, qualité, etc... permet d'imaginer la recherche, l'invocation et la connexion automatique de ces ressources.

La plus value du Web sémantique sera principalement dans la mise en relation d'éléments épars comme dans la composition de services ou de réponses. Ces capacités nécessitent l'interopérabilité entre les différentes ressources ou leurs descriptions.

Pour réaliser une telle innovation, le web sémantique exploite la notion d'ontologie.

3. Présentation de la notion d'ontologie

Dans cette section, nous présentons un aperçu global sur la notion d'ontologie.

3.1 Origine de l'ontologie

Ontologie est un terme qui a tout d'abord été défini en Philosophie comme une branche de la Métaphysique qui s'intéresse à l'existence, à l'être en tant qu'être et aux catégories fondamentales de l'existant. En effet, ce terme est construit à partir des racines grecques **ontos** qui veut dire ce qui existe, l'Être, l'existant, et **logos** qui veut dire l'étude, le discours, d'où sa traduction par *l'étude de l'Être* et par extension *de l'existence* (Amardeilh 2007).

Dans la philosophie classique, l'Ontologie correspond à ce qu'Aristote appelait la Philosophie première (*protè philosopha*), c'est-à-dire *la science de l'être en tant qu'être*, par opposition aux philosophies secondes qui s'intéressaient, elles, à l'étude des manifestations de l'être (les existants) (Psyché 2007).

Bien que ce soient les Grecs qui aient inventé cette science, ils ne l'avaient pas appelé « Ontologie », le terme étant beaucoup plus récent (XVIIe siècle) que la discipline qu'il désigne. La discipline elle-même a évolué vers une voie imprévisible, il y a seulement

une vingtaine d'années. En effet, l'ontologie a été abordée dans le domaine de l'intelligence artificielle pour la première fois par John McCarthy qui reconnut le recoupement entre le travail fait en Ontologie philosophique et l'activité de construire des théories logiques de systèmes d'intelligence artificielle. Il affirmait déjà en 1980 que les concepteurs de systèmes intelligents fondés sur la logique devraient d'abord énumérer tout ce qui existe.

3.2. Définitions

Dans le cadre de l'intelligence artificielle, **Neeches** et ses collègues (Neeches et al. 1991) furent les premiers à proposer une définition à savoir : «*une ontologie définit les termes et les relations de base du vocabulaire d'un domaine ainsi que les règles qui indiquent comment combiner les termes et les relations de façon à pouvoir étendre le vocabulaire*».

En 1993, **Gruber** (Gruber 1993) propose la définition suivante : «*spécification explicite d'une conceptualisation*» qui est jusqu'à présent la définition la plus citée dans la littérature en intelligence artificielle. Cette définition a été modifiée légèrement par **Borst** (Borst 1997) comme «*spécification formelle d'une conceptualisation partagée*».

Ces deux définitions sont regroupées dans celle de **Studer** (Studer et al. 1998) comme «*spécification formelle et explicite d'une conceptualisation partagée*».

- *Formelle* : l'ontologie doit être lisible par une machine, ce qui exclut le langage naturel. *Explicite* : la définition explicite des concepts utilisés et des contraintes de leur utilisation.
- *Conceptualisation* : le modèle abstrait d'un phénomène du monde réel par identification des concepts clefs de ce phénomène.
- *Partagée* : l'ontologie n'est pas la propriété d'un individu, mais elle représente un consensus accepté par une communauté d'utilisateurs.

Pour **Guarino & Giaretta** (Guarino et Giaretta 1995) «*Une ontologie est une spécification rendant partiellement compte d'une conceptualisation*». **Swartout** et ses collègues (Swartout et al. 1997) la définissent comme suit : «*une ontologie est un ensemble de termes structurés de façon hiérarchique, conçue afin de décrire un domaine et qui peut servir de charpente à une base de connaissances*».

La même notion est également développée par **Gomez** (Gomez 1999) comme: «*une ontologie fournit les moyens de décrire de façon explicite la conceptualisation des connaissances représentées dans une base de connaissances.* »

Pour conclure, nous pouvons donc affirmer que les définitions du terme ontologie abondent dans la littérature scientifique. Les définitions, dans leur diversité, offrent des points de vue à la fois différents et complémentaires sur un même concept.

En clair, une **ontologie** fournit les moyens d'exprimer les concepts d'un domaine en les organisant hiérarchiquement et en définissant leurs propriétés sémantiques dans un langage de représentation des connaissances formel favorisant le partage d'une vue consensuelle sur ce domaine entre les applications informatiques qui en font usage.

3.3. Composantes d'une ontologie

Comme nous l'avons abordé, les ontologies fournissent un vocabulaire commun d'un domaine et définissent la signification des termes et des relations entre elles. La connaissance dans les ontologies est principalement formalisée en utilisant les cinq types de composants (Gomez et Benjamins 1999) à savoir : concepts (ou classes), relations (ou propriétés), fonctions, axiomes (ou règles) et instances (ou individus).

3.3.1. Les concepts

Un concept est un constituant de la pensée (un principe, une idée, une notion abstraite) sémantiquement évaluable et communicable. L'ensemble des propriétés d'un concept constitue sa compréhension ou son intension et l'ensemble des êtres qu'il englobe, son extension.

Ces concepts selon Gomez (Gomez et Benjamins 1999) peuvent être classifiés selon trois dimensions:

- Niveau d'abstraction (concrets ou abstraits).
- Atomicité (élémentaires ou composés).
- Niveau de réalité (réels ou fictifs).

Un concept est composé de trois parties (Hernandez 2005):

- **Une notion** : elle correspond à la sémantique du concept, elle est définie à travers ses propriétés et ses attributs. Elle est appelée **intention** du concept.
- **Un ensemble d'objets** : il correspond aux objets définis par le concept, il est appelé **extension** du concept. Les objets sont les **instances** du concept.
- **Un (ou plusieurs) terme(s)** : les termes permettent de désigner le concept. Ces termes sont aussi appelés **labels** de concept.

Exemple : le **terme** « lapin » renvoie à la **notion** « animal » possédant de longues oreilles et une queue et à l'**ensemble des objets** ayant cette description.

a. Les propriétés portant sur un concept

Selon Guarino, les propriétés portant sur un concept sont :

- **La généralité** : un concept est générique s'il n'admet pas d'extension. *Exemple* : la vérité est un concept générique.
- **L'identité** : un concept porte une propriété d'identité si cette propriété permet de conclure quant à l'identité de deux instances de ce concept. Cette propriété peut porter sur des attributs du concept ou sur d'autres concepts.
Exemple : le concept d'étudiant porte une propriété d'identité liée au numéro de l'étudiant, deux étudiants étant identiques s'ils ont le même numéro.
- **La rigidité** : un concept est dit rigide si toute instance de ce concept en reste instance dans tous les mondes possibles.
Exemple : humain est un concept rigide, étudiant est un concept non rigide.
- **L'anti-rigidité** : un concept est anti-rigide si toute instance de ce concept est essentiellement définie par son appartenance à l'extension d'un autre concept.
Exemple : étudiant est un concept anti-rigide car l'étudiant est avant tout un humain.
- **L'unité** : un concept est un *concept unité*, si pour chacune de ses instances, les différentes parties de l'instance sont liées par une relation qui ne lie pas d'autres instances du concept.
Exemple : les deux parties d'un couteau, manche et lame sont liées par une relation «Emmanché » qui ne lie que cette lame et ce manche.

b. Les propriétés portant sur deux concepts

- **L'équivalence** : deux concepts sont équivalents s'ils ont la même extension.
Exemple : étoile du matin et étoile du soir.
- **La disjonction** : (on parle aussi d'incompatibilité) deux concepts sont disjoints si leurs extensions sont disjointes.
Exemple : homme et femme.
- **La dépendance** : Un concept C1 est dépendant d'un concept C2 si pour toute instance de C1 il existe une instance de C2 qui ne soit ni partie ni constituant de l'instance de C1.
Exemple : parent est un concept dépendant de enfant (et vice-versa).

3.3.2. Les relations

Elles représentent des interactions entre concepts permettant de construire des représentations complexes de la connaissance du domaine (Charlet et al. 2004). Elles établissent des liens sémantiques binaires, organisables hiérarchiquement.

Exemple : les concepts « Personnalité » et « Film » sont reliés entre eux par la relation sémantique « réalise (Personnalité, Film) ».

a. Les propriétés intrinsèques à une relation

- **Les propriétés algébriques** : symétrie, réflexivité, transitivité
- **La cardinalité** : nombre possible de relations de ce type entre les mêmes concepts (ou instances de concept). Les relations portant une cardinalité représentent souvent des attributs.

Exemple: une pièce a au moins une porte.

b. Les propriétés liant deux relations

- **L'incompatibilité** : on dit que deux relations R1 et R2 sont incompatibles si elles ne peuvent lier les mêmes instances de concepts.
Exemple : « être rouge » et « être vert » sont deux relations incompatibles.
- **L'inverse** : on dit que deux relations binaires R1 et R2 sont inverses lune de l'autre si, quand R1 lie deux instances I1 et I2, alors R2 lie I2 et I1.
Exemple : « a pour père » et « a pour enfant » sont deux relations inverses lune de l'autre.
- **L'exclusivité** : deux relations R1 et R2 sont exclusives si, quand R1 lie des instances de concepts, R2 ne lient pas ces instances, et vice-versa. L'exclusivité entraîne l'incompatibilité.
Exemple : l'appartenance et la non appartenance sont exclusives.

c. Les propriétés liant une relation et des concepts

- **Le lien relationnel** : Il existe un lien relationnel entre une relation R et deux concepts C1 et C2 si, pour tout couple d'instances des concepts C1 et C2, il existe une relation de type R qui lie les deux instances de C1 et C2. Un lien relationnel peut être contraint par une propriété de cardinalité, ou porter directement sur une instance de concept (Kassel 2002).
Exemple : Il existe un lien relationnel entre les concepts « texte » et « auteur » d'une part et la relation « a pour auteur » d'autre part.

- **La restriction de relation :** Pour tout concept de type C1, et toute relation de type R liant C1, les autres concepts liés par la relation sont d'un type imposé.
Exemple: si la relation « mange » portant sur une « personne » et un « aliment » lie une instance de « végétarien », concept subsumé par « personne », l'instance de « aliment » est forcément instance de « végétaux ».

3.3.3. Les fonctions

Elles présentent des cas particuliers de relations dans lesquelles le *nième* élément de la relation est unique pour les *n-1* éléments précédents (Gomez et Benjamins 1999).

Formellement, les fonctions sont définies telles que :

$$F : c1 * c2 * ... * cn-1 \rightarrow cn.$$

Exemple de fonctions : « père-de » et « carré » sont des fonctions binaires. Tandis que, « prix-de voiture-usagée » qui calcule le prix d'une voiture de seconde main en fonction du modèle de voiture, de la date de fabrication et du nombre de kilomètres est une fonction ternaire.

3.3.4. Les axiomes (ou Règles)

Les axiomes sont des expressions qui sont toujours vraies. Ils ont pour but de définir dans un langage logique la description des concepts et des relations permettant de représenter leur sémantique. Ils représentent les intentions des concepts et des relations du domaine et, de manière générale, les connaissances n'ayant pas un caractère strictement terminologique (Staab et Maedche 2000). Leur inclusion dans une ontologie peut avoir plusieurs objectifs :

- Définir la signification des composants.
- Définir des restrictions sur la valeur des attributs.
- Définir les arguments d'une relation.
- Vérifier la validité des informations spécifiées ou en déduire de nouvelles.

3.3.5. Les instances (ou individus)

Elles constituent la définition extensionnelle de l'ontologie ; elles sont utilisées pour représenter des éléments dans un domaine.

Exemple: les individus *Mohamed* et *Maha* sont des instances du concept «Personne».

Après cette présentation de l'origine et de la définition de la notion d'ontologie, nous allons présenter quelques rôles de l'ontologie.

3.4. Rôles de l'ontologie

Plusieurs chercheurs se sont intéressés à la finalité des exploitations des ontologies. Nous précisons dans ce qui suit les rôles et l'intérêt des ontologies au sein des systèmes à base de connaissances (SBC) et du Web Sémantique :

- **Les connaissances du domaine d'un SBC :** Les ontologies servent à représenter les connaissances du domaine d'un SBC. En particulier, elles servent de squelette à la représentation des connaissances du domaine dans la mesure où elles décrivent les objets, leurs propriétés et la façon dont ils peuvent se combiner pour constituer des connaissances du domaine complètes.
- **La communication :** Les ontologies peuvent intervenir dans la communication entre personnes, organisations et logiciels (Uschold et Grüninger 1996). En effet, les ontologies servent par exemple, à créer au sein d'un groupe ou d'une organisation un « vocabulaire conceptuel commun ». Dans ce cas, on est plutôt dans le cadre d'une ontologie informelle. Dans le cas de la communication entre personnes et systèmes, l'ontologie est formelle et sert en général une tâche précise dans le SBC ou le système d'information. L'ontologie est un puissant moyen pour lever les ambiguïtés dans les échanges.
- **L'interopérabilité :** le développement et l'implantation d'une représentation explicite d'une compréhension partagée dans un domaine donné, peut améliorer la communication, qui à son tour permet une plus grande réutilisation, un partage plus large et une interopérabilité plus étendue (Uschold et Grüninger 1996). L'interopérabilité est donc une spécialisation de la communication qui permet de répertorier les concepts que des applications peuvent s'échanger même si elles sont distantes et développées sur des bases différentes.
- **L'aide à la spécification de systèmes :** La plupart des logiciels conventionnels sont construits avec une conceptualisation implicite et que la nouvelle génération des systèmes utilisant les travaux en intelligence artificielle devrait être basée sur une conceptualisation explicitement représentée (Mizoguchi et Ikeda 1996). En effet, l'ontologie fournit une classification des objets que doit manipuler le système.

- **L'indexation et la recherche d'information** : Dans le Web Sémantique, les ontologies y sont utilisées pour déterminer les index conceptuels décrivant les ressources sur le Web.

Dans la section suivante, nous nous intéressons aux différentes classifications pour les ontologies.

3.5. Classification des ontologies

3.5.1 Dimensions de classification des ontologies

Les ontologies peuvent être classifiées selon plusieurs dimensions. Parmi celles-ci, nous en examinerons quatre :

1. Objet de conceptualisation.
2. Niveau de détail.
3. Niveau de complétude.
4. Niveau de formalisme de représentation.

Ces dimensions de classification sont illustrées dans la figure 1.1.

a. Typologie selon l'objet de conceptualisation

Les ontologies classifiées selon leur objet de conceptualisation par (Gómez 1999a), (Guarino 1997a), (Mizoguchi 1998), (Mizoguchi et Ikeda 1996), (VanHeijst et al. 1997), (Vanwelkenhuysen et Mizoguchi 1994), (Vanwelkenhuysen et Mizoguchi 1995) et (Wielinga et Schreiber 1993), le sont de la façon suivante :

- **Ontologie de représentation des connaissances** (Gómez 1999b), (VanHeijst et al. 1997) ce type d'ontologies regroupe les concepts (primitives de représentation) impliqués dans la formalisation des connaissances. Un exemple est l'*ontologie de Frame* qui intègre les primitives de représentation des langages à base de *frames* : classes, instances, facettes, propriétés/*slots*, relations, restrictions, valeurs permises, etc.
- **Ontologie supérieure ou de Haut niveau** (Guarino 1997b), (Sowa 1995), (Sowa 1995a). Cette ontologie est une ontologie générale. Son sujet est l'étude des catégories des choses qui existent dans le monde, soit les concepts de haute abstraction tels que: les entités, les événements, les états, les processus, les actions, le temps, l'espace, les relations, les propriétés. L'ontologie de haut de

niveau est fondée sur : la théorie de l'identité, la méréologie (*theory of whole and parts role*) et la théorie de la dépendance. Guarino et Sowa ont poursuivi chacun indépendamment des recherches sur la théorie de l'ontologie. Tous deux intègrent les fondements philosophiques comme étant des principes à suivre pour concevoir l'ontologie de haut niveau ou supérieure. Sowa introduit deux concepts importants, *Continuant* et *Occurrent*, et obtient douze catégories supérieures en combinant sept propriétés primitives. L'ontologie supérieure de Guarino consiste en deux mondes : une ontologie des *Particuliers* (choses qui existent dans le monde) et une ontologie des *Universels* comprenant les concepts nécessaires à décrire les Particuliers. La conformité aux principes de l'ontologie supérieure a son importance, lorsque le but est de standardiser la conception des ontologies.

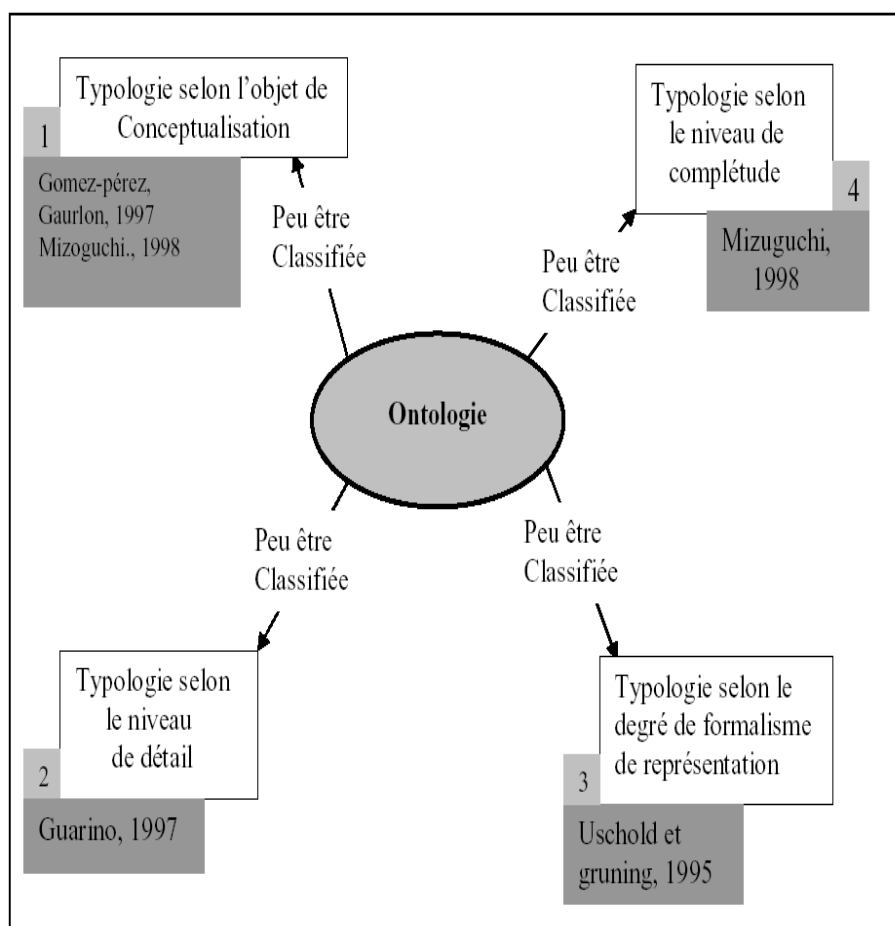


Figure 1.1- Typologies d'ontologies selon quatre dimensions de classification
(Psyché et al. 2003)

- **Ontologie Générique** (Gómez 1999a), (Gómez 1999b), (VanHeijst et al. 1997). Cette ontologie aussi appelée, méta-ontologies ou *core ontologies*, véhicule des connaissances génériques moins abstraites que celles véhiculées par l'ontologie de haut niveau, mais assez générales néanmoins pour être réutilisées à travers différents domaines. Elle peut adresser des connaissances factuelles (*Generic domain ontology*) ou encore des connaissances visant à résoudre des problèmes génériques (connaissances procédurales) appartenant à ou réutilisables à travers différents domaines (*Generic task ontology*). Deux exemples de ce type d'ontologies sont : 1) l'ontologie méréologique (Borst 1997) contenant des relations, *Partie-de* et 2) l'ontologie topologique contenant des relations, *Associé-à*.
- **Ontologie du Domaine** (Mizoguchi et al. 2000). Cette ontologie régit un ensemble de vocabulaires et de concepts qui décrit un domaine d'application ou monde cible. Elle permet de créer des modèles d'objets du monde cible. L'ontologie du domaine est une méta-description d'une représentation des connaissances, c'est-à-dire une sorte de méta-modèle de connaissance dont les concepts et propriétés sont de type déclaratif. La plupart des ontologies existantes sont des ontologies du domaine. Selon Mizoguchi, l'ontologie du domaine caractérise la connaissance du domaine où la tâche est réalisée. Dans le contexte de la formation à distance, un domaine serait par exemple : le téléapprentissage.
- **Ontologie de Tâches** (Mizoguchi et al. 2000). Ce type d'ontologies est utilisé pour conceptualiser des tâches spécifiques dans les systèmes, telles que les tâches de diagnostic, de planification, de conception, de configuration, de tutorat, soit tout ce qui concerne la résolution de problèmes. Elle régit un ensemble de vocabulaires et de concepts qui décrit une structure de résolution des problèmes inhérente aux tâches et indépendante du domaine. Selon (Mizoguchi et al. 2000), l'ontologie de tâche caractérise l'architecture computationnelle d'un système à base de connaissances qui réalise une tâche. Deux exemples d'utilisation de l'ontologie de tâche dans le domaine de l'éducation sont les suivants :
 - 1) l'ontologie de formation par ordinateur - *Computer Based Training Ontology* (Jin et al. 1999) - qui régit un ensemble de concepts spécifiques à un système d'apprentissage inhérent à des ontologies de tâche ;
 - 2) l'ontologie des objectifs d'apprentissage - *Learning Goal Ontology* (Inaba et al. 2000) - qui décrit les rôles des apprenants et des agents dans le cadre d'un apprentissage collaboratif.

- **Ontologie d'Application.** Cette ontologie est la plus spécifique. Les concepts dans l'ontologie d'application correspondent souvent aux rôles joués par les entités du domaine tout en exécutant une certaine activité (Maedche 2002).

b. Typologie selon le niveau de détail de l'ontologie

Par rapport au niveau de détail utilisé lors de la conceptualisation de l'ontologie en fonction de l'objectif opérationnel envisagé pour l'ontologie, deux catégories au moins peuvent être identifiées (Psyché et al. 2003) :

1) **Granularité fine** : correspondant à des ontologies très détaillées, possédant ainsi un vocabulaire plus riche capable d'assurer une description détaillée des concepts pertinents d'un domaine ou d'une tâche. Ce niveau de granularité peut s'avérer utile lorsqu'il s'agit d'établir un consensus entre les agents qui l'utiliseront;

2) **Granularité large** : correspondant à un vocabulaire moins détaillé comme par exemple dans les scénarios d'utilisation spécifiques où les utilisateurs sont déjà préalablement d'accord à propos d'une conceptualisation sous-jacente (Fürst 2002), (Guarino 1997a). Les ontologies de haut niveau possèdent une granularité large, compte tenu que les concepts qu'elles traduisent sont normalement raffinés subséquemment dans d'autres ontologies de domaine ou d'application (Fürst 2002).

c. Typologie selon le niveau de complétude

Le niveau de complétude a été abordé par (Mizoguchi 1998) et (Bachimont 2000). À titre d'exemple, nous décrivons la typologie de (Bachimont 2000). Ce dernier propose la classification sur trois niveaux suivante :

- **Niveau 1 - Sémantique** : Tous les concepts (caractérisés par un terme/libellé) doivent respecter les quatre principes différentiels :
 - 1) Communauté avec l'ancêtre;
 - 2) Différence (spécification) par rapport à l'ancêtre;
 - 3) Communauté avec les concepts frères (situés au même niveau);
 - 4) Différence par rapport aux concepts frères (sinon il n'aurait pas lieu de le définir).

Ces principes correspondent à *l'engagement sémantique* qui assure que chaque concept aura un sens univoque et non contextuel associé. Deux concepts sémantiques sont identiques si l'interprétation du terme/libellé à travers les quatre principes différentiels aboutit à un sens équivalent.

- **Niveau 2 - Référentiel** : Outre les caractéristiques énoncées au niveau précédent, les concepts référentiels (ou formels) se caractérisent par un terme/libellé dont la sémantique est définie par une extension d'objets. *L'engagement ontologique* spécifie les objets du domaine qui peuvent être associés au concept, conformément à sa signification formelle. Deux concepts formels seront identiques s'ils possèdent la même extension (exemple : les concepts *d'étoile du matin* et *d'étoile du soir* associés à Vénus).
- **Niveau 3 - Opérationnel** : Outre les caractéristiques énoncées au niveau précédent, les concepts du niveau opérationnel ou computationnel sont caractérisés par les opérations qu'il est possible de leur appliquer pour générer des inférences (*engagement computationnel*). Deux concepts opérationnels sont identiques s'ils possèdent le même potentiel d'inférence.

d. Typologie selon le niveau du formalisme

Par rapport au *niveau du formalisme de représentation* du langage utilisé pour rendre l'ontologie opérationnelle, (Uschold et Grüninger 1996) proposent une classification comprenant quatre catégories :

- 1) **Informelles** : ontologies opérationnelles dans un langage naturel (sémantique ouverte) ;
- 2) **Semi-informelles** : utilisation d'un langage naturel structuré et limité ;
- 3) **Semi-formelles** : langage artificiel défini formellement;
- 4) **Formelles** : utilisation d'un langage artificiel contenant une sémantique formelle, ainsi que des théorèmes et des preuves des propriétés telles la robustesse et l'exhaustivité (Gómez 1999a).

Selon Studer, "*il y a différents types d'ontologie et chaque type remplit un rôle différent dans le processus de construction du modèle du domaine*". (Studer et al. 1998).

La section suivante aborde les formalismes de représentation des ontologies.

3.6. Formalismes de représentation des ontologies

Il existe actuellement de nombreux formalismes pour représenter les ontologies. Parmi ces formalismes nous pouvons distinguer ceux qui sont simples tel que les frames et ceux qui sont complexes tels que les graphes conceptuels et les logiques de descriptions que nous utilisons dans notre travail. Nous allons présenter ces trois formalismes dans la section suivante.

Le langage pour décrire les ontologies doit avoir une sémantique bien définie pour pouvoir réaliser des raisonnements sur les formules de ce langage. Un tel langage doit définir au minimum des primitives telles que l'héritage pour être utilisable dans des situations pratiques; des langages plus expressifs peuvent être fondés sur les langages de représentation des connaissances décrits dans les sections suivantes (par exemple les logiques de description).

3.6.1 Les frames :

Introduits dès les années 70 par Minsky (Minsky 1975) comme une modélisation de base pour la représentation de connaissances dans le domaine d'Intelligence Artificielle (AI), le modèle des frames a depuis été adapté à d'autres problématiques puisqu'il a donné naissance au modèle objet, qui envahit peu à peu les différentes branches de l'informatique.

L'idée de frame est très simple. Un « frame » est dans ce contexte un objet nommé, qui est utilisé pour représenter un certain concept dans un domaine.

Un frame représente n'importe quelle primitive conceptuelle et il est doté d'attributs (slots), qui peuvent porter différentes valeurs (facets), et d'instances (Kifer et al. 1995).

Il y a une correspondance entre les systèmes de frame et ceux orientés objet où les classes et les instances correspondent avec les frames, les attributs et les associations de classes avec les slots. Entre les frames, il y a aussi la spécialisation qui donne l'héritage dans les concepts de frame. Une frame F1 est plus spécifique qu'une frame F2 si toute instance de F1 est instance de F2.

Le formalisme F-logic (Kifer et al. 1995) était proposé comme le fondement logique pour les langages de frame et orientés objet. Il permet de comprendre le modèle sémantique dans tous les langages de frame et d'aider à la construction d'une base de connaissances.

Parmi les langages de frame on peut citer YAFOOL (Yet Another Frame Based Object-Oriented Language), OML (Ontology Markup Language), Ontolingua, KRL (Knowledge Representation Language), SHIRKA et bien d'autres encore.

Le langage SHIRKA est fondé sur un modèle classe/instance. C'est un modèle totalement uniforme dans lequel tout schéma, attribut ou facette est une instance d'un schéma de plus haut niveau. Les valeurs apparaissant dans une facette sont toujours des instances ou des références à des instances. L'inférence des valeurs d'attribut indéterminées ne fait pas appel à des règles, mais à des filtres décrits par des schémas. Les mécanismes d'inférence qu'il utilise sont : héritage, attachement procédural, filtrage, valeur par défaut, spécialisation et classification (Jenhani 2003).

Cependant, il faut noter l'arrivée en 1999 de XOL (« XML Ontology Exchange Language) langage d'échange qui utilise la syntaxe XML. Un schéma XML est automatiquement généré. La phase de génération utilise des règles de passage qui permettent d'exprimer les connaissances existantes dans l'ontologie en terme de Schéma-XML. Un prototype a été réalisé, qui permet de faire l'analyse lexicale et syntaxique du fichier XOL entrant et de générer un Schéma-XML en sortie. Son développement a été inspiré d'Ontolingua et de OML.

3.6.2 Graphes conceptuels

Les graphes conceptuels ont été initialement conçus pour l'analyse et la compréhension du langage naturel. De par leur similarité avec les réseaux sémantiques, leur pouvoir d'expression est tel qu'ils sont directement applicables à la représentation des connaissances. Les graphes conceptuels sont décomposés en deux niveaux: le niveau terminologique où sont décrit les concepts, les relations et les instances de concepts, ainsi que les liens de subsomption entre concepts et entre relations et le niveau assertionnel où sont représentés les faits, les règles et les contraintes sous forme de graphes où les sommets sont des instances de concepts et les arcs des relations (Jenhani 2003).

- Le Modèle des Graphes Conceptuels

Dans le modèle des Graphes Conceptuels, un *support* représente la connaissance ontologique et les graphes conceptuels construits sur ce support représentent la connaissance assertionnelle relative à la connaissance ontologique (Delteil 2002).

Définition 1: support

Un *support* S est un couple (T_C, T_R) tel que :

- T_C , *ensemble des types de concepts*, est un ensemble partiellement ordonné par une relation « sorte de » (notée \leq) possédant un plus grand élément (noté T) appelé type universel.
- T_R , *ensemble des types de relations*, est un ensemble partiellement ordonné, partitionné en sous-ensembles de types de relations de même arité. $T_R = T_{R1} \cup \dots \cup$

T_{Rp} , où T_{Ri} est l'ensemble des types de relations d'arité i . Tout T_{Ri} admet un plus grand élément (noté T_i).

Le support décrit en figure 1.2 définit des types de concepts tels que *Ville* (qui est une sorte de *Lieu*) et des types de relations tels que *agent*.

Un GC exprime un fait en utilisant le vocabulaire défini dans le support.

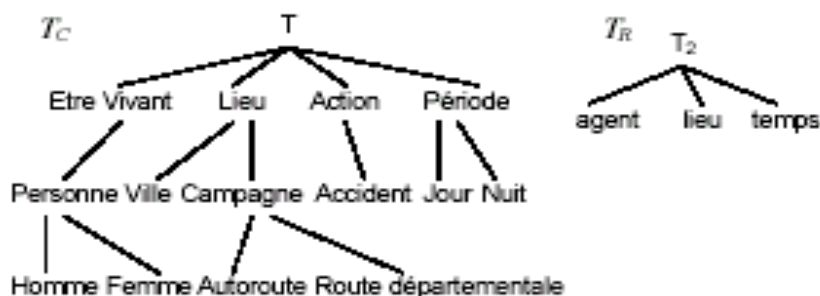


Figure.1.2 – Un support (Delteil 2002).

Définition 2 : graphe conceptuel

Un *graphe conceptuel* $G = (CG, RG, EG, \text{étiq}G)$ défini sur un support S , est un multigraphe non orienté, biparti où :

- CG est l'ensemble des *sommets concepts* et RG l'ensemble des *sommets relations*.
- EG est l'ensemble des *arêtes*. Toutes les arêtes d'un graphe conceptuel G ont une extrémité dans CG et l'autre dans RG .
- $\text{étiq}G$ est une application qui à tout sommet de NG et à toute arête de EG associe une étiquette : si $r \in RG$, $\text{étiq}G(r) \in TR$; si $c \in CG$, $\text{étiq}G(c) \in TC$; si $e \in EG$, $\text{étiq}G(e) \in Te$.

L'ensemble des arêtes adjacentes à tout sommet relation r est totalement ordonné, ce que l'on représente en étiquetant les arêtes de 1 au degré de r .



Figure. 1.3 – Un graphe conceptuel (Delteil 2002).

La figure 1.3 présente un GC défini sur le support décrit dans la figure 1.2. Les sommets concepts sont représentés par des rectangles et les sommets relations par des ellipses.

Le graphe de la figure 1.3 peut être interprété par «Accident mortel» (accident dans lequel une personne est morte). Comme l'illustre la figure, la nature graphique des GC fait qu'il est facile de les interpréter, de les modifier ou d'en créer de nouveaux. Cette facilité de création est renforcée par la séparation explicite de différents types de connaissances, et plus précisément par la définition d'un objet distinct des graphes, le support, qui aide lors de la création des graphes, car les étiquettes des sommets doivent être choisies dans ce support.

Le modèle des GC est doté d'une sémantique logique: les types de relations et de concepts correspondent à des prédicats, les liens « sorte de » sont représentés par des implications et les graphes par des assertions.

3.6.3 Les logiques de descriptions

Les logiques de description (LDs) découlent directement des travaux fondateurs de Brachmann et de son système KL-ONE. Depuis le début des années 90, la recherche en logique de descriptions s'est considérablement développée.

Les logiques de description peuvent être considérées comme un fragment de la logique du premier ordre, dans lequel les formules ont une variable libre pour les descriptions de concepts et deux variables libres pour les descriptions de relations (Buchheit et al. 1993).

Les LDs sont très riches en constructeurs ce qui permet l'expression détaillée des concepts. On remarque que plus les descriptions de concepts sont détaillées, plus les mécanismes d'intégration d'ontologies peuvent être efficaces dans la comparaison des concepts et des rôles.

Une LD est composée de deux parties : un langage terminologique *TBox* et un langage assertionnel *ABox*. Le langage assertionnel est dédié à la description de faits et le langage terminologique à la description de concepts et de rôles. La principale tâche de raisonnement au niveau terminologique est de calculer les relations de subsomption entre concepts (Napoli 1997).

Le *TBox* constitue la partie terminologique de la base de connaissances. Il contient la déclaration des termes et des rôles. Chaque nouveau concept est défini à base des concepts déjà définis, pour fournir à la fin un ensemble de concepts de plus en plus complexes. Cet ensemble de concepts est classifié en vérifiant la subsomption des concepts qui positionne chaque concept à la bonne place dans la hiérarchie.

Le mécanisme de subsomption est supporté par plusieurs algorithmes. Horrocks propose le système FaCT qui implémente un algorithme de classification pour la logique de description basé sur la notion de « Tableau Calculi » (Horrocks 1998).

Le *ABox* contient les individus (ou instances) des concepts. Il utilise les concepts établis dans le *TBox* et permet de définir des instances pour chaque concept ou rôle. Par exemple, l'assertion suivante permet de définir la personne « Sara » comme étant une mère : $Mère \sqcap Personne(Sara)$. Concernant les rôles, on peut définir l'assertion suivante : $A-Enfant(Sara, Darine)$ qui indique que la personne « Sara » a un enfant « Darine ».

a. Les constructeurs des LDs

Les entités de base qui sont définies et manipulées dans une logique de descriptions sont les concepts et les rôles. Un concept permet de représenter un ensemble d'individus, un rôle représente une relation binaire entre concepts.

Un concept et un rôle possèdent une description structurée, élaborée à partir d'un certain nombre de constructeurs. Les concepts et les rôles peuvent être primitifs ou définis. Les concepts (éventuellement les rôles) primitifs sont comparables à des atomes et servent de base à la construction des concepts définis (éventuellement les rôles définis).

Une sémantique est associée à chaque description de concept et de rôle par l'intermédiaire d'une interprétation. Les manipulations opérées sur les concepts et les rôles sont réalisées en accord avec cette sémantique (Napoli 1997).

- Syntaxe de la logique de description

Une LD est inductivement définie à partir d'un ensemble P_c de concepts primitifs, un ensemble P_r de rôles primitifs, des constantes T (dénote le concept le plus général, son extension inclut toutes les instances possibles) et \perp (dénote le concept le plus spécifique) et des règles de syntaxe suivantes :

$C, D \rightarrow$	T		\perp	plus général		absurde (concept vide)
				P		concept primitif
				$C \cap D$		conjonction de concepts
				$C \cup D$		disjonction de concepts
				$\neg C$		négation
				$\forall r.C$		restriction universelle

	$\exists r.C$	restriction existentielle
	$\leq n r.C$	cardinalité maximum
	$\geq n r.C$	cardinalité minimum
r	\rightarrow	q rôle primitif
	$r_1 \cap r_2$	conjonction de rôles
	$r_1 \cup r_2$	disjonction de rôles

- La conjonction des concepts pour exprimer l'intersection entre deux concepts ($C \cap D$). La sémantique associée désigne l'ensemble des individus obtenu par l'intersection des individus des concepts C et D .
- La disjonction des concepts pour exprimer l'union entre deux concepts ($C \cup D$). La sémantique associée désigne l'ensemble des individus obtenu par l'union des individus des concepts C et D .
- La négation d'un concept pour exprimer son complément ($\neg C$). La sémantique associée désigne l'ensemble des individus qui appartiennent au domaine complément.

Dans l'exemple suivant, on propose un concept qui représente les *personnes qui ne sont pas des femmes*. Pour cela, on utilise la conjonction et le complément comme suit: $\text{Personne} \cap \neg \text{Femme}$.

L'établissement des relations entre les concepts est l'une des caractéristiques importantes de la logique de description. Alors, un langage pour la logique de description doit proposer les formalismes nécessaires pour exprimer les rôles et pour établir les restrictions nécessaires sur eux. Dans ce contexte, deux éléments sont fournis ; les *quantificateurs* et les *cardinalités*.

Les *quantificateurs* permettent d'établir des restrictions de nombres sur les rôles. On trouve principalement deux quantificateurs ; le quantificateur universel \forall et le quantificateur existentiel \exists . Par exemple, pour désigner un concept dont les individus ont tous des enfants féminins on peut l'écrire comme suit : $\forall \text{A-Enfant.Femme}$.

Les *cardinalités* constituent un moyen important pour assurer une restriction des rôles. Elles permettent d'imposer des valeurs minimales ou maximales sur ces rôles.

Dans notre exemple concernant les membres d'une famille, on peut ainsi définir un concept qui a un nombre d'enfants entre 2 et 5 par : ≥ 2 A-Enfant $\cap \leq 5$ A-Enfant .

Les constructeurs utilisés dans cette syntaxe déterminent la puissance d'expression de la LD ainsi définie.

Par exemple, la description de concept suivante décrit toutes les personnes qui travaillent dans un projet financé par un institut :

$$\text{Personne} \cap \exists \text{travailledans.}(\text{Projet} \cap \exists \text{financépar.} \text{Institut}).$$

Selon les applications considérées, les constructeurs sont plus ou moins utiles. Ainsi pour décrire des objets, les constructeurs de cardinalité maximum et minimum peuvent être parfois très utiles.

- Sémantique

La signification formelle d'une description de concept est donnée au moyen d'une sémantique extensionnelle par une interprétation I qui est un couple (Δ_I, I) Le domaine Δ_I est un ensemble arbitraire non vide d'individus et I une fonction d'interprétation qui fait correspondre à un concept un sous-ensemble de Δ_I et à un rôle un sous-ensemble de $\Delta_I \times \Delta_I$.

La sémantique d'un concept de la LD définie plus haut est donnée par la

Table 2.1. – Syntaxe et Sémantique des descriptions d'une LD.

Constructeur	Syntaxe	Sémantique
Plus général / absurde	T / \perp	Δ_I / \emptyset
Conjonction	$C \cap D$	$C^I \cap D^I$
Disjonction	$C \cup D$	$C^I \cup D^I$
Négation	$\neg C$	$\Delta_I \setminus C^I$
Restriction universelle	$\forall r.C$	$\{a \in \Delta_I / \forall b \in \Delta_I, \text{ si } (a,b) \in r^I \text{ alors } b \in C^I\}$

Restriction existentielle	$\exists r.C$	$\{a \in \Delta_I / \exists b \in \Delta_I, (a,b) \in r^I \text{ et } b \in C^I\}$
Cardinalité maximum	$\leq n r.C$	$\{a \in \Delta_I / \{b \in \Delta_I / (a,b) \in r^I \text{ et } b \in C^I\} \leq n\}$
Cardinalité minimum	$\geq n r.C$	$\{a \in \Delta_I / \{b \in \Delta_I / (a,b) \in r^I \text{ et } b \in C^I\} \geq n\}$
Conjonction des rôles	$r_1 \cap r_2$	$\{(a,b) \in \Delta_I \times \Delta_I / (a,b) \in r_1^I \text{ et } (a,b) \in r_2^I\}$
Disjonction des rôles	$r_1 \cup r_2$	$\{(a,b) \in \Delta_I \times \Delta_I / (a,b) \in r_1^I \text{ ou } (a,b) \in r_2^I\}$

Voici quatre propriétés générales des LDs:

1. **Satisfiabilité :**

Un concept C est satisfaisable si et seulement si il existe une interprétation I telle que $C^I \neq \emptyset$.

2. **Subsomption :**

L'organisation des concepts et des rôles par niveau de généralité se fait par la relation de subsomption. Un concept C subsume un concept D si C est plus général que D au sens où l'ensemble des individus représenté par C contient l'ensemble des individus représenté par D. En effet, Un concept C est subsumé par D si et seulement si $C^I \subset D^I$ pour toute interprétation I.

3. **Equivalence :**

Un concept C est équivalent à D si et seulement si $C^I = D^I$ pour toute interprétation I.

4. **Incompatibilité :**

Deux concepts C et D sont incompatibles si et seulement si: $C^I \cap D^I = \emptyset$ pour toute interprétation I.

Pour conclure cette section, nous avons considérablement mis l'accent sur le formalisme des logiques descriptions. Ce formalisme de représentation est très riche en constructeurs ce qui permet l'expression détaillée des concepts. Nous remarquons que plus les descriptions des concepts sont détaillées, plus les mécanismes d'intégration d'ontologies peuvent être efficaces dans la comparaison des concepts et des rôles.

3.7 Le cycle de vie des ontologies

Les ontologies étant destinées à être utilisées comme des composants logiciels dans des systèmes répondant à des objectifs opérationnels différents, leur développement doit s'appuyer sur les mêmes principes que ceux appliqués en génie logiciel. En particulier, les ontologies doivent être considérées comme des objets techniques évolutifs et possédants un cycle de vie qui nécessite d'être spécifié.

Les activités liées aux ontologies sont d'une part des activités de gestion de projet (planification, contrôle, assurance qualité), et d'autre part des activités de développement (spécification, conceptualisation, formalisation); s'y ajoutent des activités transversales de support telles que l'évaluation, la documentation, la gestion de la configuration (Blázquez et al. 1998).

Un cycle de vie inspiré du génie logiciel est proposé dans (Dieng et al. 2001). Il comprend une étape initiale d'évaluation des besoins, une étape de construction, une étape de diffusion, et une étape d'utilisation. Après chaque utilisation significative, l'ontologie et les besoins sont réévalués et l'ontologie peut être étendue et, si nécessaire, en partie reconstruite. La phase de construction peut être décomposée en 3 étapes : conceptualisation, ontologisation, opérationnalisation (figure 1.4).

- **La conceptualisation:** identification des connaissances contenues dans un corpus représentatif du domaine.
- **L'ontologisation:** formalisation, autant que possible, du modèle conceptuel obtenu à l'étape précédente.
- **L'opérationnalisation:** transcription de l'ontologie dans un langage formel et opérationnel de représentation de connaissances.

L'étape d'ontologisation peut être complétée d'une étape d'intégration au cours de laquelle une ou plusieurs ontologies vont être importées dans l'ontologie à construire. Dans ce travail, nous s'intéressons à l'intégration des ontologies qui sont décrites en LDs.

Corcho (Corcho 2002) insiste sur le fait que les activités de documentation et d'évaluation sont nécessaires à chaque étape du processus de construction, l'évaluation précoce permettant de limiter la propagation d'erreurs. Le processus de construction peut être intégré au cycle de vie d'une ontologie comme indiqué en figure 1.5.

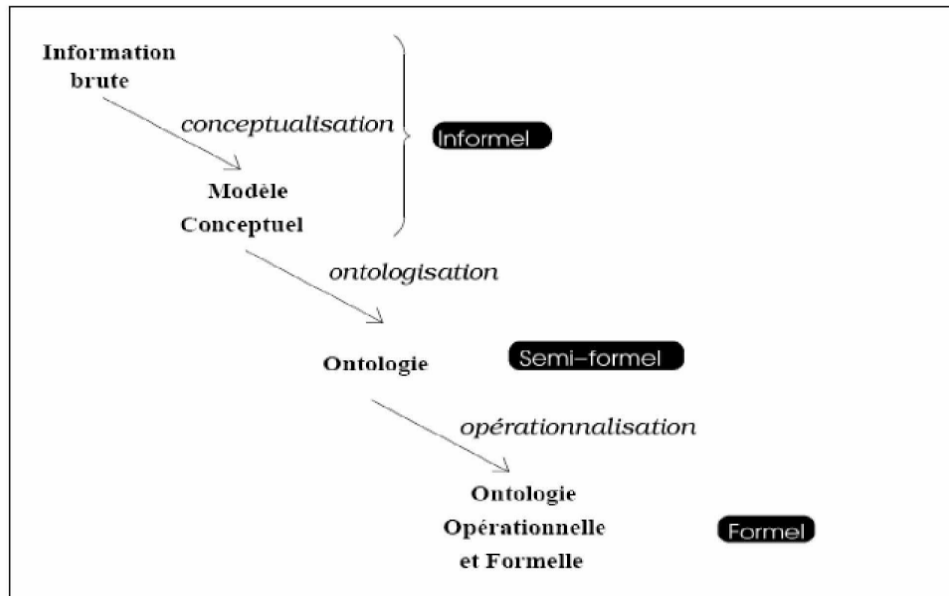


Figure- 1.4 : La phase de construction des ontologies (Dieng et al. 2001).

Dans la section suivante, nous allons présenter les différents langages qui sont développés pour l'implémentation des ontologies Web sémantique.

4. Langages pour le web sémantique

Comme nous l'avons déjà indiqué, le Web sémantique apparaît aujourd'hui comme la prochaine évolution importante du Web. De par la taille que le World Wide Web a atteinte, l'accessibilité aux informations disponibles sur celui-ci exige des moyens de raisonnement. Pour cela, différents langages de représentation des connaissances adaptés au Web ont été créés.

Dans cette section, nous présentons quelques langages de représentation des ontologies les plus connus et les plus utilisés.

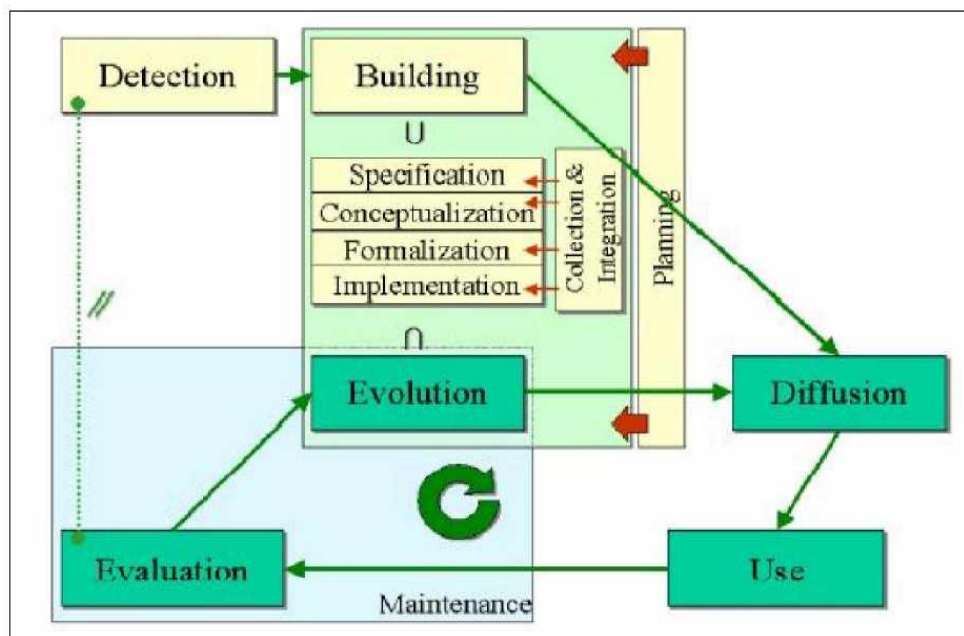


Figure- 1.5. Le cycle de vie d'une ontologie (Corcho 2002)

4.1 RDF et RDF(S) fondement du Web Sémantique

Le langage Resource Description Framework (RDF) est le standard émergent proposé par le W3C pour la représentation et l'échange de méta-données sur le Web (Rdf 1999), il est basé sur un modèle de triplets (ressource, propriété, valeur) et possède une syntaxe XML. Le langage RDF Schéma (RDFS) est le standard accompagnant RDF qui permet de représenter les connaissances ontologiques relatives aux annotations (Rdf 2000). RDF(S) désigne le couple composé de RDF et RDFS.

4.1.1 Le noyau du modèle

Le modèle de données RDF et la spécification de sa syntaxe est une recommandation du W3C (Rdf 1999). Il est dédié à la description des ressources du Web. Tout objet du Web identifié par une URI est une ressource et peut être écrite en RDF, ce peut aussi bien être un document textuel qu'une image ou encore un document sonore.

Une description RDF consiste en un ensemble de déclarations spécifiant chacune la valeur d'une propriété d'une ressource. Une déclaration RDF est donc un triplet (ressource, propriété, valeur) où la valeur est soit une ressource soit un littéral.

Considérez comme un exemple simple, la phrase : La page index a pour titre ontologie. Cette phrase est composée des parties suivantes (Figure 1.6):

- Sujet (Ressource) : index.html
- Prédicat (Propriété) : Titre
- Objet (valeur) : "ontologie"

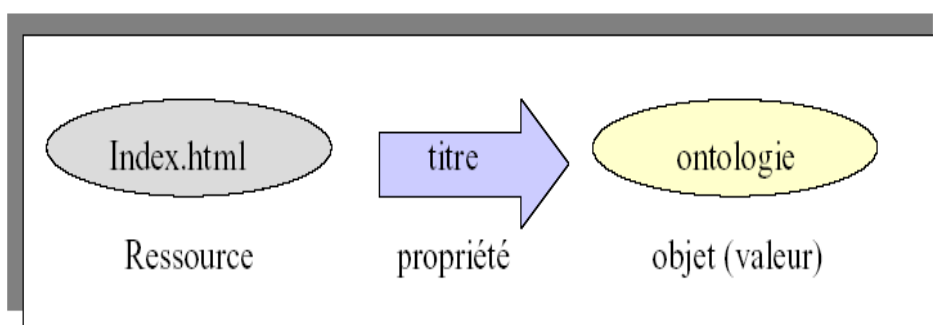


Figure-1.6 : Triplet (index.html, titre, ontologie) (Rdf 1999)

Le triplet (index.html, Titre, "ontologie") est l'élément de base d'un énoncé RDF.

Le modèle de données RDF est proche des réseaux sémantiques. Un ensemble de déclarations RDF peut être vu comme un multi-graphe orienté étiqueté, dont les sommets sont des ressources ou des littéraux et dont les arcs sont étiquetés par les propriétés des ressources.

Le modèle de données RDF est défini indépendamment de sa syntaxe. L'échange de données RDF sur le Web a conduit à associer au modèle une syntaxe XML.

4.2 RDFS

RDF Schéma (RDFS) est une recommandation du W3C en cours d'élaboration. Il s'agit du langage de spécification de schémas associé à RDF. Il est dédié à la représentation des connaissances ontologiques utilisées dans des déclarations RDF.

Un schéma RDF est un ensemble de déclarations de classes et de propriétés. Il s'apparente en cela à la partie terminologique d'une Logique de Description (LDs) où sont déclarés les rôles et les concepts d'une base de connaissances. Il permet en outre, comme dans les langages de Représentation de Connaissances par Objets (RCO), de définir les domaines et les codomaines (domaines de valeurs) des propriétés RDF ainsi que des

hiérarchies de classes. Il permet également de représenter des hiérarchies de propriétés, celles-ci étant considérées sur le même plan que les classes.

Pour représenter les connaissances spécifiques à un domaine, un schéma RDF est défini par raffinement du schéma noyau RDFS. Les classes spécifiques au domaine sont déclarées comme des instances de la classe *Class* et les propriétés spécifiques au domaine comme des instances de la classe *Property*. Les propriétés *subClassOf* et *subPropertyOf* permettent de définir des hiérarchies de classes et des hiérarchies de propriétés.

Les hiérarchies de classes et de propriétés autorisent toutes deux l'héritage multiple. Les propriétés *range* et *domain* permettent de restreindre les domaines et codomaines des ressources de type *Property*. Dans la version (Rdf 2000), la signature d'une propriété autorise un domaine à valeur unique : une propriété peut être appliquée à différentes classes mais sa valeur appartient à une classe unique.

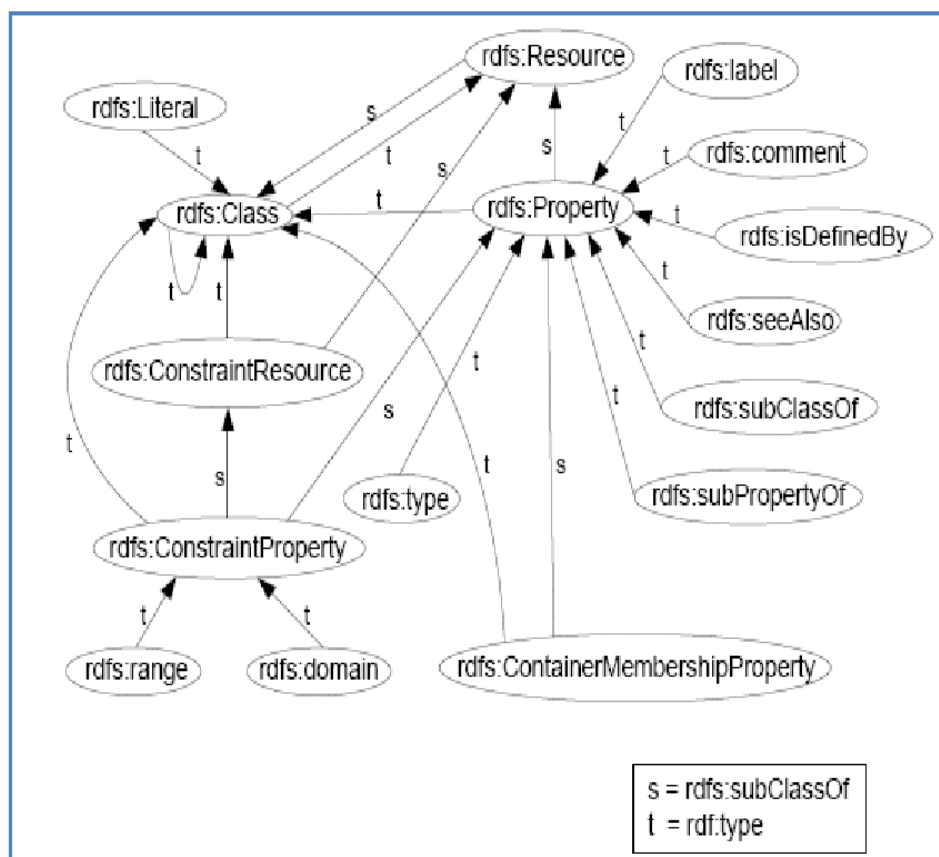


Figure- 1.7. Le schéma de RDF(S) (Rdf 1999)

La figure 1.7 montre les primitives de RDF(S). Les concepts et les relations sont déclarés dans un document RDF(S) comme instances de « Class » et de « Property ».

Une fois ce schéma stocké sur le Web, les primitives qui y sont décrites peuvent être utilisées dans une page si on y inclut une référence à l'URI du schéma. Une application nécessitant l'accès à la sémantique de la page utilisera alors le schéma d'interprétation. Le RDF-S n'est cependant pas un langage opérationnel de représentation, au sens où il ne permet pas la représentation des axiomes et leur utilisation pour raisonner.

4.3 DAML + OIL

Dans l'optique d'une utilisation d'ontologies sur le Web, le langage RDF-S a été enrichi par l'apport du langage OIL (Ontology Interchange Language1) qui permet d'exprimer une sémantique à travers le modèle des frames tout en utilisant la syntaxe de RDF-S. OIL offre de nouvelles primitives permettant de définir des classes à l'aide de mécanismes ensemblistes issus des logiques de description (intersection de classes, union de classes, complémentaire d'une classe). Il permet également d'affiner les propriétés de RDF-S en contraignant la cardinalité ou en restreignant la portée (Fensel et al. 2000).

Le langage OIL a été fusionné avec le langage DAML pour former le DAML+OIL. DAML (Darpa Agent Markup Language2) est conçu pour permettre l'expression d'ontologies dans une extension du langage RDF. Il offre les primitives usuelles d'une représentation à base de frames et utilise la syntaxe RDF (Hendler et McGuinness 2001). L'intégration de OIL rend possible les inférences compatibles avec les logiques de description, essentiellement les calculs de liens de subsomption.

4.4 OWL

La combinaison de RDF/RDF-S et de DAML+OIL a permis l'émergence de OWL (Web Ontology Language), un langage standard de représentation de connaissances pour le Web.

Développé par le groupe de travail sur le Web Sémantique du W3C, OWL peut être utilisé pour représenter explicitement les sens des termes des vocabulaires et les relations entre ces termes. OWL vise également à rendre les ressources sur le Web aisément accessibles aux processus automatisés (Gruber et Olsen 1994), d'une part en les structurant d'une façon compréhensible et standardisée, et d'autre part en leur ajoutant des méta-informations.

Pour cela, OWL a des moyens plus puissants pour exprimer la signification et la sémantique que XML, RDF, et RDF-S. De plus, OWL tient compte de l'aspect diffus des sources de connaissances et permet à l'information d'être recueillie à partir de sources distribuées, notamment en permettant la mise en relation des ontologies et l'importation des informations provenant explicitement d'autres ontologies.

4.4.1 OWL et les autres langages

XML (Bray et al. 2000) fournit une syntaxe pour des documents structurés, mais n'impose aucune contrainte sémantique à la signification des documents. RDF est un modèle de données pour représenter les objets et les relations entre eux, fournissant une sémantique simple pour ce modèle qui peut être représenté dans une syntaxe XML.

RDF-Schéma est un langage de définition de vocabulaire pour la description de propriétés et de classes représentées par des ressources RDF. RDF-S permet de définir des graphes de triplets RDF, avec une sémantique de généralisation/hiéarchisation de ces propriétés et de ces classes.

OWL ajoute du vocabulaire pour la description des propriétés et des classes, des relations entre classes (par exemple disjointness), des cardinalités et des caractéristiques de propriétés (par exemple symmetry). OWL est développé comme une extension du vocabulaire de RDF et il est dérivé du langage d'ontologies DAML + OIL.

4.1.2 Sous langages de OWL

OWL a trois sous langages de plus en plus expressifs : OWL Lite, OWL DL, et OWL Full :

1. *OWL Lite* : Il supporte les utilisateurs ayant besoin principalement d'une hiérarchie de classification et des contraintes simples (un ensemble est limité à 0 ou 1 élément, par exemple). Il a une complexité formelle inférieure à celle d'OWL DL. OWL Lite supporte seulement un sous-ensemble de constructions du langage OWL.
2. *OWL DL* : D'après son nom, OWL DL utilise la logique de description DL. Il a été défini pour les utilisateurs qui réclament une expressivité maximale tout en retenant la complétude informatique (toutes les conclusions sont garanties être calculables), et la possibilité de décision (les calculs finiront en un temps fini). Il inclut toutes les constructions du langage OWL, qui ne peuvent être utilisées que sous certaines restrictions.
3. *OWL Full* : Il a été défini pour les utilisateurs qui veulent une expressivité maximale et une liberté syntaxique de RDF mais sans les garanties informatiques. OWL Full permet à une ontologie d'augmenter la signification du vocabulaire prédéfini (RDF ou OWL). Il est peu probable que n'importe quel logiciel de raisonnement soit capable de supporter le raisonnement complet de chaque caractéristique d'OWL Full. Autrement dit, en utilisant OWL Full en comparaison avec OWL DL, le support de raisonnement est moins prévisible puisque l'implémentation complète d'OWL Full n'existe pas actuellement.

OWL Full et OWL DL maintiennent le même ensemble de constructions d'OWL. La différence se situe dans les restrictions sur l'utilisation de certaines de ses caractéristiques et sur l'utilisation des caractéristiques de RDF. OWL Lite permet le mélange libre d'OWL avec RDFS et, comme RDFS, n'impose pas une séparation stricte des classes, des propriétés, des individus, et des valeurs de données.

OWL Full peut être vu comme étant une extension de RDF, tandis que OWL Lite et OWL DL peuvent être vus comme des extensions d'une vue restreinte de RDF. Alors, les utilisateurs de RDF devraient se rendre compte qu'OWL Lite n'est pas simplement une extension de RDFS. OWL Lite met des contraintes sur l'utilisation du vocabulaire de RDF (par exemple, disjointness des classes, des propriétés, etc.). OWL Full est conçu pour la compatibilité maximale de RDF. Quant à opter pour OWL DL ou OWL Lite, il faut considérer si les avantages du OWL DL/Lite (par exemple, support de raisonnement) l'emportent par rapport aux restrictions de DL/Lite à l'utilisation des constructions de OWL et de RDF.

4.4.3 Syntaxe et sémantique du langage OWL

Pour utiliser *OWL*, deux types de syntaxe sont disponibles. Si la première est évidemment basée sur *RDFS*, la deuxième dite *syntaxe abstraite* ressemble à la syntaxe abstraite utilisée pour les langages des *frames* et les langages des logiques de descriptions (Djaghloul 2007). Elle est plus appropriée à la lecture et à l'interprétation. Dans (Patel-Schneider et Horrocks 2004), on trouve la syntaxe abstraite en détail pour le langage *OWL DL*. Le cas de la sémantique est plus clair. La sémantique du langage *OWL* est très liée à celle des logiques de descriptions.

Table 1.3 présente les constructeurs de classes OWL et leurs correspondances en LDs. C et D sont des classes, T est un type de donnée, n est un nombre, a et b sont des individus, p une propriété d'objet (ObjectProperty) et d une propriété de données (DatatypeProperty).

L'exemple suivant montre une utilisation réelle de *OWL*. On s'intéresse à la restriction transitive sur une propriété de type objet. Rappelons qu'une propriété P est transitive si : P(x,y) et P(y,z) implique P(x,z). Dans notre exemple, la propriété *Situé-Dans* de la classe *Région* est transitive. Alors, on peut proposer le code *OWL* suivant :

```
<owl:ObjectProperty rdf:ID=" Situé-Dans ">
<rdf:type rdf:resource="&owl;TransitiveProperty" />
<rdfs:domain rdf:resource="&owl;Thing" />
<rdfs:range rdf:resource="#Region" />
</owl:ObjectProperty>
```

Table 1.3: Les constructeurs OWL (Web 2011)

nom	syntaxe LD	syntaxe abstraite	sémantique
Nom de classe	C	C (URI)	$C^I \subseteq \Delta^I$
Top, Thing	\top	owl:Thing	$\top^I = \Delta^I$
Bottom, Nothing	\perp	owl:Nothing	$\perp^I = \emptyset$
Intersection	$C \cap D$	intersectionOf($C D$)	$(C \cap D)^I = C^I \cap D^I$
Union	$C \cup D$	unionOf($C D$)	$(C \cup D)^I = C^I \cup D^I$
Négation	$\neg C$	complementOf(C)	$(\neg C)^I = \Delta^I \setminus C^I$
Énumération	$\{a, b, \dots\}$	oneOf($a b \dots$)	$\{a, b, \dots\}^I = \{a^I, b^I, \dots\}$
Quantificateur existentiel	$\exists p.C$	restriction(p someValuesFrom(C))	$(\exists p.C)^I = \{x \exists y, (x, y) \in p^I$ et $y \in C^I\}$
Quantificateur universel	$\forall p.C$	restriction(p allValuesFrom(C))	$(\forall p.C)^I = \{x \forall y, (x, y) \in p^I$ alors $y \in C^I\}$
Restriction à une valeur	$\exists p.a$	restriction(p hasValue(a))	$(\exists p.a)^I = \{x (x, a^I) \in p^I\}$
Restriction non qualifiée de cardinalité	$= n p$	restriction(p cardinality(C))	$(= n p)^I = \{x \text{card}\{y (x, y) \in p^I\} = n\}$
	$\geq n p$	restriction(p minCardinality(C))	$(\geq n p)^I = \{x \text{card}\{y (x, y) \in p^I\} \geq n\}$
	$\leq n p$	restriction(p maxCardinality(C))	$(\leq n p)^I = \{x \text{card}\{y (x, y) \in p^I\} \leq n\}$
Quantificateur existentiel	$\exists d.T$	restriction(d someValuesFrom(T))	$(\exists d.T)^I = \{x \exists y, (x, y) \in d^I$ $\wedge y \in T^D\}$
Quantificateur universel	$\forall d.T$	restriction(d allValuesFrom(T))	$(\forall d.T)^I = \{x \forall y, (x, y) \in d^I$ $\rightarrow y \in T^D\}$
Restriction à une valeur	$\exists d.a$	restriction(d hasValue(a))	$(\exists d.a)^I = \{x (x, a^D) \in d^I\}$
Restriction non qualifiée de cardinalité	$= n d$	restriction(d cardinality(T))	$(= n d)^I = \{x \text{card}\{y (x, y) \in d^I\} = n\}$
	$\geq n d$	restriction(d minCardinality(T))	$(\geq n d)^I = \{x \text{card}\{y (x, y) \in d^I\} \geq n\}$
	$\leq n d$	restriction(d maxCardinality(T))	$(\leq n d)^I = \{x \text{card}\{y (x, y) \in d^I\} \leq n\}$

7. Le Web Sémantique et Web 2

Ces dernières années ont vu la montée en puissance de deux visions du Web, que l'on pourrait à première vue considérer comme disjointes. La première, le Web Sémantique, propose une extension de celui-ci définissant des formalismes de représentations unifiées pour les données dans une optique d'échange et de compréhension de celles-ci par les agents logiciels (Berners-Lee et al. 2001). L'autre, communément appelée Web 2, est beaucoup plus pragmatique et met l'accent sur la place centrale de l'utilisateur au sein de la démarche de production d'information (O'Reilly 2005). Elle met en avant les échanges, l'ouverture et la collaboration entre internautes par l'intermédiaire d'outils et services simples d'utilisation.

De manière générale, la complémentarité entre Web 2 et Web Sémantique est nécessaire pour conduire à un Web où les interactions sociales sont omniprésentes dans un objectif de production de données interprétables et interopérables. Ainsi, le Web Sémantique et ses formalismes de représentation des connaissances ne s'opposaient pas – bien au contraire – à l'utilisation d'outils et de principes Web 2 (Passant 2009).

7.1 Apports du Web 2 pour le Web Sémantique

Si l'on se base sur la vision du Web 2 en tant que système centré sur l'utilisateur, il nous semble important pour le Web Sémantique de réutiliser certains paradigmes de celui-ci afin de monter en puissance (Passant 2009):

- l'utilisation d'outils simples pour la production à grande échelle de données formalisées selon les principes du Web Sémantique, publiées de manière personnelle ou collaborative.
- la masse importante d'utilisateurs passés du statut de consommateur à celui d'acteur. Si le Web 2 est en effet un read-write Web, qui plus est collaboratif, les outils du Web Sémantique peuvent ainsi bénéficier d'une masse importante d'utilisateurs producteurs de données formalisées, pour autant que les outils soient simples et adaptés;
- la collaboration entre utilisateurs à des fins de création collective et consensuelle d'informations et de connaissances, en corollaire des éléments précédents. Ainsi, les wikis (un wiki est un site Web dynamique et évolutif, au sens où il permet à chaque lecteur de modifier les pages consultées et d'en ajouter de nouvelles mais aussi d'en supprimer) peuvent être utilisés pour peupler ou maintenir des ontologies de manière collaborative;

- l'utilisation d'interfaces simples et ergonomiques, pour la visualisation et la navigation de graphes complexes d'annotations sémantiques. Si ces structures de données sont relativement complexes, l'utilisation d'outils comme des interfaces à facettes permettrons de masquer cette complexité aux utilisateurs;

7.2 Apports du Web Sémantique pour le Web 2

Réciproquement, si les outils du Web 2.0 proposent des méthodes qui nous semblent efficaces en termes d'usages et d'interfaces, nous pensons que l'utilisation des technologies du Web Sémantique ne peut être qu'un plus en termes de structuration et d'échange de données sur le Web 2 (Passant 2009).

En se basant sur les outils et langages du Web Sémantique, les outils du Web 2 peuvent ainsi profiter de (Figure 1.8) :

- l'utilisation de modèles communs pour représenter leurs métadonnées, logiquement basés sur RDF, en lieu et place d'APIs hétérogènes. SIOC (Breslin et al. 2005) répond à cette problématique en proposant un vocabulaire pour définir les métadonnées du Web 2 (notion de billet, d'utilisateur, de commentaire ...) via les technologies du Web Sémantique. Il permet ainsi de ne plus les services en ligne comme des îlots indépendants et isolés sur le Web, mais comme des services interconnectés où l'échange d'informations peut se faire de manière transparente;
- l'utilisation d'ontologies métier pour permettre la structuration des connaissances produites via ces outils. L'utilisation d'ontologies du domaine doit permettre de capitaliser des connaissances de manière formelle à des fins de réutilisation entre services. En ce sens, les wikis sémantiques nous semblent un bon exemple d'utilisation de technologies du Web Sémantique pour augmenter le potentiel d'outils existants et déjà bien acceptés sur le Web 2;
- l'utilisation de protocoles de requêtes et d'échange standardisés. L'utilisation de RDF pour la production de données et de SPARQL (**SPARQL** est un langage de requête adapté à la structure spécifique des graphes RDF) pour leur interrogation permet ainsi de simplifier l'interopérabilité entre applications. On favorise en ce sens la découverte de contenus répartis sur différents services Web 2.

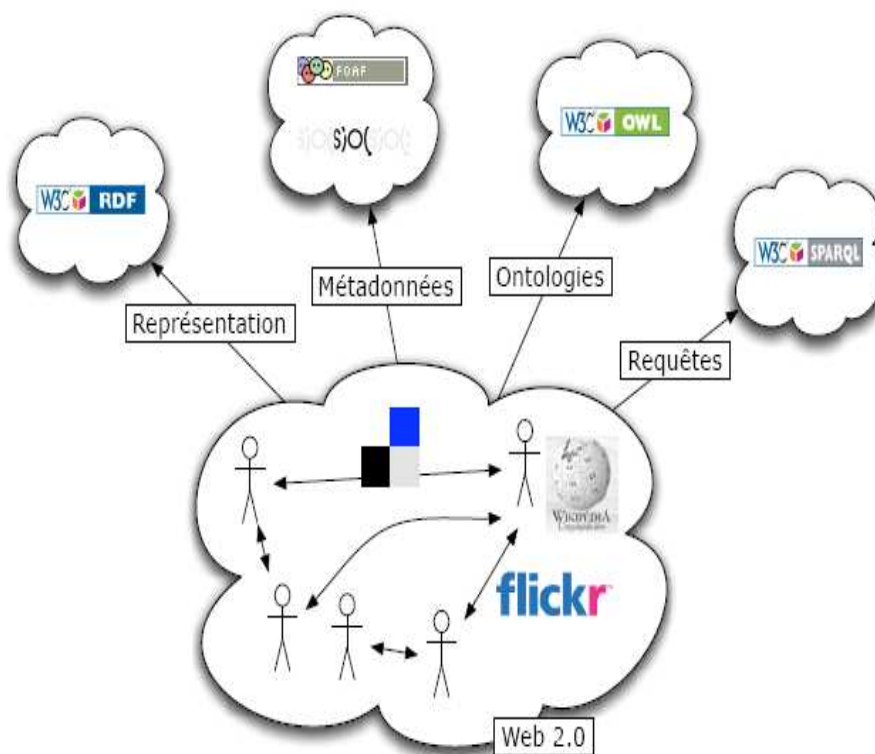


Figure- 1.8. Web Sémantique pour le Web 2 (Passant 2009)

Ainsi, les outils du Web 2 peuvent bénéficier des technologies du Web Sémantique pour assurer la structuration et l'homogénéité des données produites : en s'affranchissant des formats internes et d'APIs propriétaires, on facilite les échanges entre et depuis des systèmes originellement hétérogènes. En conséquence, les outils du Web 2 peuvent également participer à cet essor du Web of Data, en produisant non plus de simples documents, mais un ensemble de données interopérables (Passant 2009).

8. Conclusion

Nous avons décrit dans ce chapitre le cadre technique de notre travail. Nous avons présenté les concepts de base des ontologies et du web sémantique, qui sont la base de notre travail. Voici un résumé des principaux points que nous avons abordés:

- Le Web sémantique transforme le contenu du Web pour qu'il soit compréhensible par la machine. Pour réaliser cette tâche, il exploite la notion d'ontologie.

- Une ontologie est une spécification explicite et formelle d'une conceptualisation partagée ;
- Les ontologies sont considérées comme des objets techniques évolutifs possédant un cycle de vie ;
- Dans l'état de l'art sur les formalismes de représentation d'ontologies, nous avons souligné la grande expressivité des logiques de descriptions. Dans la suite de ce travail nous allons exploiter ces logiques pour la description des ontologies à intégrer.
- Une partie importante de ce chapitre a été dédiée au langage *OWL*, langage que nous avons choisi pour l'implantation des ontologies développées dans le cadre de cette thèse. Ce choix est motivé par les possibilités offertes par ce langage, à savoir un fondement théorique consistant en se basant sur les logiques descriptions et le support d'une syntaxe utilisable dans le cadre du Web.
- Les Graphes conceptuels sont un formalisme de notation graphique de la logique dans lequel le raisonnement peut se faire par projection de graphes. Donc, nous allons exploiter ce formalisme pour décrire l'ontologie de l'alignement pour sa lisibilité. En plus, l'ontologie de l'alignement générée par notre système représente des cas simples dont on n'a pas besoin des constructeurs.

Après ce panorama des technologies qui constituent le paysage technique de ce mémoire, le chapitre suivant se focalise sur les travaux de recherche autour de l'intégration d'ontologies.

Intégration des ontologies

1. Introduction

Les ontologies jouent un rôle clé dans le domaine du web sémantique notamment dans l'intégration des systèmes d'information et dans la recherche de l'information. Comme nous l'avons souligné dans le chapitre précédent, elles permettent de représenter les connaissances d'une manière formelle et explicite. Elles constituent alors la brique supportant les échanges et le partage des informations en étendant l'interopérabilité syntaxique du web en une interopérabilité sémantique.

Le web sémantique exploite les ontologies dans la représentation des connaissances (Charlet et al. 2005). Partant du fait qu'il existe plusieurs connaissances qui peuvent prendre des représentations différentes, nous trouvons de nos jours plusieurs ontologies de domaine pour un même champ d'application (Zghal et Ben Yahia 2007). Ainsi, La mise en correspondance (mapping) des ontologies, l'alignement et la fusion des ontologies devient un grand défi pour établir un lien entre plusieurs sources d'information pour échanger des informations d'une source à l'autre.

Dans ce chapitre, nous commençons par évoquer le problème de l'intégration des ontologies. En suite, nous présentons les hétérogénéités qui peuvent exister entre les différentes ontologies et leurs impacts sur le processus de l'intégration. Après, nous définissons le calcul des mesures de similarité entre les entités sémantiques. Puis, Nous passons à la présentation en détail des types d'intégration des ontologies, nous les illustrons par des exemples d'application. Le chapitre se termine par une conclusion.

2. Intégration 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).

Il est généralement admis que la démarche d'intégration d'ontologies est basée sur trois sous-processus (Kavouras 2003): le processus d'extraction de la sémantique, le processus de comparaison et enfin le processus d'intégration proprement dit (figure 2.1).

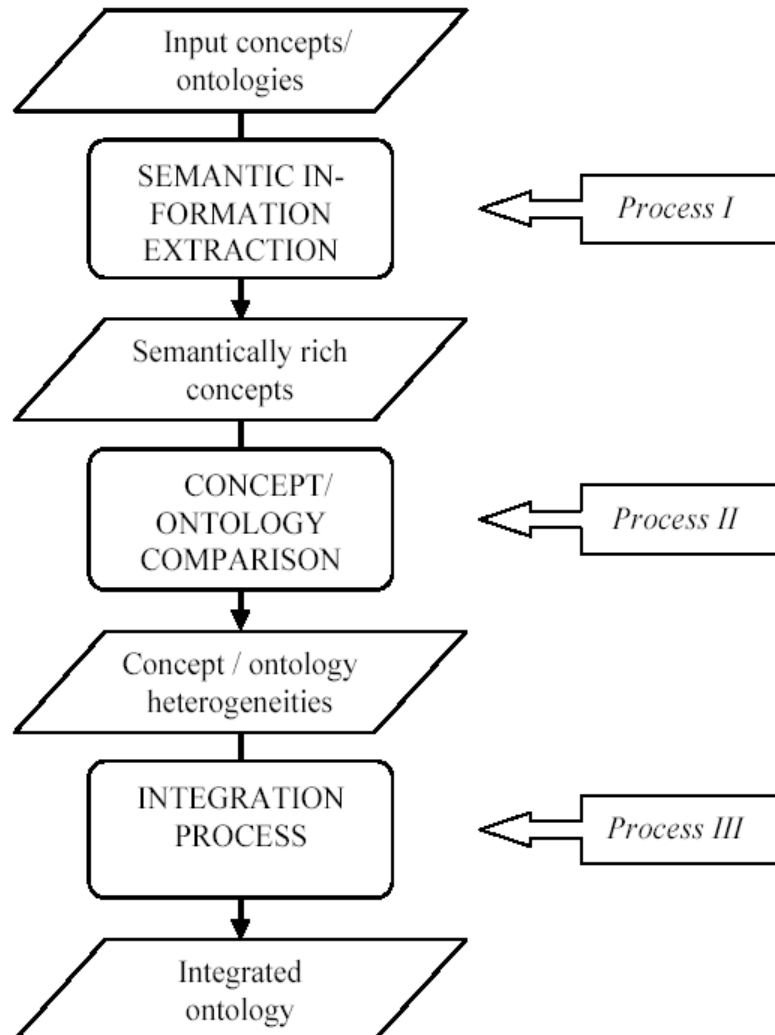


Figure2.1- Méthodologie d'intégration d'ontologies (Kavouras 2003)

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. Et enfin, le dernier processus permet d'intégrer les ontologies concernées.

Le problème d'intégration des ontologies est la conséquence logique de l'utilisation des architectures se basant sur plusieurs ontologies. Plusieurs travaux existent qui étudient cet aspect et qui proposent des solutions pour résoudre les différentes hétérogénéités qui peuvent exister au niveau des ontologies (Saïd 2006).

Dans la section suivante nous allons étudier les hétérogénéités d'ontologies et nous allons ensuite présenter les solutions typiques utilisées pour leur résolution.

2.1 Hétérogénéité des ontologies

Le projet Knowledge Web (Knowledge 2004) 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.

2.1.1 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 utilisées 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) qui constitue un cas particulier du problème d'intégration d'ontologies. Ce problème est généralement résolu de façon relativement simple par la mise en œuvre 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.

2.1.2. 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 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.

- Des variations syntaxiques d'un même mot (abréviations, etc.).

Comme le précise (Knowledge 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 plu part des hétérogénéités qui occurrent dans la réalité se situent à ce niveau. Ce qui rend ce dernier assez critique.

2.1.3. 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 2.2) :

- 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. Par exemple: on peut définir deux ontologies décrivant les composants microélectroniques avec deux points de vue différents: l'un concerne l'aspect fabrication et l'autre concerne l'aspect marketing.

2.1.4 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 ce qui suit nous présentons en détail les différents types d'intégration des ontologies.

2.2 Différents types d'intégration des ontologies

Un certain nombre de techniques ont été proposées dans la littérature pour réaliser l'intégration des ontologies. 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.

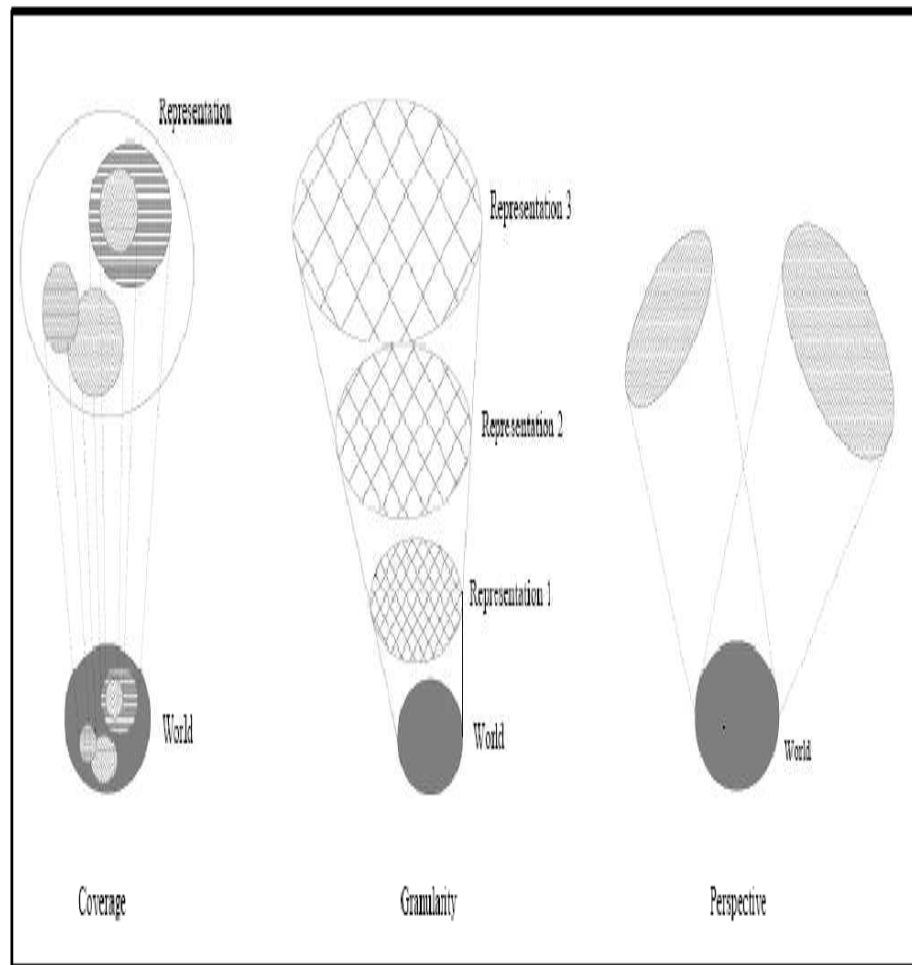


Figure 2.2- Trois dimensions d'hétérogénéités conceptuelles (Knowledge 2004)

Dans l'ouvrage "Semantic Web Technologies" (Bruijn et al. 2006), l'auteur distingue trois catégories principales qui sont :

1. 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.

2. L'alignement d'ontologies, pour qui l'objectif consiste à découvrir des correspondances entre les ontologies. L'ensemble des mappings représente l'alignement.
3. 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.

Ces trois techniques seront détaillées dans les sections suivantes.

2.2.1. La mise en correspondance (mapping) des ontologies

La mise en correspondance entre les entités sémantiques des ontologies représente le cœur de l'intégration de ces ontologies, elle peut se situer au niveau sémantique et au niveau conceptuel. Elle permet d'établir des liens sémantiques entre les entités des ontologies à intégrer pour échanger les informations des sources représentées par ces ontologies. Les travaux existant pour établir les relations de mise en correspondance se focalisent à la fois sur les aspects d'automatisation et de découverte de ces relations et aussi sur l'aspect formalisation pour le raisonnement.

Le point central aux approches visant l'automatisation est l'utilisation d'heuristiques permettant d'identifier des similarités syntaxiques et structurelles. L'absence de structure commune, de similarités syntaxiques et d'instances communes rend le problème de mise en correspondance difficile et souvent basé sur de fortes hypothèses pas toujours très réalistes. Le problème principal auquel est confronté le processus de mise en correspondance est lié à l'hétérogénéité sémantique introduite quand différentes ontologies sont construites avec différentes terminologies pour représenter la même information dans les contextes relatifs aux différentes sources (Bennacer et Aufaure 2006).

Selon (Mellal 2007), trois phases principales peuvent être distinguées dans le processus de mapping (voir la figure 2.3) :

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

a. 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 (Knowledge 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).

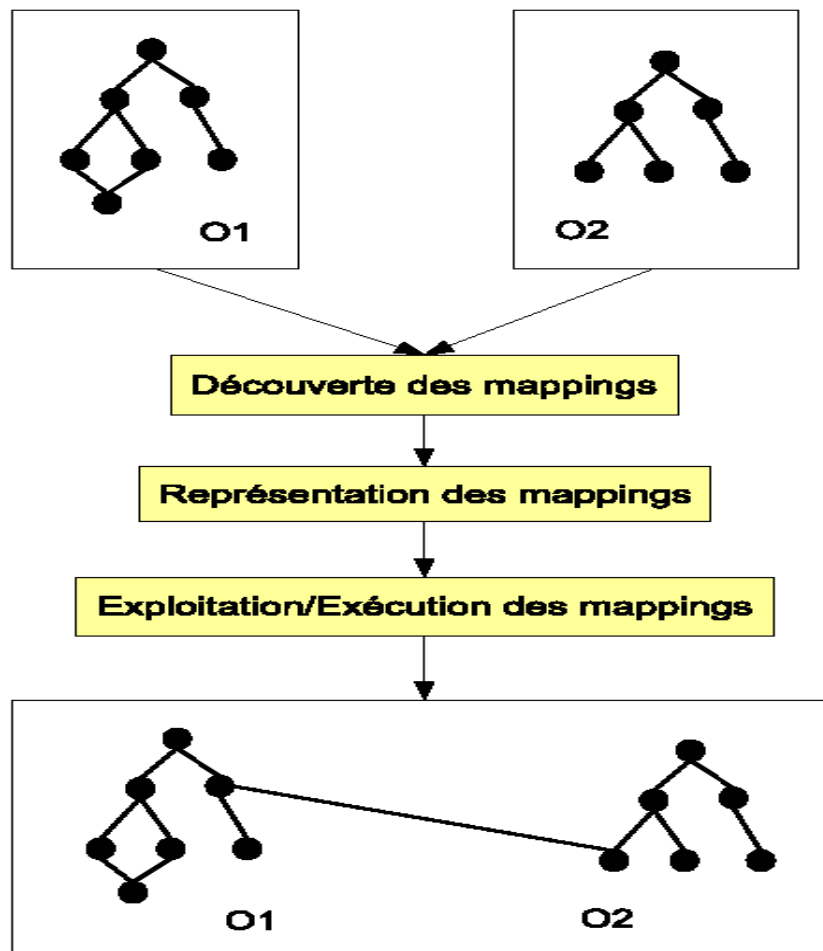


Figure 2.3 – Le mapping des ontologies (Mellal 2007)

Du point de vue abstrait, (Knowledge 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, (Knowledge 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 2.4) :

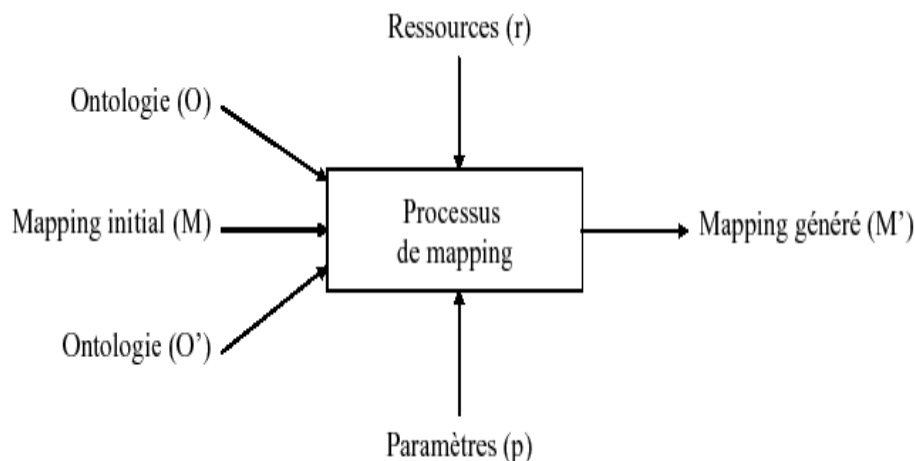


Figure 2.4- Processus générique de mapping (adapté de (Knowledge 2004))

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

$$M' = f(O_1, O_2, \dots, O_n, M, p, r)$$

Le mapping initial M peut être (s'il n'est pas vide) un mapping injecté (de façon très souvent manuelle) 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.

b. Mesures de similarité

La mise en correspondance repose sur le calcul des mesures de similarité entre les entités sémantiques, pour identifier pour chaque entité d'une ontologie, des entités qui lui sont les plus similaires dans les autres ontologies. D'après (Rahm et Bernstein 2001), les différentes mesures de similarité sont organisées comme suit:

La première méthode se base sur la terminologie des entités. En effet elle consiste à comparer les labels des entités. Elle est décomposée en approches purement syntaxiques et celles utilisant un lexique. L'approche syntaxique effectue la correspondance à travers les mesures de dissimilarité des chaînes (EditDistance). Tandis que, l'approche lexicale effectue la correspondance à travers les relations lexicales (synonymie, hyponymie, etc.).

La deuxième méthode s'appuie sur la comparaison des structures internes des entités (intervalle de valeur, cardinalité d'attributs, etc.).

Contrairement à la deuxième méthode, la troisième méthode permet la comparaison des structures externes. Elle compare les relations d'entités avec d'autres. Elle est décomposée en méthodes de comparaison des entités au sein de leurs taxinomies et méthodes de comparaison des structures externes en tenant compte des cycles.

La quatrième méthode des mesures de similarité compare les extensions des entités (instances des classes).

Finalement la cinquième méthode compare les interprétations (ou plus exactement les modèles) des entités.

c. Méthodologies du mapping

Les outils et méthodologies, parmi les plus significatifs dans cette catégorie, sont MAFRA (Maedche et al. 2002), IF-Map (Kalfoglou et Schorlemmer 2003), RDFT (Omelayenko 2003), C-OWL (Bouquet et al. 2004) et OntoMap (Angele et Schnurr 2005).

- MAFRA

MAFRA (MApping FRAmework) est une plate-forme pour le mapping d'ontologies distribuées. Son approche repose sur la notion fondamentale de Passerelle qui permet de formaliser les mappings en établissant des correspondances entre les entités d'une ontologie source et celles d'une ontologie cible.

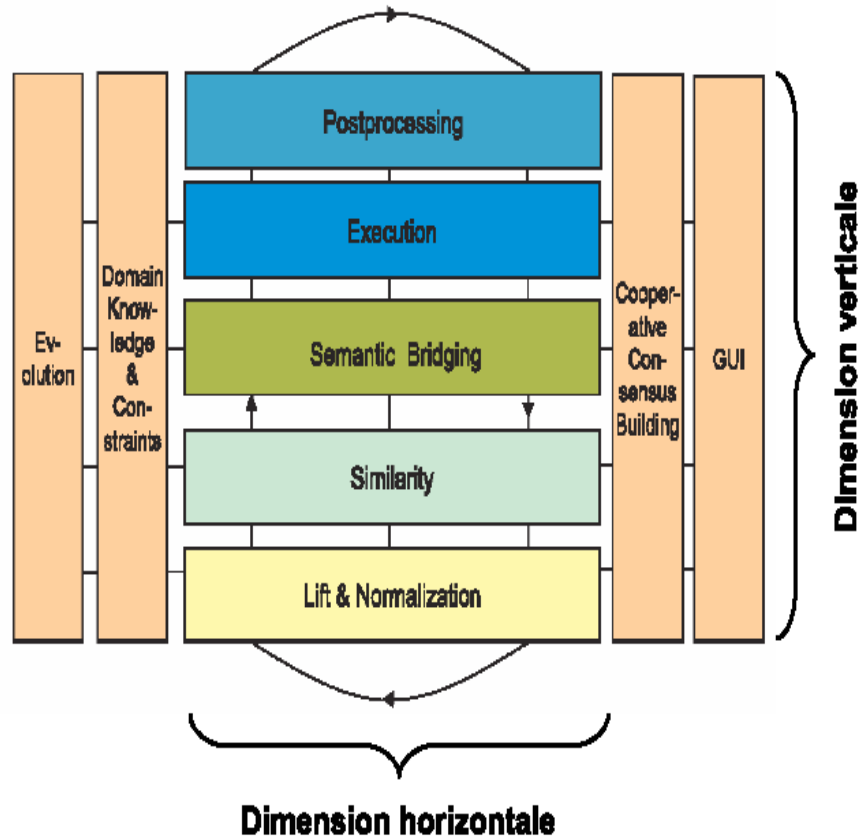


Figure 2.5 – L'architecture conceptuelle de MAFRA selon (Maedche et al. 2002)

L'architecture conceptuelle de MAFRA distingue la dimension horizontale et la dimension verticale (Maedche et al. 2002) (voir la figure 2.5).

La *dimension horizontale* est caractérisée par cinq modules. Le module de normalisation où les ontologies sont représentées de manière uniforme. Le module de similarités établit les ressemblances entre l'ontologie source et l'ontologie cible. Le module du "bridging" sémantique, basé sur les similarités, vise à établir les correspondances entre les entités appartenant aux ontologies sources et cibles en utilisant des heuristiques. Ce module spécifie des passerelles entre entités pour lesquelles chaque instance représentée dans l'ontologie source est transformée en l'instance la plus similaire dans l'ontologie cible. Ce module prend en compte les entités (concepts, relations et instances), la cardinalité (1:1, 1:n et n:1), la structure (spécialisation, alternative, composition, abstraction), les contraintes et les transformations.

Le module Exécution transforme les instances à partir de l'ontologie source vers l'ontologie cible en évaluant les passerelles sémantiques. Enfin, le module Post-Processing

recupère le résultat d'exécution afin de vérifier et d'améliorer la qualité des résultats de transformation.

La dimension verticale est également divisée en sous modules. Le module d'Evolution vise à préserver les passerelles sémantiques générées dans le module bridging sémantique. Celui de Construction d'un consensus coopératif consiste à établir un consensus sur les passerelles sémantiques entre deux communautés participant au processus du mapping. Le module Domaine de connaissances et contraintes est utilisé pour améliorer les passerelles sémantique. Enfin, le module Interface graphique pour l'utilisateur (GUI) est toujours recommandé pour masquer la difficulté des conceptions complexes.

En résumé, l'établissement des mappings dans la plateforme MAFRA repose essentiellement sur la génération de passerelles, lesquelles établissent des correspondances entre les entités de différentes ontologies.

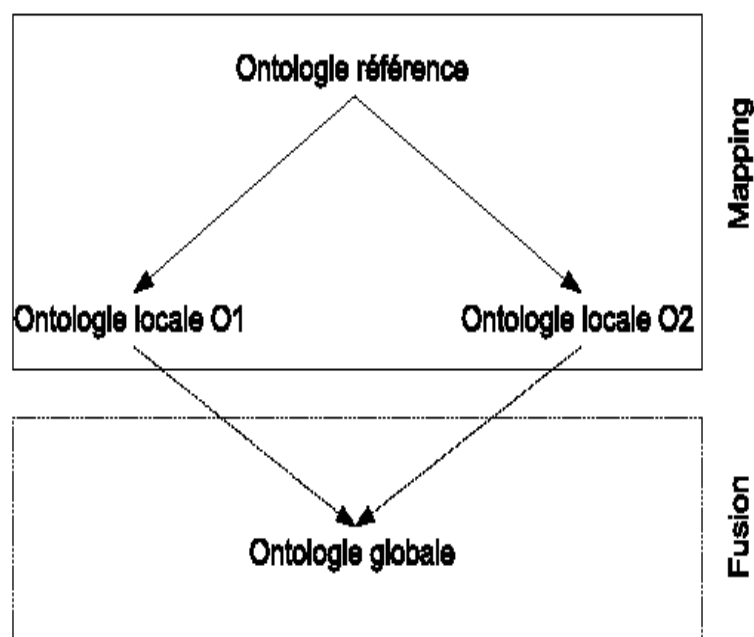


Figure 2.6 – Le mapping entre deux ontologies locales (Kalfoglou et Schorlemmer 2003)

- IF-Map

IF-Map (Information Flow Mapping) est une méthodologie proposée par Schorlemmer et Kalfoglou réalisant des mappings entre différentes ontologies de domaine (Kalfoglou et Schorlemmer 2003). IF-Map repose sur la théorie de Barwise "The logic of Information Flow", noté IF Model (Barwise et Seligman 1997).

Les auteurs étudient la théorie des flux d'information qui fournit une description théorique consistante du processus partiel d'intégration sémantique effectué par deux agents. Pour cela ceux-ci réalisent un alignement progressif des ontologies à l'aide des instances de celles-ci. Leur approche est basée sur une mise en correspondance d'ontologies locales. Deux autres ontologies sont utilisées dans le processus de mapping : l'une que l'on appelle *ontologie de référence* et l'autre, *ontologie globale* " (voir la figure 2.6).

Ils supposent que les ontologies locales sont utilisées par différentes communautés et sont peuplées par des instances de leur domaine respectif, alors que l'ontologie de référence est une interprétation commune dédiée au partage de la connaissance mais privée d'instances. L'ontologie globale est une ontologie virtuelle créée au vol pour les besoins de la fusion. Les relations entre les ontologies locales et l'ontologie virtuelle sont du type infomorphismes logiques.

Basées sur le modèle *IF*, les ontologies sont exprimées en termes de logiques locales. Les mappings sont alors formalisés par les infomorphismes. Le processus de mapping de l'approche IF-Map (Kalfoglou et Schorlemmer 2003) inclut quatre étapes principales représentées à la figure 2.7.

- *L'acquisition des ontologies* (Ontology Harvesting) qui utilise des ontologies existantes, en les téléchargeant à partir du Web, ou en les créant.
- *La traduction* d'un ensemble d'ontologies d'entrée qui peut se faire en clauses de Horn puisque IF-Map est spécifiée dans cette logique. Par exemple, une ontologie générée en RDF peut être traduite en Prolog.
- *La génération des infomorphismes* (Infomorphisms Generation) qui construit des liens formels entre les éléments similaires de deux différentes ontologies.
- *L'affichage des mappings* (Display Mappings) qui traduit les infomorphismes en syntaxe RDF et les stocke dans une base de connaissances pour une utilisation ultérieure par d'autres applications.

- **RDFT**

Omelayenko a proposé RDFT (*Resource Description Framework Transformation*) pour résoudre les mappings entre méta-ontologies (Omelayenko 2003) en spécifiant un langage entre des ontologies exprimées au moyen de DTDs XML et de schémas RDF.

RDFT dispose d'une classe de base appelée " *BRIDGE* " ou " Passerelle ". Cette classe vise à connecter des concepts et à spécifier leurs correspondances selon le type de passerelle. Comme le montre la figure 2.7, deux types de passerelles sont requis, les passerelles *Un-à-plusieurs* (one-to-many) qui détaillent les correspondances entre une seule entité et un ensemble d'entités et les passerelles *Plusieurs-à-un* (many-to-one) dédiées aux

correspondances inverses. RDFT se rapproche de MAFRA en ce qui concerne l'idée d'utiliser les passerelles pour le mapping d'ontologies.

The IF-Map Architecture

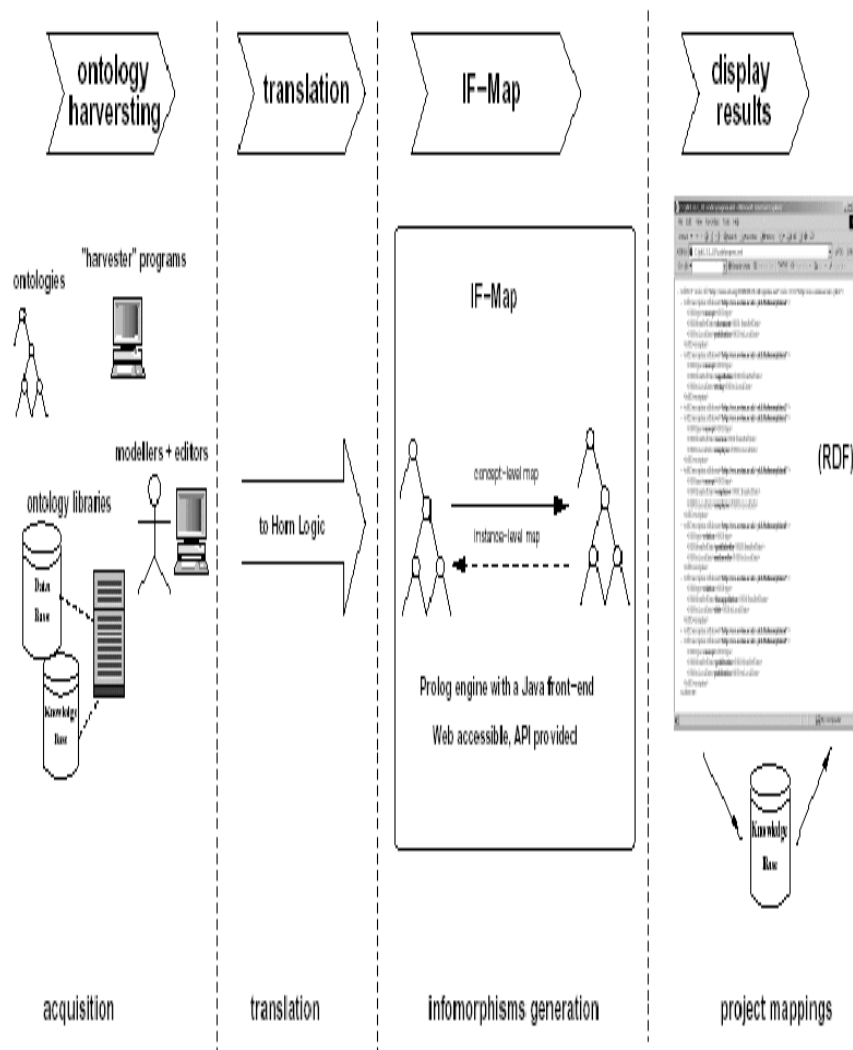


Figure 2.7 – L'architecture de IF-Map (Kalfoglou et Schorlemmer 2003)

La relation entre les composants source et cible de la passerelle peut être une relation d'équivalence ou encore une relation de version qui déclare que les éléments de l'ensemble cible forment une version postérieure des éléments de l'ensemble source dans l'hypothèse où les domaines sont identiques pour les deux ensembles.

- C-OWL

C-OWL " Contextual Ontology Web Language " est une extension de OWL 6 qui a été proposée pour la représentation d'ontologies contextualisées (Bouquet et al. 2004). Ces dernières sont des représentations locales, appelées contextes. Les liens entre les contextes sont représentés en relation avec d'autres contextes au travers des mappings.

Deux notions principales caractérisent l'approche C-OWL : l'espace de contextes et les passerelles; Les connaissances sont contenues dans un ensemble de contextes, appelé espace de contextes où chaque contexte est exprimé par une ontologie décrite en OWL possédant son propre langage et sa propre interprétation.

Les mappings entre ontologies s'expriment sous la forme de passerelles. Une passerelle entre un contexte, représenté par une ontologie OWL O_i , et un autre contexte, représenté par une autre ontologie OWL O_j , permet de déclarer une correspondance entre les éléments de connaissances de ces deux contextes. Sur la base de ces correspondances, une partie des connaissances contenues dans O_i peut être interprétée et réutilisée dans O_j .

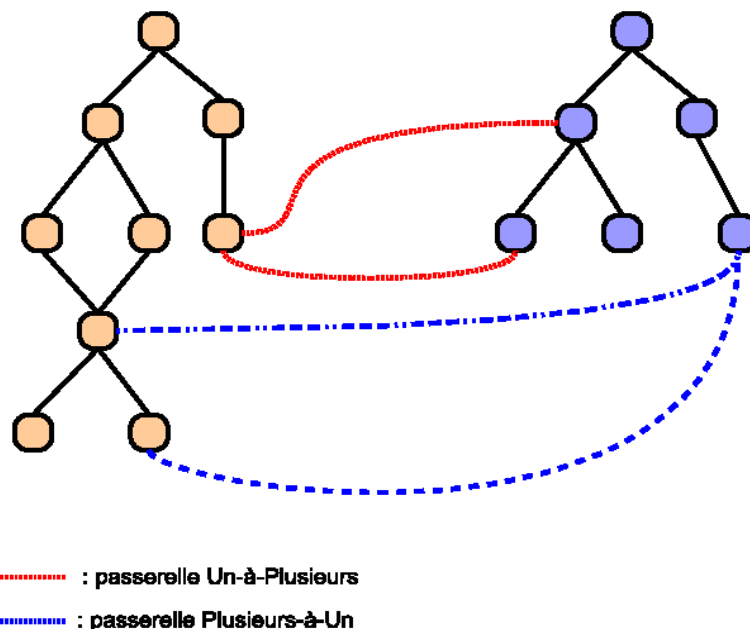


Figure 2.8 – Les passerelles (Mellal 2007)

Selon les auteurs (Bouquet et al. 2004), C-OWL permet l'exploitation de mécanismes de raisonnements globaux, s'appuyant sur les passerelles (figure 2.8). Ainsi, le calcul de la subsomption globale repose sur ce qui est appelé le principe de propagation de la subsomption : la subsomption dans un contexte O_j peut être déduite de la subsomption dans un contexte O_i et de passerelles de O_i à O_j .

L'instanciation globale repose sur le principe de propagation de l'instanciation, fortement inspiré du principe de propagation de la subsomption. Il est possible d'en déduire qu'un individu est une instance d'une classe dans un contexte en s'appuyant sur l'instanciation dans un autre contexte et sur les passerelles. Un mapping en C-OWL est défini par un ensemble de passerelles entre deux ontologies. L'ensemble des ontologies OWL et de leurs mappings constituent l'espace de contextes.

- **OntoMap**

OntoMap représente un "plugin" dans la plate-forme OntoStudio. Il permet la création et la gestion des mappings d'ontologies. Ceux-ci sont accessibles par une représentation graphique et avec l'environnement *Schema-View* dédié à l'ontologie (Angele et Schnurr 2005). OntoMap associe des phrases formelles à la déclaration des mappings limitant ainsi la tâche des utilisateurs à la compréhension de la sémantique de représentations graphiques (par exemple, une flèche qui connecte deux concepts). Les utilisateurs de OntoMap disposent également de la fonctionnalité "Drag and Drop" et peuvent aussi vérifier la consistance des propriétés.

OntoMap supporte un ensemble de patterns de mapping tels que concept-à-concept, attribut-à-attribut, relation-à-relation, et attribut-à-concept.

2.2.2 L'alignement des ontologies

L'alignement des ontologies permet d'établir des liens sémantiques entre les concepts et des relations inter-ontologies.

L'alignement d'ontologies est un processus de découverte des correspondances entre deux ontologies source. Il est, généralement, décrit comme une application de l'opérateur MATCH (appariement) (Rahm et Bernstein 2001), dont l'entrée est constituée d'un ensemble d'ontologies et la sortie, formée des correspondances entre ces ontologies (voir la figure 2.9).

On trouve dans la littérature plusieurs algorithmes qui implémentent cet opérateur. Ces algorithmes sont généralement regroupés en quatre classes. On trouve le matching basé-schéma, le matching basé-instance, le matching niveau-élément et le matching niveau-structure. Un matcher basé-schéma prend en compte différents aspects des concepts et des

relations dans les ontologies et utilise des mesures de similarité pour déterminer la correspondance.

Les instances appartenant aux concepts des différentes ontologies sont comparées pour découvrir les similarités entre les concepts. Dans le matching niveau-élément et le matching niveau-structure, le " matcher niveau-élément " compare les propriétés d'un concept particulier et d'une relation particulière (par exemple pour trouver des similarités à partir du nom). Le " matcher niveau-structure " compare la structure, c'est-à-dire la hiérarchie du concept pour trouver les similarités (Noy et Mussem 2001) (Giunchiglia et al. 2005). Selon Ehrig, ces " matchers " peuvent également être combinés (Ehrig et Staab 2004).

Les techniques d'alignement jouent un rôle crucial dans la construction d'un lien sémantique entre les ontologies d'un même domaine. Quelques approches d'alignement (Aleksovski et al. 2006), (Stuckenschmidt et al. 2004) et (Hage et al. 2005) considèrent que l'utilisation d'une connaissance sur le domaine est une manière assurant la correspondance sémantique entre la dissimilarité syntaxique des ontologies. L'obtention de la bonne connaissance sur le domaine est primordiale.

D'autres approches n'exploitent pas une connaissance sur le domaine et ne réalisent pas un modèle sémantique formel pour l'alignement des structures produites. Dans ce cas, la structure obtenue est difficile à exploiter pour répondre aux requêtes interrogeant les ontologies (Lopez et al. 2006). En outre, les approches courantes d'alignement d'ontologies (Noy 2004), (Shvaiko et Euzenat 2005) et (Bach 2006) sont basées sur les mesures de similarité entre chaînes de caractères et des structures composites. Les ontologies à aligner peuvent être représentées avec différents langages.

L'alignement suffit dans le cas de l'utilisation d'ontologies portant sur des domaines de connaissance complémentaires, ou sur des domaines de niveaux sémantiques différents. Par exemple, l'utilisation, dans un même système, d'une ontologie de haut niveau et d'une ontologie de domaine ne va nécessiter qu'une compatibilité entre les deux. La compatibilité de deux ontologies est assurée par l'utilisation des mêmes formalismes de représentation, ou l'utilisation de formalismes compatibles, mais également par la compatibilité des modèles de connaissance utilisés (Maedche et al. 2002).

Parmi les approches d'alignement d'ontologies, on trouve Anchor-PROMPT (Noy et Mussem 2001), GLUE (Doan et al. 2003), (Giunchiglia et al. 2005), QOM (Ehrig et Staab 2004) et ASCO (Thanh et al. 2004):

- **Anchor-PROMPT**

Anchor-PROMPT est un algorithme qui a été développé par Noy et Mussem (Noy et Mussem 2001) afin de découvrir automatiquement les termes sémantiquement similaires. Anchor-PROMPT traite une ontologie comme un graphe, les noeuds de ce graphe représentant les classes et les arcs représentant les propriétés. L'algorithme utilise deux

paires de termes relatifs comme entrée. Il analyse les chemins dans le sous-graphe délimité par des ancres et détermine quelles sont les classes qui apparaissent fréquemment dans des positions similaires sur des chemins similaires.

L'algorithme cherche alors des termes le long des chemins qui pourraient être similaires aux termes d'autres chemins. Ces nouveaux termes relatifs sont identifiés par une similarité qui peut être modifiée pendant l'évaluation d'autres chemins dans lesquels ces termes apparaissent. Les termes qui sont fortement semblables sont présentés à l'utilisateur pour améliorer l'ensemble des suggestions possibles.

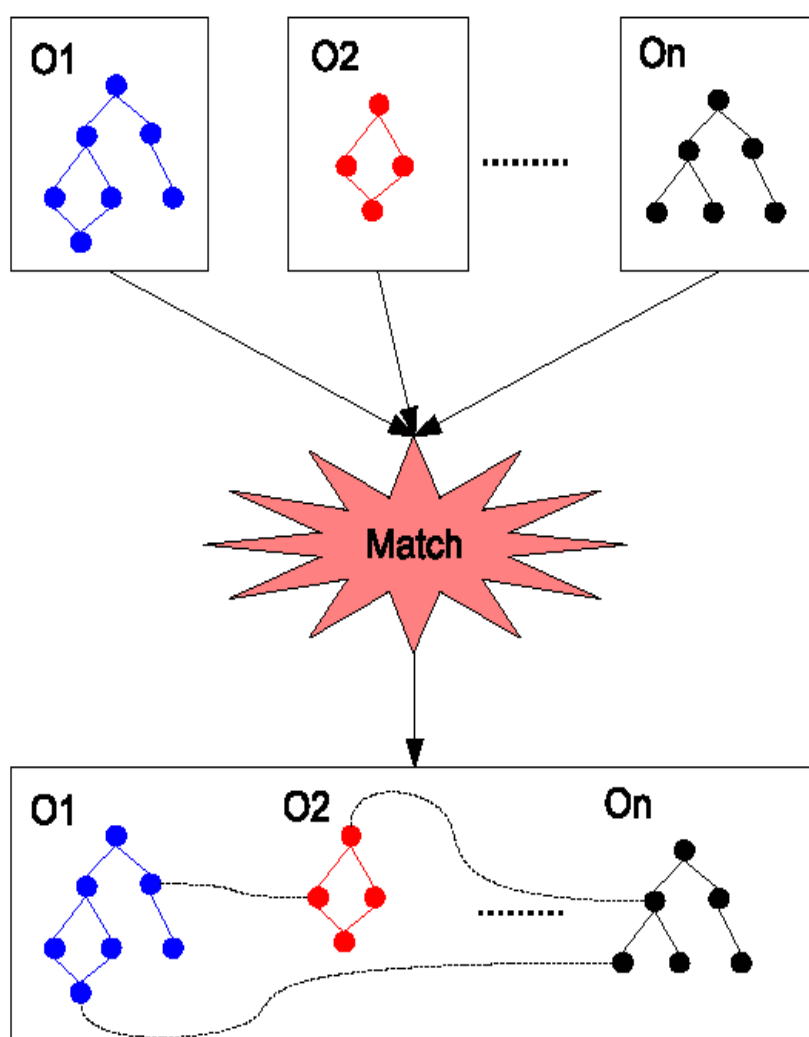


Figure 2.9 – L'opérateur MATCH (Mellal 2007)

- **GLUE**

GLUE est un système qui emploie une approche de type " machine learning " pour créer des mappings semi-automatiques entre ontologies hétérogènes. Il se base sur des données d'instances, une ontologie étant vue comme une taxinomie de concepts (Doan et al. 2003). GLUE se focalise sur la détermination des mapping de type 1-à-1. La similarité de deux concepts A et B dans deux taxinomies O_1 et O_2 utilise l'ensemble des instances des deux concepts qui convergent. Pour déterminer si une instance du concept B est également une instance du concept A, un classifieur est d'abord construit en utilisant les instances de A comme ensemble d'apprentissage.

Ce classifieur est ensuite utilisé à son tour pour traiter les instances de B. Le classifieur décide alors pour chaque instance de B, s'il est également une instance de A ou non. Avec ces classifications, quatre probabilités sont calculées : $P(A|B)$, $P(B|A)$, $P(A,B)$ et $P(\bar{A},\bar{B})$. La probabilité $P(A,B)$ par exemple, s'interprète par l'appartenance de l'instance du domaine à A et la non appartenance à B. Ces quatre cas peuvent ensuite être employées comme paramètres pour calculer la distribution de probabilité commune pour les concepts A et B, laquelle est une fonction écrite par l'utilisateur.

- **S-Match**

S-Match est une approche pour le matching des hiérarchies de classifications (Giunchiglia et al. 2005). Les auteurs mettent en oeuvre un opérateur de correspondance qui admet en entrée deux structures sous forme de graphes (par exemple, schémas de base de données ou ontologies) et produit un mapping entre les éléments qui sont en correspondance sémantique.

Selon Giunchiglia et Shvaiko, presque toutes les anciennes approches utilisant les schémas et le matching d'ontologies sont des approches de matching syntaxiques par opposition au matching sémantique. Dans le matching syntaxique, les labels et parfois la structure syntaxique du graphe sont associés. Un certain coefficient de similarité exprimé dans l'intervalle [0.1] est obtenu. Il indique la similarité entre deux noeuds du graphe.

Le matching sémantique calcule une relation, de nature ensembliste, entre les noeuds en tenant compte de la signification de chaque noeud. La sémantique d'un noeud est déterminée par son label et celle de tous les noeuds qui sont plus haut dans la hiérarchie. Les relations possibles retournées par l'algorithme du matching sémantique sont l'égalité, l'intersection, la disparité, la généralité ou la spécificité.

Dans ce cas, le problème du matching est vu comme un problème de satisfaction d'un ensemble de formules du calcul propositionnel. Les graphes et les correspondances à tester sont traduits en formules de la logique propositionnelle en considérant non seulement leur nom, mais également la position des concepts dans le graphe.

- QOM

QOM (Quick Ontology Mapping) est une approche qui a été conçue pour fournir un outil efficace de matching pour la création au vol des alignements entre les ontologies (Ehrig et Staab 2004).

Afin d'accélérer l'identification des similarités entre deux ontologies, QOM ne compare pas celles de la première ontologie avec toutes les entités de la seconde ontologie, mais emploie des heuristiques (par exemple, labels semblables) pour abaisser le nombre de mappings candidats. Le calcul réel de similarité est effectué en utilisant une large gamme de fonctions de similarité, telles que la similarité des strings.

Plusieurs mesures de similarité sont calculées et servent d'entrée à une fonction d'agrégation. QOM applique une fonction sigmoïde qui fait ressortir différentes similarités élevées et basses. Les correspondances réelles entre les entités des ontologies sont extraites en appliquant un seuil de mesure agrégée de similarité. La sortie d'une itération peut être utilisée en tant qu'élément d'entrée pour l'itération suivante afin de raffiner le résultat. Après un certain nombre d'itérations, une table de correspondances entre les ontologies est obtenue.

- ASCO

ASCO est un algorithme qui permet de comparer deux ontologies (Thanh et al. 2004). Il trouve des mappings en suivant un processus à deux phases.

- La phase linguistique dans laquelle la valeur de similarité entre deux entités, telles que des concepts ou des relations provenant de deux ontologies différentes, est calculée à partir de différentes informations disponibles telles que leurs noms, leurs étiquettes (des labels qui fournissent une version compréhensible par un humain du nom du concept ou de la relation) et leurs descriptions. Le calcul de la valeur de similarité linguistique est effectué de plusieurs manières. Pour améliorer la précision du calcul et pour exploiter les relations de synonymie ou hyperonymie entre termes, ASCO intègre WordNet.
- La phase structurelle exploite les informations taxonomiques dans les structures des ontologies. Elle utilise des heuristiques et les connaissances du domaine pour calculer les valeurs de similarité structurelle entre entités des deux ontologies. Les valeurs de similarité dans les deux phases sont combinées pour obtenir les valeurs de similarité finales entre les entités. Les alignements sont déduits de ces valeurs.

2.2.3 Fusion des ontologies

Le processus de fusion d'ontologies est fondé sur deux ontologies d'entrée (ou plus) et retourne une seule ontologie basée sur les ontologies sources. La fusion manuelle d'ontologies à l'aide d'outils conventionnels d'édition sans support est longue, difficile, et

sujette à l'erreur. Par conséquent, plusieurs approches de fusion sont récemment proposées (Gaëlle 2002).

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 2.10). 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.

Dans la nouvelle ontologie, on retrouve des régions des anciennes ontologies qui sont plus ou moins inchangées. Ce genre d'approche est utilisé quand les différentes ontologies à intégrer ont une forme ou un contenu très hétérogène. Le résultat est une ontologie plus homogène qu'avec une approche par assemblage, mais le processus d'intégration n'est pas facilement automatisable (Pinto et al. 1999).

Dans le cadre de la fusion il faut faire attention à:

- Préserver la hiérarchie des entités conceptuelles.
- Préserver les types des entités conceptuelles.
- Trouver les entités conceptuelles qui peuvent se superposer.
- Corréler les entités conceptuelles qui sont sémantiquement proches par équivalence ou par des relations de subsomptions.

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 et al. 2000), FCA-Merge (Stumme et Maedche 2001) et OntoMerge (Dou et al. 2002).

- **PROMPT**

PROMPT est un outil dont le processus de fusion est de type interactif. L'ensemble de phases associé à ce processus comprend les étapes suivantes :

- Les candidats à la fusion sont identifiés à partir des similarités des noms de classes. Le résultat est présenté à l'utilisateur comme une liste d'opérations potentielles de fusion.

- L'utilisateur choisit une des opérations suggérées par la liste ou spécifie directement l'opération de fusion.
- Le système effectue l'action demandée et exécute automatiquement les changements additionnels dérivés de cette action.
- Le système crée une nouvelle liste d'actions suggérées par l'utilisateur en se basant sur la nouvelle structure de l'ontologie. Il détermine les conflits présentés par la dernière action, les solutions possibles à ces conflits puis présente ces derniers à l'utilisateur.

PROMPT identifie un ensemble d'opérations pour la fusion d'ontologies (fusion des classes, fusion de slots, fusion des liens, etc.) et un ensemble de conflits possibles consécutifs à l'application de ces opérations (conflits de nom, redondance dans la hiérarchie des classes).

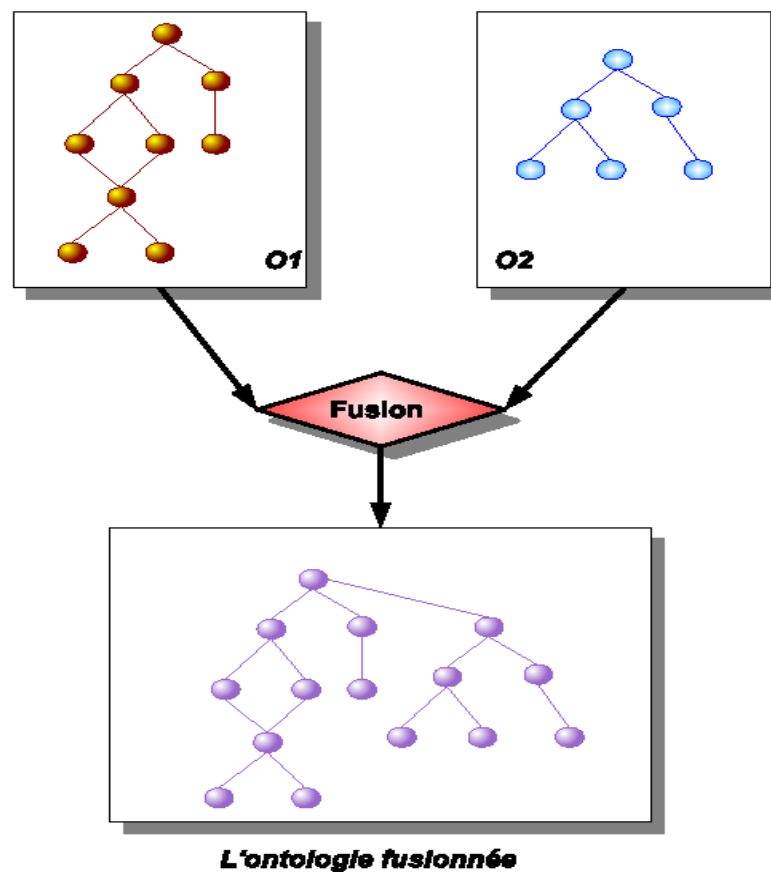


Figure 2.10 – Le principe de la fusion d'ontologies (Mellal 2007)

- **CHIMAERA**

CHIMAERA est un environnement pour la fusion d'ontologies (McGuinness et al. 2000) qui aide les utilisateurs à créer et à maintenir des ontologies distribuées dans le Web. Il fournit des outils de diagnostic et supporte deux fonctions principales :

- Fusionner plusieurs ontologies.
- Analyser des ontologies individuelles ou multiples.

La fusion est exprimée par un opérateur entre les paires de termes, de noms et de définitions considérées comme candidats à la fusion. CHIMAERA dispose également de techniques permettant de lier les termes par des relations de subsomption, disjonction, etc. L'analyse effectuée par Chimaera comprend aussi bien une vérification de la rigueur logique d'une ontologie que le diagnostic des erreurs habituelles dans sa conception.

- **FCA-Merge**

Stumme et Maedche ont proposé FCA-Merge (Formal Conceptual Analysis Merge) dans le but de fusionner des ontologies locales qui partagent le même ensemble d'instances. Pour cela, les auteurs exploitent l'analyse formelle des concepts (Stumme et Maedche 2001). Le processus de fusion nécessite trois étapes :

- L'extraction des instances à partir de documents de type texte.
- La génération du treillis de concepts en appliquant l'analyse formelle des concepts aux instances. Chaque noeud du treillis est associé à un ensemble de concepts des ontologies locales lorsque les instances associées sont contenues dans les mêmes documents.
- La génération interactive de l'ontologie fusionnée est l'étape finale de l'analyse du treillis qui construit l'ontologie globale. Cette étape est à la charge du concepteur.

- **OntoMerge**

OntoMerge est un outil qui facilite la création d'une ontologie appelée Ontologie Passerelle. Celle-ci importe les ontologies originelles et relie les concepts grâce à un certain nombre d'axiomes (Dou et al. 2002).

OntoMerge est une approche en ligne dans laquelle les ontologies source sont maintenues après l'opération de fusion, alors que dans PROMPT l'ontologie fusionnée remplace les ontologies source. Le résultat de l'opération de fusion dans OntoMerge n'est

pas une ontologie complètement fusionnée, comme dans PROMPT, mais une ontologie passerelle qui importe des ontologies source.

Des règles de traduction issues des axiomes relient la partie de convergence des ontologies source. Les ontologies source sont traitées avec les axiomes passerelle comme une seule théorie par un démonstrateur de théorèmes optimisé pour trois opérations principales :

- La traduction de l'ensemble de données d'une représentation à une autre.
- La génération d'extensions d'ontologie qui, étant données deux ontologies reliées O_1 et O_2 , engendre une extension O_2s de O_2 ainsi qu'une extension O_1s de O_1 .
- L'interrogation de différentes ontologies.

3. Discussion

La majorité des approches citées dans les différentes catégories (mapping, fusion et alignement) exploitent des mécanismes qui ne reposent pas sur des fondements théoriques et/ou modèles mathématiques robustes. La plupart de ces mécanismes utilisent des heuristiques, la logique propositionnelle ou les probabilités.

Ces approches reposent sur l'utilisation des calculs de similarités syntaxiques afin d'identifier les correspondances entre concepts, mais prennent rarement en considération leur sémantique.

Les techniques citées précédemment s'accordent sur certaines propriétés mais elles présentent certaines limites :

1. La plupart de ces approches, telles que MAFRA, RDFT etc, sont limitées à l'utilisation d'algorithmes semi-automatiques pour le mapping, la fusion et l'alignement d'ontologies.
2. La prolifération des concepts introduits dans les ontologies engendre souvent une explosion combinatoire.
3. Pour lier les concepts de différentes ontologies, ces approches s'appuient sur des similarités syntaxiques entre ces concepts. Toutefois, deux concepts peuvent avoir une même syntaxe alors que leur sémantique est différente car placée dans des contextes différents. A l'inverse, deux concepts peuvent avoir la même sémantique alors qu'ils sont décrits par différentes syntaxes.
4. Enfin, ces méthodologies sont difficilement compréhensibles par les utilisateurs à cause de leur complexité.

Dans l'infrastructure web sémantique qui repose sur les logiques de descriptions pour la construction des ontologies (voir chapitre1), la fusion de ces ontologies repose sur l'insertion des entités sémantiques des ontologies initiales à l'ontologie globale qui doit respecter les notions de subsomption et classification.

Pour cela nous devons déterminer les relations sémantiques (équivalence, subsomption, disjonction, ...) existantes entre ces concepts qui sont déjà organisés par la subsomption dans les ontologies initiales. Où, la relation d'équivalence permet de fusionner les entités similaires, la relation de subsomption permet d'organiser les concepts et les rôles en hiérarchies dans un graphe et la relation de disjonction permet d'éviter les comparaisons inutiles, car si deux entités sont disjointes alors leur subsumés sont disjointes aussi, par exemple: les entités homme et femme.

Cependant, la majorité des approches d'intégration des ontologies comme nous l'avons déjà indiqué, tel que: GLUE (Doan et al. 2002), QOM (Ehrig et Staab 2004) et S-Match (Giunchiglia et al. 2005) sont basées uniquement sur le calcul des mesures de similarité entre les entités sémantiques pour effectuer le processus d'appariement des ontologies (voir section 2).

Ces mesures renvoient comme résultat une valeur qui détermine les entités similaires (seulement la relation d'équivalence) des ontologies comparées. Cela signifie que ces mesures ne sont pas suffisantes pour la construction de l'ontologie globale. En outre, ils ne peuvent pas résoudre tous les problèmes des équivalences entre les entités sémantiques, par exemple nous pouvons avoir la même valeur pour deux calculs différents car ils reposent sur des critères syntaxiques et structurels.

4. Conclusion

Ce chapitre introduit la notion de l'intégration des ontologies. Nous avons commencé par la présentation des hétérogénéités qui peuvent exister entre les différentes ontologies et leurs impacts sur le processus de l'intégration. Par la suite, nous avons défini le calcul des mesures de similarité entre les entités sémantiques. Enfin, nous avons présenté en détail les types d'intégration des ontologies.

Pour conclure, nous pouvons constater que l'unicité de format de représentation de la connaissance est un mécanisme qui facilite l'intégration des différentes ontologies, et sert à améliorer la communication entre les différents acteurs dans un même domaine. Dans le chapitre suivant, nous proposons un système pour l'intégration des ontologies décrites en logique de description, en appuyant notre système sur

un exemple concret. Notre travail fait partie de la deuxième catégorie des méthodes d'intégration qui consiste à aligner des ontologies décrites en LDs. Le système proposé génère une ontologie générique et extensible des relations sémantiques entre les entités sémantiques des ontologies intégrées.

Proposition d'un système de détection des relations sémantiques dans les ontologies basée sur le Raisonnement à partir de cas

1. Introduction

Les études d'évaluation ont montré que le mapping résultant des approches d'appariement existantes soit il contient un montant équitable des erreurs ou bien il ne couvre qu'une petite partie des ontologies concernées (Euzenat et al. 2006) (Euzenat et al. 2007) (Caracciolo et al 2008). Afin de pallier ces insuffisances, nous proposons de réduire l'application des mesures autant que possible en utilisant des outils plus fiables comme le raisonnement à partir de cas (RÀPC) basé sur des mécanismes formels tels que la subsomption.

Comme nous l'avons déjà mentionné dans le chapitre1, les ontologies Web sémantique sont basées sur la relation de subsomption, où tout concept se compose d'une description propre définie par des propriétés locales et d'une description partagée avec ses subsumants (comme c'est le cas entre une sous-classe et ses super-classes dans un langage à objets) (Napoli 1997). Cela implique que la comparaison des entités pour déduire des relations sémantiques entre eux peut bénéficier de la comparaison de leurs subsumants à travers des petites modifications. La réutilisation de la solution d'un problème déjà résolu pour résoudre un autre problème similaire est le principe du RÀPC. Selon (D'aquin, 2006), Le mécanisme de RÀPC consiste à résoudre un problème cible, en adaptant la solution d'un problème source remémoré de la base de cas, considéré comme similaire au problème cible.

Dans ce travail, nous proposons un nouveau système (Kolli et Boufaida 2009) (Kolli et Boufaida 2010) basé sur un mécanisme du RÀPC pour la détection des relations sémantiques entre les entités sémantiques des ontologies en utilisant les techniques d'extraction de connaissances dans des bases de données (ECBD) et la distance sémantique inspirée par (Cullot et Jouanot 2003) et adaptée aux logiques de description dans (Kolli et Boufaida 2004).

Ce système génère une ontologie générique et extensible des relations sémantiques entre les entités sémantiques. Nous avons décrit cette ontologie avec le formalisme des graphes conceptuels qui nous permet de représenter les cas dans une forme précise, lisible et utilisable par un ordinateur.

La suite de ce chapitre est organisée comme suit: Dans la prochaine section, nous présentons brièvement le cycle de vie du mécanisme du RÀPC. La section 3 décrit de manière détaillée le système proposé basé sur deux phases: la phase de conception et la phase de production. Au cours du processus, nous utilisons un exemple pour bien expliquer le fonctionnement du système proposé. Le chapitre se termine par une conclusion.

2. Le raisonnement à partir de cas (RÀPC)

Comme nous l'avons déjà évoqué, le raisonnement à partir de cas résout de nouveaux problèmes en s'appuyant sur la remémoration et l'adaptation de problèmes passés déjà résolus. D'après (Cordier et al. 2006) le cycle du RÀPC est constitué de cinq étapes :

- *Elaboration.* Les informations nécessaires à la formulation d'un problème sont collectées et structurées de façon à constituer un nouveau cas : le cas cible. Lors de cette étape, le système sollicite l'utilisateur ou l'environnement extérieur (bases de données, systèmes d'information) pour obtenir l'ensemble des informations nécessaires à la poursuite du raisonnement.
- *Remémoration.* L'étape de remémoration consiste à chercher dans la base de cas un ou plusieurs cas passés résolus jugés similaires au cas cible. La sélection des cas similaires s'appuie généralement sur une mesure de similarité. Certains systèmes conservent plusieurs cas remémorés pour les combiner par la suite, mais la plupart du temps, un seul cas est conservé pour la suite du processus. On l'appelle le cas source. La sélection du cas source parmi les cas remémorés peut être la conséquence d'une sélection par le système ou bien résulter d'un choix effectué par l'utilisateur.
- *Réutilisation.* Cette étape permet d'obtenir une solution au cas cible à partir de la solution du cas source sélectionné qui est tout d'abord recopiée puis éventuellement adaptée afin de satisfaire les contraintes du problème posé. L'adaptation s'appuie sur des connaissances d'adaptation qui peuvent prendre différentes formes selon les systèmes.
- *Révision.* Il est possible que la solution proposée par le système ne convienne pas à l'utilisateur ou bien qu'une fois mise en application, elle s'avère inapte à résoudre le problème posé. L'utilisateur a donc, pendant la phase de révision, l'opportunité de modifier, corriger ou même refuser la solution proposée. L'étape de révision permet d'identifier les causes éventuelles des échecs et de proposer des adaptations supplémentaires pour conduire à une solution satisfaisante : il s'agit du cas révisé. Cette étape est à la base du processus d'apprentissage permettant à la fois d'améliorer des connaissances d'adaptation existantes et d'en faire émerger de nouvelles.
- *Mémorisation.* Traditionnellement, on considère l'étape de mémorisation comme l'étape durant laquelle la base de cas est enrichie par le cas cible révisé. Cette mémorisation implique une mise à jour des index permettant de retrouver les cas et parfois un processus de maintenance pour réorganiser la base de cas. Mais l'étape de mémorisation est également le siège de l'apprentissage d'autres types de connaissances. En effet, c'est durant cette étape que l'on peut concrétiser les efforts effectués durant les autres étapes pour apprendre d'autres connaissances.

3. Détection des relations sémantiques entre les ontologies en utilisant le raisonnement à partir de cas (RÀPC)

L'objectif de notre travail est de proposer un système basé sur le RÀPC pour déduire les relations sémantiques entre les entités sémantiques des ontologies. Ici les ontologies sont décrites en logiques de description. En effet les hiérarchies de concepts ou de relation de ces ontologies sont organisées selon les deux mécanismes : classification et subsumption. Cette organisation implique l'existence des concepts ou bien des relations dans les définitions intensionnelles de leurs subsumés. Ce pendant la détection des relations sémantiques entre les entités sémantiques pourra se faire à travers la combinaison des relations de similitudes entre leurs subsumants. La réutilisation de la solution d'un problème déjà résolu pour résoudre un autre problème similaire est le principe du RÀPC.

Comme tout système à base des connaissances, le cycle de vie d'un système de RÀPC distingue plusieurs phases : la phase de conception, la phase de production et la phase de maintenance. Dans la phase de conception, nous définissons les méthodes de résolution de problèmes qui seront utilisées pour constituer les bases de connaissances du système, définir une base de cas résolus initiale, décrire les connaissances du domaine, les connaissances de similarité et les connaissances d'adaptation.

Le système peut également être utilisé avec des cas connus afin d'instancier la base de cas et de fournir une base au raisonnement. La question du formalisme de représentation des connaissances se pose également à ce moment. La phase de production consiste à utiliser le système pour résoudre de nouveaux problèmes. Le cycle de raisonnement appliqué pour mener cette tâche à bien est étudié dans le deuxième paragraphe. Enfin, notons que le système peut effectuer seul les opérations de maintenance, être sollicité par l'expert pour les accomplir ou bien encore solliciter l'expert pour qu'il lui apporte son aide (Cordier et al. 2006).

Dans ce chapitre, nous détaillons les différentes phases de notre système en l'appliquant sur deux ontologies du même domaine pour pouvoir représenter les cas particuliers. Les figures 3.1 et 3.2 représentent les deux ontologies que nous avons choisies. Ces ontologies sont décrites en LDs.

3.1 Phase de conception

D'après Kolodner (Kolodner 1998), un cas est constitué des mêmes composantes quelque soit le domaine d'application. Ces composantes sont un problème et une solution. En effet, un cas est un couple, noté (pb, sol(pb)), associant un problème pb à sa solution sol(pb). Dans cette même optique, pour notre système, un cas est composé des deux éléments suivants :

- *Le problème.* C'est déterminer la relation de similitude entre deux concepts C_i et C_j qui appartiennent à deux ontologies différentes.
- *La solution.* La détection de la relation sémantique entre ces deux concepts. Nous considérons limitons la relation sémantique rel à une des relations de l'ensemble: $\{\equiv, \perp, \&, \subseteq, \supseteq\}$.
 - $rel(C_i, C_j) = \subseteq$ signifie que C_i est plus générale que C_j . Par conséquent, nous pouvons dire que C_i est le subsumant de C_j .
 - $rel(C_i, C_j) = \supseteq$ signifie que C_i est moins générale que C_j . Donc, C_i est le subsumé de C_j .
 - $rel(C_i, C_j) = \equiv$ signifie que C_i est équivalent à C_j . La relation d'équivalence est définie comme la subsumption dans les deux sens. $rel(C_i, C_j) = \&$ signifie que C_i est superposé avec C_j .
 - Enfin, $rel(C_i, C_j) = \perp$ signifie que C_i est différent de C_j .

Personne \subseteq TOP
 Ensemble \subseteq TOP
 Algérien \subseteq Personne
 Homme \subseteq Personne
 Femme = personne \cap \neg (homme)
 Mineur = personne \cap \forall âge.nombre \cap $<$ 18
 Adulte = personne \cap \forall âge.nombre \cap \geq 18
 Algerien _membre = personne \cap \forall
 membre.algérien
 Equipe = ensemble \cap \forall membre.personne \cap \geq 2
 membre

Figure 3.1– La description des concepts de l'ontologie O_1

```
Humain =  $\exists$  nom.chaîne  $\cap$   $\exists$ 
adresse.chaîne
Mâle  $\subseteq$  TOP
Parent = humain  $\cap$   $\exists$  enfant.humain
Femelle =  $\neg$  Mâle
Homme = humain  $\cap$   $\forall$  sexe.mâle
Femme = humain  $\cap$   $\forall$  sexe.femelle
Fille = femme  $\cap$   $\forall$  enfant.humain
Garçon = homme  $\cap$   $\forall$  enfant.humain
```

Figure 3.2– La description des concepts de l'ontologie O_2

L'application du RÀPC pour la déduction des relations sémantiques entre des ontologies diffusées au sein du web sémantique implique la représentation des connaissances utiles à ce raisonnement d'une manière formelle à l'aide des logiques de descriptions. Les mécanismes de raisonnements associés à ce formalisme peuvent être utilisés pour le RÀPC, en particulier, la subsomption et la classification qui permettent de construire une hiérarchie de concepts de la base de cas.

A la fin de chaque étape de résolution de problème, le nouveau cas résolu est mémorisé dans la base de cas pour pouvoir être réutilisé ultérieurement. Il en résulte une augmentation graduelle de taille de la base de cas, ce qui met donc en évidence le besoin d'organiser et de maintenir la base de cas tout au long de la vie du système. Pour répondre à ce besoin, nous utilisons l'ontologie qui a pour rôle de modéliser ce type de connaissances au sein du web sémantique. Cette ontologie générique et extensible est appelée ontologie d'alignement O_A .

Le formalisme que nous utilisons pour la description de cette ontologie sont les graphes conceptuels qui nous permettent de représenter les cas de manière précise, lisible et utilisable par un ordinateur. L'ensemble des relations déduites à ce niveau représente l'alignement des ontologies A (voir la figure 3.3).

De manière générale, un graphe conceptuel est défini comme un graphe avec deux types de nœuds (Chawk 1996): les concepts et les relations conceptuelles,

- Les concepts représentent dans notre situation les entités sémantiques des deux ontologies O_1 et O_2
- Les relations conceptuelles symbolisent la relation sémantique entre deux concepts.

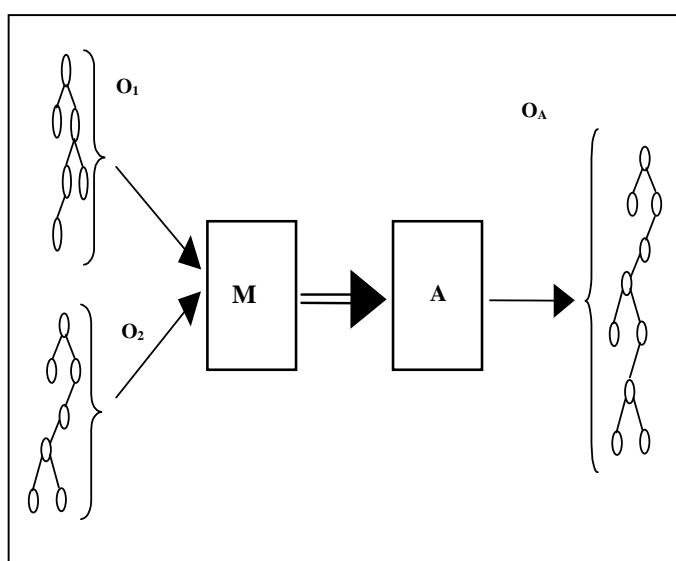


Figure 3.3– Processus d'appariement des ontologies

Les concepts sont représentés graphiquement par des rectangles ou sont placés entre crochets [Concept]; les relations conceptuelles sont représentées par des cercles ou bien sont placés entre parenthèses (relation conceptuelle), avec un seul arc entrant et un seul arc sortant. Un arc entrant associe un concept à une relation conceptuelle, et un arc sortant relie une relation conceptuelle à un concept.

Les libellés des arcs ont été ajoutés pour une meilleure compréhension. Dans le formalisme des graphes conceptuels, les arcs sont numérotés afin de les différencier.

A ce niveau, le processus d'appariement des ontologies (**M**) consiste à comparer l'ensemble des entités sémantiques primitives de deux ontologies en utilisant la distance sémantique. Le résultat sera une des relations de l'ensemble: $\{\equiv, \perp, \cong\}$ (Figure 3.4). $C_i \cong C_j$: signifie que la relation entre C_i et C_j est floue. Cette relation floue peut être l'une des relations de l'ensemble: $\{\&, \subseteq, \supseteq\}$.

Si le résultat de la comparaison n'est pas flou, c'est à dire: qu'il est égal à une des relations de l'ensemble: $\{\equiv, \perp\}$ alors: directement, nous ajoutons ce cas à la base de cas (l'ontologie O_A). Mais, si le résultat de la comparaison est flou, c'est à dire: qu'il est égal à \equiv , alors: sa précision sera calculé au niveau de l'étape de la réutilisation.

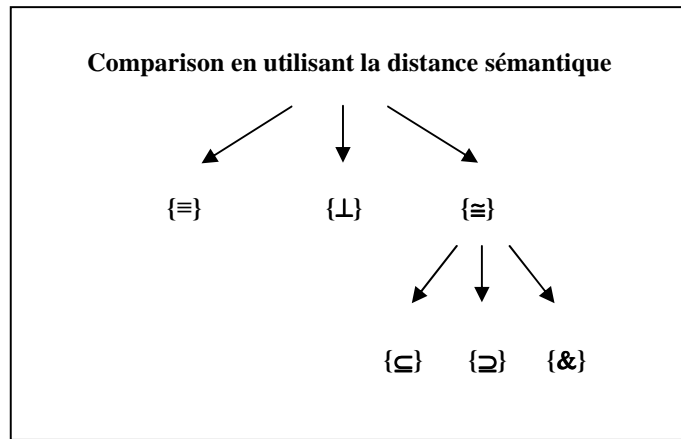


Figure 3.4– Les résultats de la comparaison

3.1.1 Distance sémantique

Le calcul de la distance sémantique inspirée de (Cullot et Jouanot 2003) entres les entités sémantiques utilise la notion de synonymie et d'antonymie.

La définition de synonymie d'après (Miller 1990) est la suivante : « Deux expressions sont synonymes dans un contexte linguistique C si la substitution de l'une pour l'autre en C ne modifie pas la valeur de vérité de la phrase dans laquelle la substitution est faite. Un synonyme donc est un terme de substitution pour nommer une entité sémantique. Un antonyme est un terme qui contredit fondamentalement la signification d'une entité. Pour calculer cette distance nous supposons que chaque ontologie possède un dictionnaire qui définit les synonymes et les antonymes de chaque concept et de chaque rôle de l'ontologie.

Pour pouvoir appliquer cette distance sur les ontologies utilisées, nous rajoutons à celles-ci une hiérarchie de meta-classes conceptuelles qui servent comme des méta-données sur les concepts en associant à chaque concept une liste de synonymes et d'antonymes.

- **Calcul de la distance sémantique:**

Le calcul de la distance, appelée distance globale, entre deux entités est calculée à partir de calculs de quatre distances élémentaires, deux distances élémentaires (DMC et DC) sont appliquées sur les concepts et les deux autres (Dr et Dr_j) sont appliquées sur les rôles.

Ces distances prennent en compte les différents éléments de l'environnement d'un concept dans son contexte. Ces éléments sont le domaine dont dérive éventuellement un concept, la terminologie du concept et les rôles joués.

Le calcul de la distance globale s'effectue en quatre étapes. Chaque étape permet soit de restreindre le nombre des concepts retenus comme potentiellement similaires, soit d'affiner le calcul de la distance entre des concepts retenus aux étapes précédentes.

Le premier calcul effectué se base sur les relations éventuelles des entités sémantiques avec leurs domaines de définition.

• **Distance élémentaire DMC (C, C')**

Soient C et C' deux concepts de deux ontologies différentes. La distance DMC (C, C') se calcule de la manière suivante :

- $DMC(C, C') = 1$ si C et C' sont du même domaine.
- $DMC(C, C') = 0$ sinon.

• **Distance élémentaire DC (C, C')**

Le calcul de la deuxième distance élémentaire appelée DC se base sur la taxonomie des concepts à comparer (C et C'). Si le concept C est un synonyme du concept C' alors : $DC(C, C') = DMC(C, C')$. Sinon $DC(C, C') = 0$.

Deux autres distances élémentaires sont calculées. Elles prennent en compte les rôles associés aux concepts C et C'.

• **Distance élémentaire $Dr(C.r_i, C'.r_j)$**

La distance Dr mesure la similarité entre des rôles (resp. r_i et r_j) de deux concepts (resp. C et C').

- $Dr(C.r_i, C'.r_j) = 1$ si r_i est synonyme r_j

- $Dr(C.r_i, C'.r_j) = 0$ sinon.

Chaque rôle du concept C est comparé à chaque rôle du concept C' . Une matrice des distances des rôles est ainsi construite où chaque élément $Dr(C.r_i, C'.r_j)$ compare le rôle r_i de C avec le rôle r_j de C' .

- **Distance élémentaire $Dr_j(C, C')$**

Enfin la distance élémentaire Dr_j mesure la similarité entre deux concepts en fonction des rôles. Pour cela on calcule, une norme de la matrice MR des rôles (calculée avec la distance Dr).

$$\|MR\| = (\sum_{i=1..n} \lambda(i)) / n$$

- avec $\lambda(i) = 1$ si $\sum_{j=1..m} Dr(C.r_i, C'.r_j) > 0$,
- $\lambda(i) = 0$ sinon.

La distance élémentaire $Dr_j(C, C')$ vaut :

- $DC(C, C') \times \|MR\|$ si C et C' jouent des rôles,
- 0 si C et C' n'ont aucun rôle.

- **Distance globale $D(C, C')$ (dans le cas général) :**

La distance globale entre deux concepts C et C' considère la taxonomie et l'environnement complet de ces deux concepts. L'évaluation prend en compte, comme le montre la figure 3.5, les rôles joués par les concepts $C (r_1 \dots r_u)$ et $C' (r_1 \dots r_t)$, les concepts qui définissent le concept $C (C_1 \dots C_n)$ et le concept $C' (C'_1 \dots C'_m)$ et la taxonomie des deux concepts C et C' .

- $C = C_1 \cap C_2 \dots \cup C_i \dots \cap r_1.C_j \cup \dots \cap r_u.C_n$
- $C' = C'_1 \cap C'_2 \dots \cup C'_i \dots \cap r_1.C'_j \cup \dots \cap r_u.C'_m$

Figure 3.5– Comparaison des concepts

On calcule tout d'abord une matrice de rapprochement qui compare deux à deux les concepts qui définissent les concepts C et C'.

La distance globale $D(C, C')$ est calculée comme une pondération de la distance $DC(C, C')$, de la distance $Drj(C, C')$ qui mesure l'environnement des rôles de C et C' et d'une moyenne sur la matrice de rapprochement qui mesure les similarités entre les concepts qui définissent les concepts C et C'.

$$D(C, C') = \frac{1}{2} DC(C, C') + \frac{1}{2} (\alpha (\sum_{i=1..n} \sum_{j=1..n} R_{C, C'}(i, j) / n) + \beta Drj(C, C'))$$

Avec $\alpha \in [0, 1]$ et $\beta \in [0, 1]$ et $\alpha + \beta = 1$.

Afin d'instancier notre base de cas (l'ontologie de l'alignement O_A), nous prenons toutes les entités primitives des deux ontologies O_1 et O_2 de l'exemple que nous avons pris et qui sont:

- Les concepts primitifs : personne, homme, ensemble et algérien de l'ontologie O_1 et les concepts humain et mâle de l'ontologie O_2 .
- les rôles primitifs : membre et age de O_1 et les deux rôles: enfant et sexe de O_2 .

Le premier problème ici (noté pb_1) est de trouver le relation sémantique entre les concepts racines des deux ontologies O_1 et O_2 c à d: $pb_1 = \text{Rel}(\text{personne}, \text{humain})$. La résolution de ce problème par le calcul de la distance sémantique est comme suit :

1. Calcul de la distance DMC (humain, personne) :

$DMC(\text{humain}, \text{personne}) = 1$ (humain et personne sont du même domaine).

2. Calcul de la distance DC (humain, personne) :

Humain est synonyme de personne donc : $DC(\text{humain}, \text{personne}) = DMC(\text{humain}, \text{personne}) = 1$.

3. Calcul de la distance Dr (humain.ri, personne.rj) :

La distance Dr mesure la similarité entre des rôles (ri et rj) de deux concepts (humain et personne). Nous remarquons que les concepts : humain et personne n'ont aucun rôle. Donc $Dr(\text{humain.ri}, \text{personne.rj}) = Drj(\text{humain}, \text{personne}) = 1$.

4. Calcul de la distance globale D (humain, personne) :

$$D(\text{humain, personne}) = \frac{1}{2} DC(\text{humain, personne}) + \frac{1}{2} (\alpha (\sum_{i=1..n} R_{\text{humain, personne}}(i)/n) + \beta Drj(\text{humain, personne})).$$

Les concepts humain et personne ne sont pas définis avec d'autres concepts. Donc : $(\sum_{i=1..n} R_{\text{humain, personne}}(i, j) / n) = 1$. Ce qui implique : $D(\text{humain, personne}) = 1$.

Deux entités (concepts ou rôles) sont similaires si la distance sémantique entre elles est au moins égale à une valeur minimale fixée. Nous supposons que cette valeur est égale à 0.8.

$D(\text{humain, personne}) = 1 \geq 0.8$ ce qui implique que le concept humain est similaire au concept personne.

Après avoir appliqué la mesure de similarité, qui est la distance sémantique sur les concepts personne et humain, nous avons trouvé que personne est équivalent à humain.

Donc, la solution $sol_1 = \equiv$. Ce cas (cas_1) est implémenté en formalisme des graphes conceptuels, comme suit:

$$[\text{personne}] \xrightarrow{e_1} (\equiv) \xrightarrow{s_1} [\text{humain}]$$

De la même façon, nous appliquons la distance sémantique sur les autres concepts primitifs. Les résultats de la comparaison entre ces concepts sont implémentés aussi en graphes conceptuels, comme suit:

$$[\text{ensemble}] \xrightarrow{e_2} (\perp) \xrightarrow{s_2} [\text{mâle}]$$

$$[\text{ensemble}] \xrightarrow{e_3} (\perp) \xrightarrow{s_3} [\text{humain}]$$

$$[\text{algérien}] \xrightarrow{e_4} (\equiv) \xrightarrow{s_4} [\text{humain}]$$

$$[\text{algérien}] \xrightarrow{e_5} (\equiv) \xrightarrow{s_5} [\text{humain}]$$

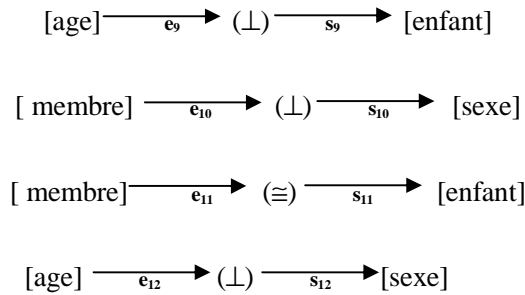
$$[\text{personne}] \xrightarrow{e_6} (\equiv) \xrightarrow{s_6} [\text{mâle}]$$

$$[\text{homme}] \xrightarrow{e_7} (\equiv) \xrightarrow{s_7} [\text{humain}]$$

$$[\text{homme}] \xrightarrow{e_8} (\equiv) \xrightarrow{s_8} [\text{mâle}]$$

Les trois premiers résultats seront stockés dans l'ontologie de l'alignement O_A . Les autres résultats (flous) seront déterminés après.

Les résultats de la comparaison entre les rôles primitifs des ontologies O_1 et O_2 sont les suivants:



2.1. Phase de production

Après avoir présenté les mécanismes et les outils utilisés, nous montrons dans cette phase comment appliquer les étapes du RÀPC pour résoudre le problème de La détection des relations de similitude et nous proposons quelques stratégies permettant cette détection.

Le cycle du RÀPC que nous utilisons se décompose en cinq opérations (figure 3.6) : l'élaboration, la remémoration, la réutilisation, la révision et la mémorisation.

2.1.1. L'élaboration

Cette étape permet la constitution de la première partie du cas cible c'est à dire : le problème, qui est dans notre cas la détection de la relation de sémantique entre deux concepts des ontologies à intégrer.

Dans la phase précédente nous avons instancié notre base de cas avec les cas suivant :

$\text{cas}_1 = (\text{Rel}(\text{personne}, \text{humain}), \cong)$

$\text{cas}_2 = (\text{Rel}(\text{ensemble}, \text{humain}), \perp)$

$\text{cas}_3 = (\text{Rel}(\text{ensemble}, \text{mâle}), \perp)$

$\text{cas}_4 = (\text{Rel}(\text{âge}, \text{enfant}), \perp)$

$\text{cas}_5 = (\text{Rel}(\text{membre}, \text{enfant}), \cong)$

$\text{cas}_6 = (\text{Rel}(\text{âge}, \text{sexe}), \perp)$

et $cas_7 = (\text{Rel}(\text{membre}, \text{sexe}), \perp)$.

Nous prenons maintenant deux autres concepts qui sont parent et algérien _membre. Ces deux concepts sont définis dans leurs ontologies (figure 3.1 et 3.2) comme suit :

$$\text{Parent} = \text{humain} \cap \exists \text{enfant.humain}$$
$$\text{Algérien_membre} = \text{personne} \cap \forall \text{membre.algérien}$$

3.2.2. La remémoration

Le but de la remémoration n'est pas seulement de trouver le cas source le plus similaire au cas cible, nous cherchons aussi le cas le plus facile à adapter. Dans ce même esprit, Leake suggère qu'une bonne remémoration d'un cas permet de faciliter l'effort d'adaptation (Leake et al. 1997).

La remémoration repose traditionnellement sur les mesures de similarité sémantiques qui sont basées sur des hypothèses et des règles. D'après (Cordier et al. 2006), ces mesures peuvent conduire à des résultats médiocres dans la mesure où elles permettent parfois de se remémorer des cas sources, certes très "ressemblants" au cas cible, mais difficiles voire impossibles à adapter. Cette observation montre la limite des mesures de similarité au regard du processus complet de raisonnement.

Dans notre situation, les concepts en question sont organisés selon la relation de la subsumption dans leurs ontologies de base. En effet, les concepts sont définis avec leurs subsumants. Ainsi, les cas les plus faciles à adapter sont ceux qui contiennent les subsumant des concepts concernés.

Nous proposons que le processus de la détection se fait tout d'abord en largeur, c'est à dire : il faut déterminer la relation de similitude entre un concept de l'ontologie O_1 et tous les concepts voisins de l'ontologie O_2 . En suite, répéter cette opération avec leurs subsumés. Dans ce cas, nous pouvons profiter de la relation de subsumption pour faciliter l'effort de l'adaptation. En effet, les cas remémorés sont donc les cas qui contiennent leurs subsumants.

Dans l'exemple précédant le problème pb_2 est la détection de la relation sémantique entre les concepts parent et algérien _membre qui sont des concepts définis. Nous remarquons que le concept humain est le subsumant du concept parent et le concept personne est le subsumant du concept algérien_membre. En effet, le cas remémoré est le cas: cas_1 .

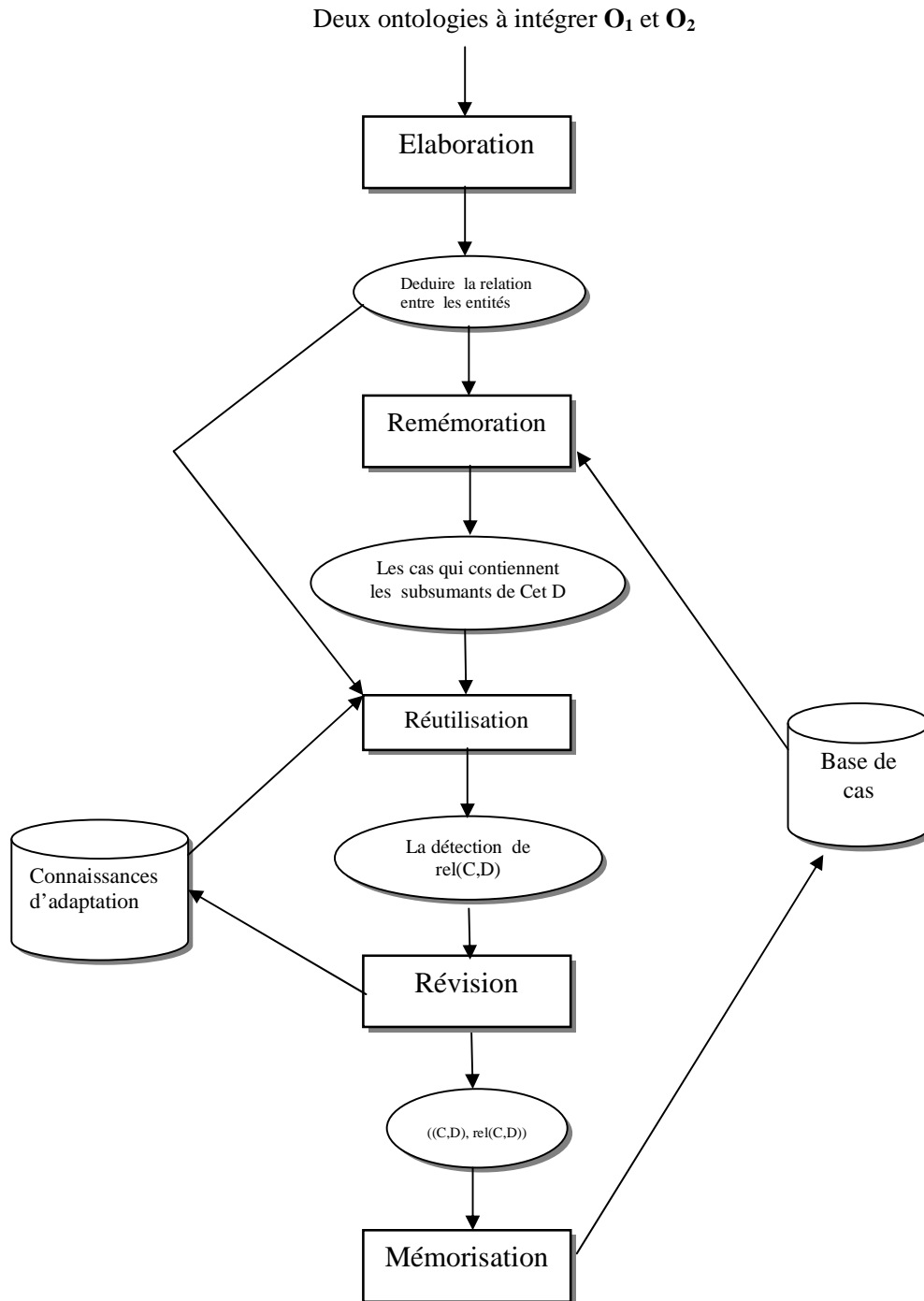


Figure 3.6– Le cycle du RÀPC utilisé pour la détection des relations sémantiques des entités sémantiques

3.2.3. La réutilisation

Cette étape consiste à combiner les solutions des cas remémorés et les adapter pour trouver une solution au cas cible. Dans ce travail, nous nous appuyons sur le modèle d'adaptation proposé dans (d'Aquin et al. 2006) que nous présentons brièvement ci-après :

1. (srce, cible) \rightarrow Δpb , où Δpb encode les similarités et dissimilarités entre des problèmes srce et cible.
2. (Δpb , CA) \rightarrow Δsol , où CA est un ensemble de connaissances d'adaptation et Δsol encode les similarités et dissimilarités entre Sol(srce) et la solution Sol(cible) à construire pour cible.
3. (Sol(srce), Δsol) \rightarrow Sol(cible), Sol(srce) est modifié en Sol(cible) selon Δsol .

Ce modèle est basé sur les techniques d'extraction de connaissances dans des bases de données (ECBD). Dans cette étape nous rappelons les différentes étapes du processus d'ECBD et nous détaillons la façon dont celles-ci sont effectuées dans notre système. Le but de l'ECBD est d'obtenir des connaissances à partir de données. Le processus d'ECBD se fait sous la supervision d'un analyste, qui est un expert du domaine. Une fois l'acquisition des données réalisée, il se déroule en deux étapes: la préparation des données et la fouille de données (Figure 3.7).

a. Préparation des données

La préparation des données est une étape de mise en forme et de sélection des données. L'opération de mise en forme met les données dans un format acceptable pour les opérations de la fouille de données. La sélection des données permet de concentrer la fouille sur un sous-ensemble pertinent de concepts et d'éliminer les données bruitées.

L'étape de préparation des données génère un ensemble des concepts à partir de la base de cas BC, en appliquant successivement deux transformations.

La première transformation Φ formate chaque cas source remémoré (srce, Sol(srce)) en deux ensembles de propriétés booléennes : $\Phi(srce)$ et $\Phi(Sol(srce))$. Le vocabulaire utilisé pour décrire les cas étant celui des ontologies initiales des concepts en question, si srce = Relation(C, D) alors $\Phi(srce) = (\Phi(C), \Phi(D)) = (\{p_{1,C}, \dots, p_{n,C}\}, \{p_{1,D}, \dots, p_{n,D}\})$, avec (p_1, \dots, p_n) sont les entités (concepts et des rôles) qui définissent les concepts C et D. Afin de comparer les concepts avec les concepts et les rôles avec les rôles, nous avons mis c comme un indice pour indiquer les concepts et r pour indiquer les rôles.

Pour l'exemple que nous avons pris, le cas source est: cas-srce = (Relation (personne, humain), \equiv) et le cas-cible = (Rel (parent, Algerien _membre), Sol (cible)).

L'application de la fonction Φ sur le cas source et le cas cible est comme suit: $\Phi(\text{srce}) = \{\{\text{Personne}_c\}, \{\text{humain}_c\}\}$, $\Phi(\text{Sol}(\text{srce})) = \{\equiv\}$ and $\Phi(\text{cible}) = \{\{\text{personne}_c, \text{enfant}_r\}, \{\text{humain}_c, \text{membre}_r, \text{algerien}_c\}\}$

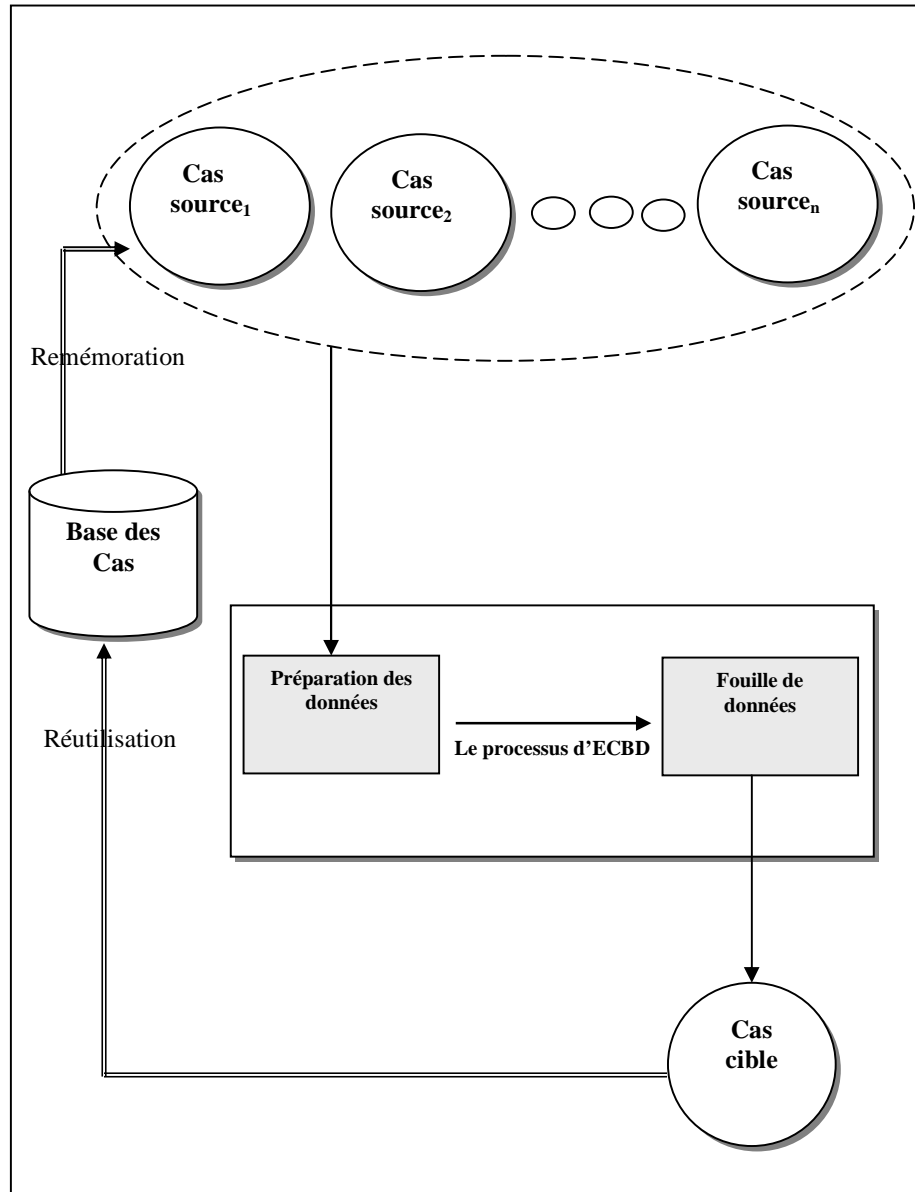


Figure 3.7– La détection des relations sémantiques entre les ontologies avec un système RÀPC

La deuxième transformation produit un concept x à partir de chaque couple de cas sources ($\Phi(\text{cas-source}), \Phi(\text{cas-cible})$). Suivant le modèle d'adaptation présenté dans (d'Aquin et al. 2006), x doit encoder les propriétés de Δ_{pb} et de Δ_{sol} . Δ_{pb} encode les similarités et dissimilarités de $srce$ et cible, c-à-d:

- Les propriétés communes à $srce$ et cible (marquées par “=”),
- Les propriétés de $srce$ que cible ne partage pas (“-”) et
- Les propriétés de cible que $srce$ ne partage pas (“+”).

Toutes ces propriétés sont reliées à des problèmes et sont marquées par pb_1 et pb_2 . pb_1 correspond au deux premiers concepts de $srce$ et cible et pb_2 correspond au deux derniers concepts de $srce$ et cible. Nous comparons respectivement la première et la deuxième partie de $srce$ avec la première et la deuxième partie de cible. Δ_{sol} est calculé de façon similaire et $x = \Delta_{pb} \cup \Delta_{sol}$. Par exemple,

$$\text{si } \begin{cases} \Phi(srce_1) = \{\{a, b, c\}, \{a, d\}\}, \Phi(Sol(srce)) = \{\equiv\} \\ \Phi(srce_2) = \{\{c, d\}, \{b, c, d\}\}, \Phi(Sol(cible)) = \{\perp\} \end{cases}$$

$$\text{Alors : } x = \{a^-_{pb1}, c^-_{pb1}, b^-_{pb1}, d^+_{pb1}, a^-_{pb2}, d^-_{pb2}, c^-_{pb2}, \equiv^-_{sol}, \perp^+_{sol}\}$$

Après l'application de ce modèle sur notre exemple:

$$x = \{\text{Personne}^-_{pb1,c}, \text{enfant}^+_{pb1,r}, \text{humain}^-_{pb2,c}, \text{membre}^+_{pb2,r}, \text{algerien}^+_{pb2,c}, \equiv^-_{sol}\}.$$

b. Fouille de données

La fouille de données permet d'extraire des éléments d'information à partir des données. Nous réalisons cette extraction en appliquant les connaissances d'adaptation (CA). Nous présentons ces connaissances d'adaptation en proposant une nouvelle méthode formelle de la détection des relations sémantiques entre les concepts, que l'on appelle méthode d'appariement des ontologies.

- Appariement des ontologies

Notre méthode d'appariement est divisée en deux opérations: l'agrégation ou la composition des relations sémantiques. Ces opérations sont appliquées sur les relations sémantiques entre les subsumants des concepts en question qui sont déjà déduites avant.

- Composition

Dans l'infrastructure Web sémantique, qui repose sur les LDs pour la construction des ontologies, nous remarquons que les entités sémantiques sont organisées selon la relation de subsomption. Cela, permet la déduction des relations de subsomption (\subseteq , \supseteq) entre les concepts de la même ontologie d'une manière simple et directe.

Cette observation nous permet de dire que la réutilisation des cas stockés dans l'ontologie \mathbf{O}_A peut être faite par l'opération de composition. Ainsi, s'il existe une relation sémantique entre le concept C_1 de l'ontologie \mathbf{O}_1 et un concept C_2 de l'ontologie \mathbf{O}_2 et une autre relation sémantique qui est la relation de subsomption entre le concept C_2 et le concept C_3 de la même ontologie (\mathbf{O}_2), alors il devrait être possible d'obtenir la relation sémantique entre les concepts C_1 et C_3 .

Les résultats possibles de cette composition sont indiqués dans la table 3.1.

Table 3.1 : Composition des relations sémantiques

$C_1 R C_2$ \ $C_2 R C_3$	\equiv	\subseteq	\supseteq	$\&$	\perp
\subseteq	\subseteq	\subseteq	\equiv	$\&$	\equiv
\supseteq	\supseteq	\equiv	\supseteq	\equiv	\perp

La preuve de ces résultats est faite par la notion d'interprétation des LDs comme suit:

1. $((C_1 \equiv C_2 \rightarrow C_1^I = C_2^I) \cap (C_2 \subseteq C_3 \rightarrow C_2^I \subseteq C_3^I)) \Rightarrow (C_1^I \subseteq C_3^I \rightarrow C_1 \subseteq C_3), (\forall I).$
2. $((C_1 \subseteq C_2 \rightarrow C_1^I \subseteq C_2^I) \cap (C_2 \subseteq C_3 \rightarrow C_2^I \subseteq C_3^I)) \Rightarrow (C_1^I \subseteq C_3^I \rightarrow C_1 \subseteq C_3), (\forall I).$
3. $((C_1 \& C_2 \rightarrow (C_1 \cap C_2)^I \neq \emptyset) \rightarrow C_1^I \cap C_2^I \neq \emptyset) \cap (C_2 \subseteq C_3 \rightarrow C_2^I \subseteq C_3^I) \Rightarrow ((C_1 \cap C_3)^I \neq \emptyset \rightarrow C_1 \& C_3), (\forall I).$
4. $((C_1 \equiv C_2 \rightarrow C_1^I = C_2^I) \cap (C_2 \supseteq C_3 \rightarrow C_2^I \supseteq C_3^I)) \Rightarrow (C_1^I \supseteq C_3^I \rightarrow C_1 \supseteq C_3), (\forall I).$
5. $((C_1 \supseteq C_2 \rightarrow C_1^I \supseteq C_2^I) \cap (C_2 \supseteq C_3 \rightarrow C_2^I \supseteq C_3^I)) \Rightarrow (C_1^I \supseteq C_3^I \rightarrow C_1 \supseteq C_3), (\forall I).$

$$6. ((C_1 \perp C_2 \rightarrow C_1^I \neq C_2^I) \wedge (C_3 \subseteq C_2 \rightarrow C_3^I \subset C_2^I)) \Rightarrow (C_1^I \neq C_3^I \rightarrow C_1 \perp C_3), (\forall I).$$

Toujours, selon la notion d'interprétation des LDs, on déduit qu'il existe des résultats flous (\equiv) dans la table 3.1. Cela signifie que la composition de ces relations génère plusieurs résultats.

Nous pouvons prendre par exemple le cas où nous avons: $(C_1 \supseteq C_2)$ et $(C_2 \subseteq C_3)$. Cela implique: $(C_1^I \supset C_2^I)$ et $(C_2^I \subset C_3^I)$. Donc, nous pouvons déduire que: $(C_1^I = C_3^I)$ or $(C_1^I \supset C_3^I)$ or $(C_1^I \subset C_3^I)$. Dans ce cas, la relation entre C_1 et C_3 peuvent être une des relations de l'ensemble: $\{\equiv, \supseteq, \subseteq\}$.

Ainsi, l'application de l'opération de composition à ces cas ne résout pas le problème de la déduction des relations. Pour la raison de la détection de toutes les relations possibles entre tous les concepts, nous proposons d'utiliser également dans cette étape une autre opération, qui est l'agrégation de similarité.

Dans l'exemple précédent, nous avons le cas: (personne \equiv humain) stockés dans l'ontologie \mathbf{O}_{A1} . Nous pouvons aussi déduire les deux cas: (homme \supseteq personne) et (Algérien _membre \supseteq personne) de l'ontologie \mathbf{O}_1 . Selon la table 3.1, l'application de l'opération de composition à ces cas nous donne comme résultat les deux cas: (humain \supseteq homme) et (humain \supseteq Algérien _membre).

Aussi, nous pouvons déduire selon la relation de subsomption les cas suivants: (homme \supseteq humain), (femme \supseteq humain) et (parent \supseteq humain) de l'ontologie \mathbf{O}_2 .

Selon la table 3.1, l'application de l'opération de composition à ces cas, nous donne comme résultat les cas suivants: (personne \supseteq homme), (personne \supseteq femme) et (personne \supseteq parent).

- Agrégation de similarité

En général, on peut avoir plusieurs subsomphants pour la même entité sémantique (pour cette opération, nous ne traitons que les entités définies). Ainsi, ils existent plusieurs cas sources pour chaque paire d'entités. Leurs solutions doivent être combinées en une seule solution pour le cas cible. Nous appelons cette opération: l'agrégation de similarité qui prend en entrée toutes les valeurs des relations de similarité (solutions) des cas source obtenues comme résultat de l'étape précédente afin de les agréger en une seule valeur de relation sémantique en exploitant les techniques des LDs.

Dans la proposition 1, nous donnons quelques combinaisons possibles entre deux concepts. Nous montrons aussi comment appliquer la fonction d'agrégation sur ces concepts.

Proposition 1:

Soient C et D deux concepts définies comme suit: $C = (\cap C_1 \dots C_v \cup C_n \dots C_i)$ et $D = (\cap D_1 \dots D_U \dots \cup D_m \dots D_j)$. Considérons les relations sémantiques $R_1, R_2, \dots R_n$ déduites entre les concepts C_i et D_j .

$\text{Aggreg}(R_1, R_2, \dots R_k) = R$ est la fonction d'agrégation appliquée aux relations sémantiques entre les différents concepts qui définissent les deux concepts C et D. Cette fonction Renvoie la relation R entre les deux concepts C et D. La relation R peut être déduite comme suit:

1. Si $(\exists R_s \in \{R_1, R_2, \dots R_k\} / s = 1 \dots k, \text{ tel que: } R_s = \perp)$ et (si les concepts C et D ne sont que des conjonctions) alors: $\text{Aggreg}(R_1, R_2, \dots R_k) = \perp$ c à d: $C \perp D$.
2. Si $(\forall R_s \in \{R_1, R_2, \dots R_k\} / s = 1 \dots k, \text{ tel que: } R_s = \equiv)$ alors: $\text{Aggreg}(R_1, R_2, \dots R_k) = \equiv$ c à d: $C \equiv D$.
3. Si $(\forall R_s \in \{R_1, R_2, \dots R_k\} / s = 1 \dots k, \text{ tel que: } R_s = \equiv)$ et $(\exists R_t / t=1 \dots k \text{ et } t \neq s \text{ tel que: } R_s = \subseteq)$ alors: $\text{Aggreg}(R_1, R_2, \dots R_k) = \subseteq$ c à d: $C \subseteq D$.
4. Si $(\forall R_s \in \{R_1, R_2, \dots R_k\} / s = 1 \dots k, \text{ tel que: } R_s = \equiv)$ et $(\exists R_t / t=1 \dots k \text{ et } t \neq s \text{ tel que: } R_s = \supseteq)$ alors: $\text{Aggreg}(R_1, R_2, \dots R_k) = \supseteq$ c à d: $C \supseteq D$.
5. Si $(\exists R_s \in \{R_1, R_2, \dots R_k\} / s = 1 \dots k, \text{ tel que: } R_s = \& \text{ ou } R_s = \equiv \text{ ou } R_s = \supseteq \text{ ou } R_s = \subseteq)$ alors: $\text{Aggreg}(R_1, R_2, \dots R_k) = \&$ c à d: $C \& D$.

Il est facile de prouver la proposition 1 en utilisant la notion d'interprétation des LDs: Soient $R_1, R_2, \dots R_n$, les relations sémantiques déduites entre les concepts C_i et D_j .

1. $\exists R_s \in \{R_1, R_2, \dots R_k\} / s=1 \dots k, \text{ tel que: } R_s = \perp \Rightarrow \exists C_v / v = 1 \dots i \text{ et } \exists D_u / u = 1 \dots j, \text{ tel que: } C_v \perp D_u$
 $\Rightarrow C^I_v \cap D^I_u = \emptyset, (\forall I)$
 $\Rightarrow (\cap C_1 \dots C_i)^I \cap (\cap D_1 \dots D_j)^I = \emptyset$ (C et D ne sont que des conjonctions)
 $\Rightarrow C^I \cap D^I = \emptyset$
 $\Rightarrow C \perp D$
2. $\forall R_s \in \{R_1, R_2, \dots R_k\} / s=1 \dots k, R_s = \equiv \Rightarrow \forall C_v / v = 1 \dots i \text{ et } \forall D_u / u = 1 \dots j, C_v \equiv D_u$
 $\Rightarrow C^I_v = D^I_u, (\forall I)$
 $\Rightarrow (\cap C_1 \dots C_v \dots \cup C_n \dots C_i)^I = (\cap D_1 \dots D_U \dots \cup D_m \dots D_j)^I$
 $\Rightarrow C^I = D^I$
 $\Rightarrow C \equiv D$

$$3. (\forall R_s \in \{R_1, R_2, \dots, R_k\} / s=1 \dots k, R_s = \equiv) \Rightarrow (\forall C_v / v = 1 \dots i \text{ et } \forall D_u / u = 1 \dots j, C_v \equiv D_u) \quad (1)$$

$$(\exists R_t / t=1 \dots k \text{ et } t \neq s, \text{ tel que: } R_s = \subseteq) \Rightarrow (\exists C_o / o = 1 \dots i \text{ et } \exists D_p / p = 1 \dots j, \text{ tel que: } C_v \subseteq D_u) \quad (2)$$

$$(1) \text{ et } (2) \Rightarrow C^I_v = D^I_u \text{ et } C^I_o \subset D^I_p, (\forall I)$$

$$\begin{aligned} &\Rightarrow (\cup C_1 \dots C_i)^I = (\cup D_1 \dots D_j)^I \text{ et } (C^I_o \subset D^I_p) \\ &\Rightarrow (\cup C_1 \dots C_i)^I \cup / \cap C^I_o \subset (\cup D_1 \dots D_j)^I \cup / \cap D^I_p \\ &\Rightarrow (\cap C_1 \dots C_v \dots \cup C_o \dots C_i)^I = (\cap D_1 \dots D_u \dots \cup D_p \dots D_j)^I \\ &\Rightarrow C^I \subseteq D^I \\ &\Rightarrow C \subseteq D \end{aligned}$$

4. De la même façon, nous pouvons prouver le quatrième résultat en inversant la relation de subsumption.

$$5. \exists R_s \in \{R_1, R_2, \dots, R_k\} / s=1 \dots k, \text{ tel que: } R_s = \equiv \Rightarrow \exists C_v / v = 1 \dots i \text{ et } \exists D_u / u = 1 \dots j, \text{ tel que: } C_v \equiv D_u$$

$$\Rightarrow C^I_v = D^I_u, (\forall I)$$

$$\Rightarrow (\cap C_1 \dots C_v \dots \cup C_n \dots C_i)^I \cap (\cap D_1 \dots D_u \dots \cup D_m \dots D_j)^I \neq \emptyset$$

$$\Rightarrow C^I \cap D^I \neq \emptyset$$

$$\Rightarrow C \& D$$

De la même façon, nous pouvons prouver les autres combinaisons ($R_s = \equiv$ or $R_s = \supseteq$ or $R_s = \subseteq$).

Dans l'exemple précédent, nous comparons les deux concepts: algérien_membre et parent. Dans le concept x, nous avons: les concepts personne et humaine sont similaires, et dans la base des cas nous avons : les rôles ensemble et enfant sont différents. En effet, les concepts Algérien_ membre et parent se superposent (selon la règle 5).

3.2.4. La révision

La révision des connaissances extraites peut produire des unités de connaissances. Dans notre exemple, on remarque que l'application de la règle 1 est possible sur les concepts: équipe et parent c à d: le concept ensemble est le subsumant du concept équipe et le concept humain est le subsumant des concepts parent. Les concepts: ensemble et humain sont différents. Donc équipe et parent sont aussi différents.

3.2.5. La mémorisation

L'opération d'adaptation génère de nouveaux cas qui doivent être rajoutés à la base de cas après la révision. L'étape de la mémorisation consiste à sauvegarder les nouveaux cas:

(Rel (humain, homme), \supseteq)

(Rel (humain, Algérien _membre), \supseteq)

(Rel (personne, homme), \supseteq)

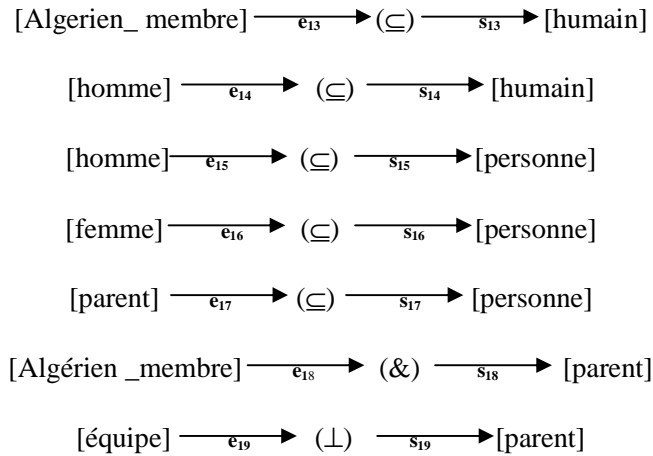
(Rel (personne femme), \supseteq)

(Rel (personne, parent), \supseteq)

(Rel (parent, Algérien _membre), $\&$) et

(Rel (parent, equipe), \perp).

Cette mémorisation est implémentée comme suit:



De la même façon, nous déduisons les relations sémantiques entre les différents autres concepts des deux ontologies O_1 et O_2 .

4. Conclusion

La détection des relations sémantiques entre les différentes entités sémantiques des ontologies à intégrer représente le cœur du processus d'appariement par ce qu'elle permet la construction de la hiérarchie de l'ontologie globale.

Dans ce travail nous avons proposé un système qui permet la détection de ces relations pour intégrer des ontologies décrites en logiques de description. Ce système est basé sur le raisonnement à partir de cas qui permet la combinaison des relations déjà trouvées entre les subsumants des entités sémantiques en question pour détecter les relations sémantiques entre eux. Ici, la déduction de ces relations est faite par l'agrégation ou la composition des relations des subsumants. Les résultats obtenus par ces opérations sont validés par les mécanismes des LDs.

Nous avons utilisé dans ce travail les techniques des logiques de descriptions, en particulier, la subsumption qui a permis la remémoration des cas sources.

Aspects techniques pour l'implantation et étude de *cas*

1. Introduction

Après avoir effectué la conception de notre système basé sur un mécanisme du RÀPC pour la détection des relations sémantiques entre les entités sémantiques des ontologies, nous allons à présent entamer la réalisation d'une partie de ce système. Cette implémentation permet de valider la contribution proposée en l'appliquant dans une étude de cas appropriée.

Dans ce chapitre, nous nous focalisons sur deux principaux axes :

- 1) Les aspects techniques de l'implantation de notre contribution.
- 2) L'application de l'opération de composition dans une étude de cas.

Nous commençons par présenter rapidement les technologies utilisées, à savoir, *Protégé-OWL*, l'environnement de développement et les frameworks spécialisés telle que *OWL API* pour la gestion de l'ontologie *OWL*. Ces aspects sont importants dans la mise en œuvre du prototype de l'application. Ensuite, nous présentons quelques expériences effectuées sur deux ontologies réelles décrites en *OWL* pour évaluer l'efficacité de l'opération de composition, et pour vérifier que la solution proposée renvoi des résultats complets et précis. Les résultats obtenus semblent très prometteurs, en comparant avec les mappings d'un expert du domaine. Le chapitre se termine par une conclusion.

2. Aspects techniques pour l'implantation

Avant de commencer l'implémentation de l'opération de composition que nous avons proposée, nous allons tout d'abord spécifier les outils utilisés qui nous ont semblés être un bon choix de part les avantages qu'ils offrent.

2.1 Choix de l'éditeur d'ontologies

Pour l'implémentation de nos ontologies, nous avons opté pour l'éditeur d'ontologies *Protégé-OWL*. Plusieurs raisons ont motivé notre choix :

- *Protégé-OWL* est un éditeur open source et gratuit.
- *Protégé-OWL* permet d'importer et d'exporter des ontologies dans les différents langages d'implémentation d'ontologies (*RDF-Schéma*, *OWL*, *DAML*, *OIL*,...etc.)
- *Protégé-OWL* possède une interface modulaire, ce qui permet son enrichissement par des modules additionnel (plugins).

- *Protégé-OWL* permet l'édition et la visualisation d'ontologies.
- *Protégé-OWL* permet le contrôle de la cohérence de l'ontologie par des vérifications de contraintes.
- *Protégé-OWL* est fourni avec une *API* écrite en *JAVA*, qui permet de développer des applications pouvant accéder aux ontologies de Protégé et de les manipuler (voir annexe).

2.2 Langage de programmation et frameworks spécialisés

La programmation en *JAVA* nécessite seulement le compilateur et la machine virtuelle associée (groupés dans le *SDK*). Cependant il est vivement recommandé d'utiliser un environnement de développement qui assiste le développeur et facilite le codage. Plusieurs environnements sont disponibles pour le langage *JAVA* (plus de 20 environnements). Toutefois, il existe trois principaux acteurs : *Code Gear JBuilder* qui est payant, *SUN NetBeans* et *IBM Eclipse for JAVA* qui sont libres.

Notre choix est porté sur *Eclipse for Java*. En fait, *Eclipse* est un environnement complet avec des possibilités d'extension énormes via les composants extensibles (*Plug-ins*).

Eclipse possède de nombreux points forts qui sont à l'origine de son énorme succès dont les principaux sont :

- Support de plusieurs plateformes d'exécution: *Windows, Linux, Mac OS X, ...*
- Une plate-forme ouverte pour le développement d'applications et extensible grâce à un mécanisme de *plug-ins*.
- Plusieurs versions d'un même plugin peuvent cohabiter sur une même plateforme.
- Un support multi langage grâce à des *plug-ins* dédiés : *Cobol, C, PHP, C#, ...*
- Malgré son écriture en *Java*, *Eclipse* est très rapide à l'exécution grâce à l'utilisation de la bibliothèque *SWT* (Standard Widget Toolkit) est développé en *Java* par *IBM* en utilisant au maximum les composants natifs fournis par le système d'exploitation sous jacent.
- Un historique local des dernières modifications.
- Une exécution des applications dans une *JVM* dédiée sélectionnable avec possibilité d'utiliser un débogueur complet (points d'arrêts conditionnels,

visualiser et modifier des variables, évaluation d'expression dans le contexte d'exécution, changement du code à chaud avec l'utilisation d'une *JVM 1.4*, ...)

- Propose le nécessaire pour développer de nouveaux *plug-ins*.
- La plate-forme est entièrement internationalisée dans une dizaine de langue sous la forme d'un *plug-in* téléchargeable séparément.

Les frameworks spécialisées sont celles qui apportent des fonctionnalités spécifiques dans un contexte particulier. Elles peuvent être livrées ou non avec le langage utilisé. Dans notre cas, nous avons besoin de deux frameworks, La première pour la gestion des ontologies et la deuxième pour la vérification de l'ontologie de l'alignement. Dans ce travail, on utilise la framework *OWL API* (*OWL API* 2010) et le raisonneur *FaCT⁺⁺*.

Nos ontologies sont implémentées en langage *OWL (ontology Web Language)*, or les fichiers *OWL* sont inexploitable en état brut car leur structure est très complexe. Donc pour pouvoir l'exploiter il nous a fallu un « traducteur » capable de traduire les balises et la sémantique véhiculée par le fichier *OWL* en objet manipulable par des programmes.

Un *API* (*Application Programming Interface*) est une interface de programmation d'application, un jeu de fonctions ou de méthodes, utilisé pour accéder à certaines fonctionnalités. Cette interface permet d'accéder aux services d'une application depuis un langage de programmation (et non pas à travers une interface utilisateur traditionnelle). Un *OWL API* est donc une interface fournissant des fonctionnalités traitant le *OWL*.

OWL API propose les opérations nécessaires pour la création, la modification et l'interrogation de l'ontologie. *OWL API* supporte les ontologies en *OWL*, ce qui motive son utilisation dans notre implémentation.

Un autre avantage de *OWL API* est le mécanisme de raisonnement intégré. Il permet de profiter de l'aspect sémantique de l'ontologie *OWL* en exploitant les relations sémantiques de ce dernier.

Pour la vérification de l'ontologie de l'alignement, il existe différentes implantations du raisonnement au travers d'outils performants. Nous avons choisi le moteur d'inférence *FaCT (Fast Classification of Terminologies)* qui est un nouveau raisonneur LD conçu par des chercheurs de l'université de Manchester (Royaume-Uni) pour fournir une plate-forme d'expérimentation de nouveaux algorithmes d'aide à la décision. Il utilise la méthode des tableaux et comprend deux raisonneurs, l'un base sur la logique de description *SHIF* et l'autre base sur *SHIQ*. *FaCT⁺⁺* est une nouvelle version de *FaCT* implantée en *C++*.

3. Etude de cas

Afin de motiver nos recherches d'un point de vue pratique, nous proposons d'appliquer l'opération de composition dans une étude de cas appropriée. La suite de cette section décrit le fonctionnement réel d'une application dédiée à l'intégration des ontologies décrites en logique de descriptions.

Nous avons choisi deux parties de deux ontologies qui sont du même domaine pour pouvoir représenter les cas spéciaux, afin de présenter une étude de cas qui concrétise la méthode proposée.

3.1. Présentation de l'étude de cas

Dans cette section, nous effectuons une mise en œuvre de l'opération de composition sur deux ontologies O_1 et O_2 (figures 4.1 et 4.2). Ces ontologies sont mises en œuvre avec l'éditeur d'ontologie bien connu: *protegeOWL version 4.1_beta*.

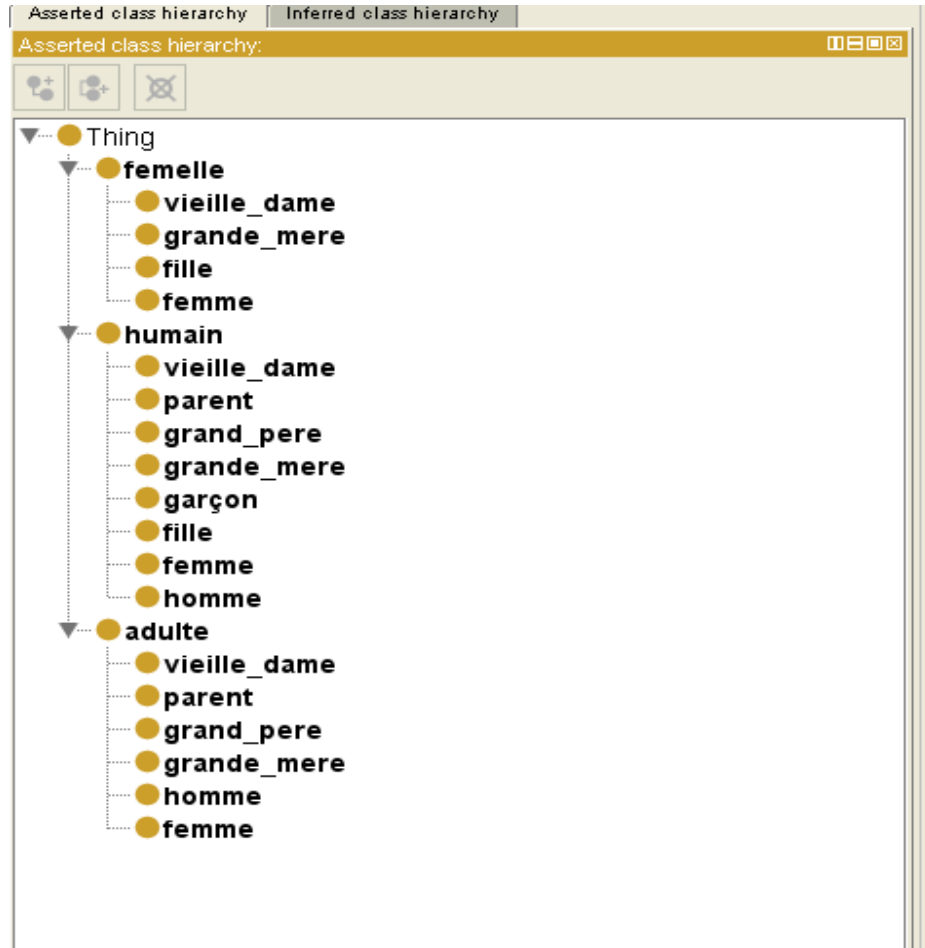


Figure 4.1– La hiérarchie des concepts de l'ontologie O_1 avec *Protégé OWL*

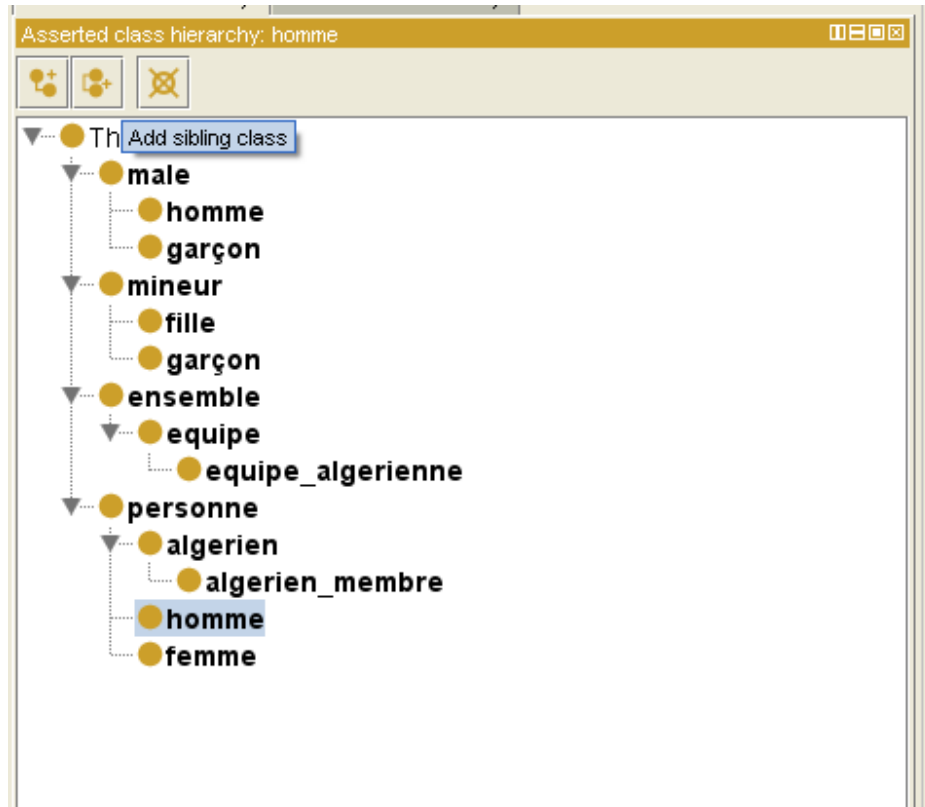


Figure 4.2– La hiérarchie des concepts de l'ontologie O_2 avec Protégé OWL

Comme nous l'avons déjà montré dans le chapitre 3, notre système doit être utilisé avec des cas connus afin d'instancier la base des cas (l'ontologie de l'alignement O_A) et fournir une base de raisonnement. Pour faire ça, nous prenons les entités définies des ontologies O_1 et O_2 , qui sont: les concepts: humain, femelle et adulte de O_1 et les concepts: personne, mâle, mineur et ensemble de O_2 . Nous les comparons avec la distance sémantique.

Les résultats de la comparaison sont comme suit :

$$\begin{aligned}
 & [personne] \xrightarrow{e_1} (\equiv) \xrightarrow{s_1} [humain] \\
 & [ensemble] \xrightarrow{e_2} (\perp) \xrightarrow{s_2} [humain] \\
 & [ensemble] \xrightarrow{e_3} (\perp) \xrightarrow{s_3} [femelle] \\
 & [m\^ale] \xrightarrow{e_4} (\perp) \xrightarrow{s_4} [femelle]
 \end{aligned}$$

$[mineur] \xrightarrow{e_5} (\perp) \xrightarrow{s_5} [adulte]$

$[ensemble] \xrightarrow{e_6} (\perp) \xrightarrow{s_6} [adulte]$

Ainsi, ces cas sont mis en œuvre avec *Protégé OWL*, comme suit (figure 4.3 et 4.4):

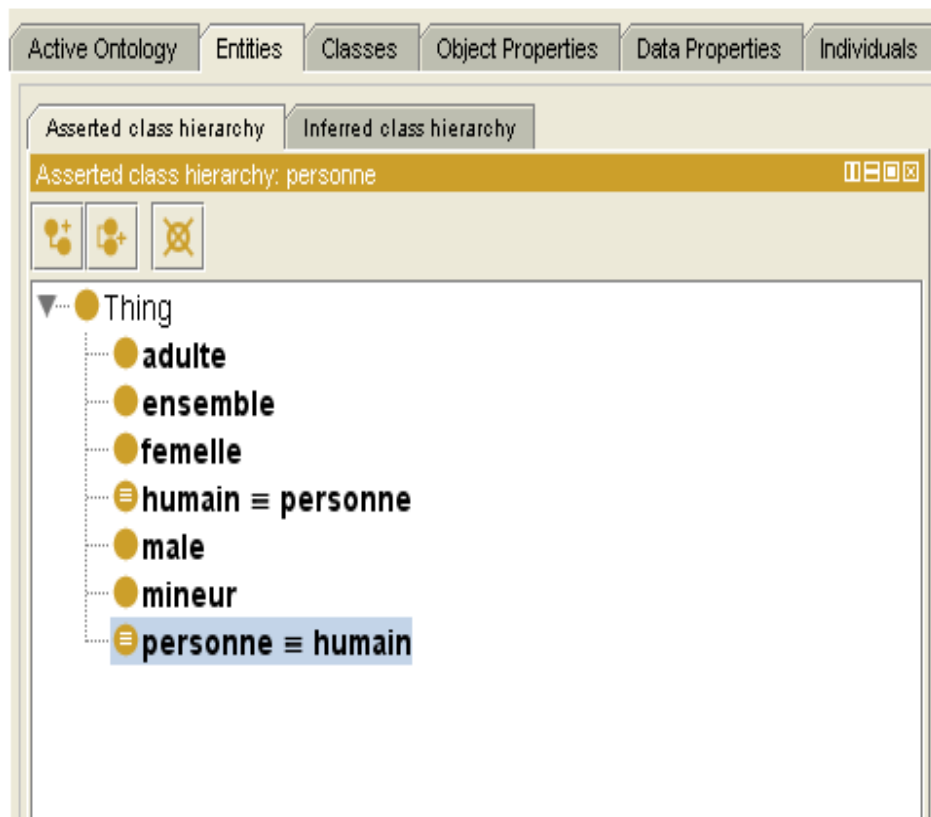


Figure 4.3– La hiérarchie des concepts de l'ontologie O_A avec *Protégé OWL*



Figure 4.4– La traduction des cas de l'ontologie O_A avec le raisonneur $FaCT^{++}$.

4. Prototype

Le prototype que nous avons implémenté découle de l'algorithme suivant (Kolli et Boufaïda 2011):

Algorithme 1: Composition des relations sémantiques.

Données :

- 1) O_1 et O_2 : deux ontologies à aligner
- 2) O_A : ontologie initiale d'alignement

Résultat : O_A : ontologie finale d'alignement

début

/* parcours des classes de l'ontologie O_1 */

pour chaque ($cls1 \in O1$) **faire** {

```

/* lire les subsumants de cls1*/
pour chaque (sup1 ∈ super_class(cls1)) faire {
/* parcours des classes de l'ontologie O2 */
pour chaque (cls2 ∈ O2) faire {
/* lire les subsumants de cls2*/
pour chaque (sup2 ∈ super_class(cls2)) faire {
LIRE les définitions des concepts sup1 et sup2 dans l'ontologie OA
si (sup1 est équivalentes à sup2) alors {
/* Ajouter cls1 comme subsumé de sup2 et cls2 comme subsumé de sup1 à
OA*/
AJOUTER((cls1, sup2), ⊆, OA)
AJOUTER((cls2, sup1), ⊆, OA)
}
Sinon {
Si (sup1 est différentes à sup2) alors {
/* cls1 et cls2 sont différentes*/
AJOUTER((cls1, cls2), ⊥, OA)
}
Sinon {
Si (sup1 est superposé avec sup2) alors
/* cls1 et cls2 sont superposées */
AJOUTER((cls1, cls2), &, OA)
}
Sinon {
Si (sup1 est subsumé par sup2) alors
/* cls1 est subsumé par sup2 */
AJOUTER((cls1, sup2), ⊆, OA)
}
Sinon {

```

```

    Si (sup2 est subsumé par sup1) alors
    /* cls2 est subsumé par sup1 */
    AJOUTER((cls2, sup1),  $\subseteq$ ,  $O_A$ )
    }
    Retourner ( $O_A$ )
    }}}}
fin

```

L'algorithme 1 permet de composer les relations sémantiques entre des concepts pour déduire de nouvelles relations sémantiques entre d'autres concepts de deux ontologies. Il prend en entrée les deux ontologies O_1 et O_2 à aligner, décrites en LDs, ainsi que l'ontologie d'alignement initiale O_A , et donne en retour l'ontologie d'alignement finale O_A . Il prend deux concepts **cls1** et **cls2** qui appartiennent respectivement, aux ontologies O_1 et O_2 . Il extrait leurs subsumants respectivement **sup1** et **sup2**. Puis, selon la relation entre **sup1** et **sup2** définie dans l'ontologie d'alignement O_A , il déduit la relation entre **cls1** et **cls2**, en respectant la table 3.1 du chapitre 3. En fin, il ajoute le cas résultant à l'ontologie O_A .

Dans ce qui suit, nous présentons une partie du programme de composition (table 4.1), qui met en œuvre le cas: $((C_1 \equiv C_2 \text{ et } C_2 \subseteq C_3) \Rightarrow (C_1 \subseteq C_3))$ de la table 3.1 du chapitre précédent. L'interprétation de ce cas est comme suit: si le concept C_3 est subsumé par le concept C_2 et ce dernier est similaire au concept C_1 alors : C_3 est aussi subsumé par le concept C_1 .

Etant donné deux ontologies O_1 et O_2 à comparer, notre programme de composition mis en œuvre en langage : *OWL API* utilise l'ontologie O_A pour déduire le relations sémantiques entre les différents concepts des deux ontologies O_1 et O_2 .

Table 4.1 : Une partie du programme de l'opération de composition.

```

Program 1.

1  for(OWLClass cls1 : O1.getReferencedClasses())
    {

```

```

2 Set<OWLDescription> superc = cls1.getSuperClasses(O1);
3 for(OWLDescription desc : superc)
4     {
5         for(OWLClass cls2 : O2.getReferencedClasses())
6             {
7                 Set<OWLDescription> superc2 = cls2.getSuperClasses(O2);
8                 for(OWLDescription desc2 : superc2)
9                     {
10                        for(OWLClass cls : OA.getReferencedClasses())
11                            {
12                                if ( cls == desc )
13                                    {
14                                        Set<OWLDescription> equi = cls.getEquivalentClasses(OA);
15                                        for(OWLDescription descequi : equi)
16                                            {
17                                                if ( descequi == desc2)
18                                                    {
19                                                        OWLClass clsA = factory.getOWLClass(URI.create(OAURI+"#" +cls1));
20                                                        OWLClass clsB = factory.getOWLClass(URI.create(OA URI+"#" +cls2));
21
22                                                        OWLAxiom axiom = factory.getOWLSubClassAxiom(clsB, cls);
23                                                        OWLAxiom axiom1 = factory.getOWLSubClassAxiom(clsA, cls);
24
25                                                        AddAxiom addAxiom = new AddAxiom(OA, axiom);
26                                                        AddAxiom addAxiom1 = new AddAxiom(OA, axiom1);
27                                                        manager.applyChange(addAxiom);
28                                                        manager.applyChange(addAxiom1);
29                                                        manager.saveOntology(OA);
30                                                    }
31                                                }
32                                            }
33                                        }
34                                }
35                            }
36                        }
37                    }
38                }
39            }
40        }
41    }

```

Le programme 1 fonctionne de la manière suivante: il prend deux classes des ontologies O_1 et O_2 respectivement, **cls1** et **cls2** en utilisant la fonction: *getReferencedClasses()* (voir les lignes 1 et 4 du programme 1). La fonction: *GetSuperClasses()* (voir les deux lignes 2 et 5) permet d'extraire pour chaque classe, ses subsumants. Ensuite, le programme 1 vérifie si les subsumants de la classe **cls1** sont équivalents aux subsumants de la classe **cls2** en appliquant la fonction *getEquivalentClasses()* sur les définitions des subsumants de ces deux classes dans l'ontologie O_A (voir, la ligne 9).

Après avoir vérifié l'équivalence des subsumants, programme 1 ajoute **cls1** comme subsumé des subsumants de la classe **cls2** et ajoute **cls2** comme subsumé des subsumants de la classe **cls1** à la définition de ces subsumants dans l'ontologie d'alignement O_A en utilisant l'axiome: *OWLSubClassAxiom* (voir les lignes 11 jusqu'à 20).

Dans l'exemple précédent, les concepts: femme, homme, fille, garçon, parent, grand père, grande mère et vieille dame sont subsumés par le concept: humain et les concepts: algérien, homme et femme sont subsumés par le concept: personne.

Selon l'ontologie O_A (figure 4.4), les concepts : personne et humain sont équivalents. En effet, les subsumés du concept personne sont les subsumés du concept humain et vice versa. Ainsi, le concept personne subsume les concepts: femme, homme, fille, garçon, parent, grand père, grande mère et vieille dame et le concept humain subsume les concepts: algérien, homme et femme.

Comme nous l'avons déjà mentionné, les résultats du programme de composition sont mémorisés dans l'ontologie d'alignement O_A . Ainsi, après l'exécution du programme 1 sur notre exemple, l'ontologie O_A sera comme suit (figure 4.5):

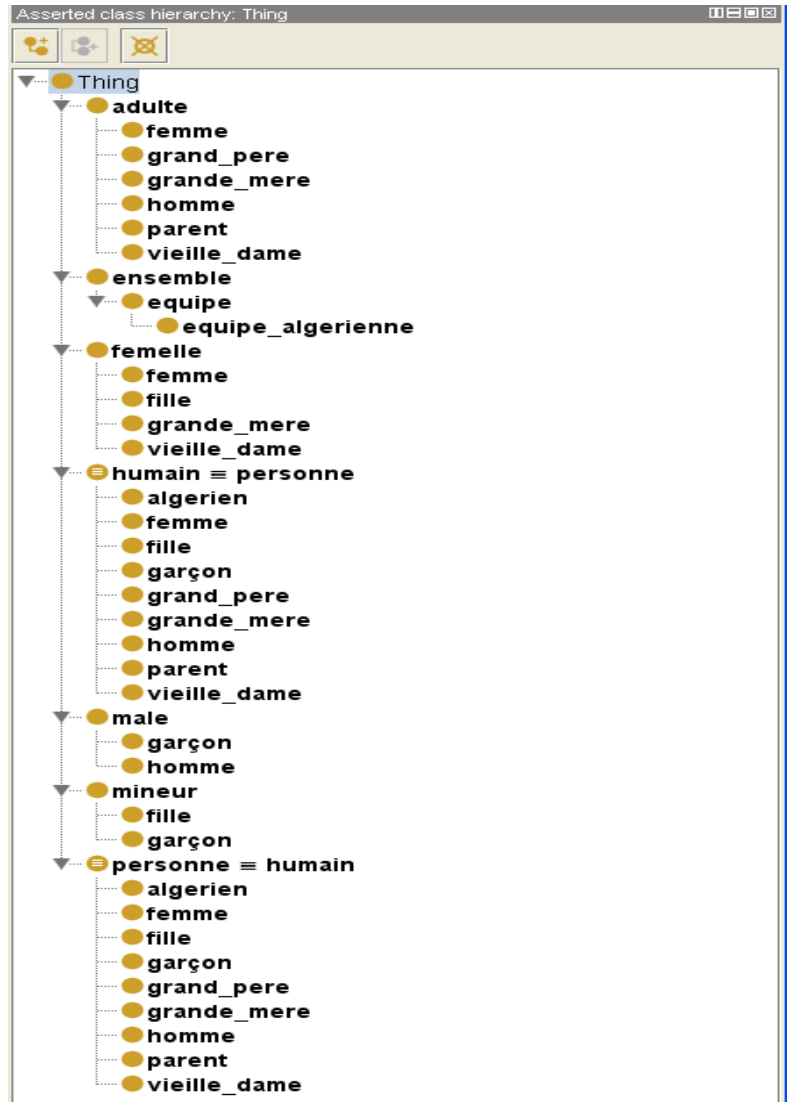


Figure 4.5– La hiérarchie des concepts de l'ontologie O_A obtenue après l'exécution du programme de composition.

5. Test et vérification

En ce qui concerne la méthode de test, pour évaluer la qualité des résultats de l'appariement, toutes les paires d'entités sémantiques ont été appariées manuellement pour avoir des mappings de l'expert. Les résultats générés par notre programme ont été comparés avec les mappings de l'expert. La fenêtre à gauche (Figure 4.6) montre les cas des mappings déduits après l'exécution du programme de composition. On

peut aussi dire que notre ontologie est valide puisque tous les cas sont identiques aux mappings de l'expert.

Nous constatons que les entités portant sur les cas présentés dans la table 3.1 sont bien appariées. Cette comparaison, bien que préliminaires, montre l'efficacité de notre approche.

La vérification de la cohérence de notre ontologie d'alignement \mathbf{O}_A a été effectuée grâce au raisonneur *FACT++* disponible sur *protegeOWL*. L'application de ce raisonneur permet de détecter les moindres anomalies et les incohérences de la hiérarchie de l'ontologie.

FACT++ est un classificateur, il permet la classification des concepts selon leur définition en se basant sur les logiques de descriptions. En effet, il permet aussi le test de classification. Comme le montre la fenêtre à droite, la hiérarchie des classes de \mathbf{O}_A déduite par *FACT++* est identique à la hiérarchie des concepts de \mathbf{O}_A après l'exécution de notre programme (Figure 4.5).

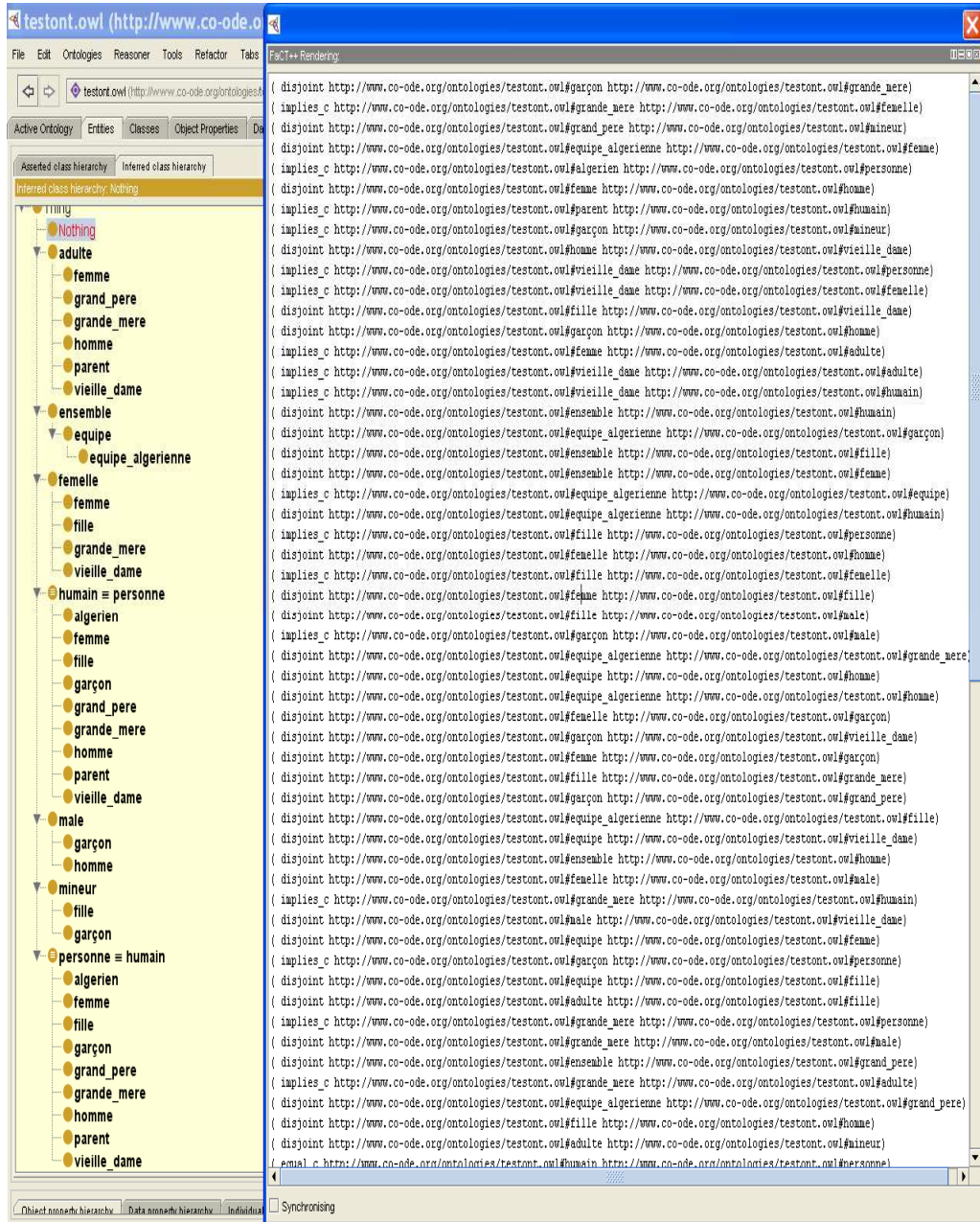


Figure 4.6– La hiérarchie des concepts de l'ontologie O_A obtenue après l'exécution du raisonneur $FaCT^{++}$.

6. Conclusion

Dans ce chapitre, nous avons présenté les aspects techniques utilisés dans notre approche, et l'application de cette dernière dans une étude de cas.

Nous avons commencé par présenter les aspects d'implémentation qui sont utilisés dans notre travail. L'implantation repose essentiellement sur *protegeOWL* pour l'édition des ontologies, le langage *JAVA* et l'environnement *Eclipse* pour le codage, *OWL API* pour la gestion et l'interrogation des ontologies *OWL* et *FaCT⁺⁺* pour la validation de l'ontologie d'alignement.

Par la suite, nous avons appliqué l'opération de composition dans une étude de cas et les résultats semblent très prometteurs, en comparant avec les mappings d'experts. A travers cette étude de cas, nous pouvons remarquer l'apport de notre contribution, en assurant une intégration sémantique des ontologies basée sur les techniques des logiques descriptions.

Conclusion générale

L'utilisation d'ontologies pour le Web Sémantique entraîne un besoin de techniques de gestion de ces ontologies, notamment pour les intégrer. Le domaine de l'intégration des ontologies est très vaste et nécessite énormément de concentration et d'approfondissement.

Le problème principal de tous les travaux sur l'interopérabilité porte sur la comparaison et le mapping des différentes ontologies. Etant donnée la nature décentralisée du développement du Web, le nombre d'ontologies est très important. Nous pouvons trouver plusieurs ontologies qui décrivent soit des domaines semblables, mais avec l'utilisation de terminologies différentes, soit des domaines complémentaires. Pour intégrer les données des ontologies distinctes, nous devons connaître les correspondances sémantiques entre leurs éléments.

La détection des relations sémantiques entre les différentes entités sémantiques des ontologies à intégrer représente le cœur du processus de l'intégration par ce qu'elle permet les échanges de manière sémantique des connaissances.

Cette thèse investit le domaine de l'intégration des ontologies dans le domaine l'ingénierie des connaissances et du Web sémantique. Notre objectif a été de tirer profit des travaux menés notamment dans le domaine de l'interopérabilité sémantique des connaissances, dans le but d'intégrer des ontologies et également d'accéder à des ressources distantes de façon transparente. Le résultat de notre travail est un système originale d'intégration des d'ontologies.

Dans la suite de cette section, nous allons résumer la contribution apportée, les limitations de notre système et donner quelques perspectives pour les travaux futurs.

1. Contributions

Dans ce contexte, plusieurs contributions ont été développées :

- **La détection des relations sémantiques entre les différentes entités sémantiques des ontologies**

Dans ce travail nous avons proposé un système qui permet la détection des relations sémantiques entre les différentes entités sémantiques pour intégrer des ontologies décrites en logiques de description. Ce système est basé sur un outil fiable qui est : le raisonnement à partir de cas.

L'utilisation de ce mécanisme ici permet la combinaison des relations déjà trouvées entre les subsumants des entités sémantiques en question pour détecter les relations sémantiques entre ces entités.

Nous avons utilisé dans ce travail les techniques d'extraction de connaissances dans des bases de données (ECBD) et les techniques des logiques de descriptions, en particulier, la subsumption et la classification qui ont permis la remémoration des cas sources.

- **La formalisation de l'appariement des ontologies avec les logiques de descriptions**

Ce travail propose une nouvelle approche formelle pour résoudre le problème de l'appariement des ontologies. Cette approche est basée sur les techniques de logiques de descriptions et la distance sémantique pour la détection des relations sémantiques entre les entités sémantiques de deux ontologies différentes.

Ici, la déduction de ces relations est faite par l'agrégation ou la composition des relations des subsumants. Les résultats obtenus par ces opérations sont validés par les mécanismes des DLs.

- **L'implantation et l'étude de cas.**

Nous avons présenté les différents aspects techniques qui sont utilisés dans l'implantation de notre application. Ces aspects sont principalement, le langage *Java* et l'environnement *Eclipse* pour le codage, *OWL API* pour la gestion et l'interrogation des ontologies *OWL* et *FaCT⁺⁺* pour la validation de l'ontologie d'alignement. Par la suite, nous avons appliqué l'opération de composition dans une étude de cas et les résultats semblent très prometteurs, en comparant avec les mappings d'experts. A travers cette étude de cas, nous pouvons remarquer l'apport de notre contribution, en assurant une intégration sémantique des ontologies basée sur les techniques des logiques descriptions.

2. Perspectives

Les travaux proposés dans cette thèse nous permettent d'ouvrir plusieurs perspectives dont on peut citer :

- **Passage à l'échelle**

Les expérimentations menées jusqu'à présent ont porté sur des ontologies de taille relativement modeste. Il est important d'évaluer nos propositions sur des ontologies de tailles plus réalistes (plusieurs centaines de concepts).

- **Problème d'hétérogénéité du langage d'ontologie**

Nous pouvons pallier ce problème en traduisant toutes les ontologies dans un langage possédant l'expressivité de tous les autres (il s'agit dans notre cas des logiques de descriptions).

- **Le développement de l'étape de maintenance**

Notre orientation principale des travaux futurs est certainement: le développement de l'étape de maintenance en appliquant les techniques d'apprentissage des connaissances afin de permettre le système proposé d'effectuer seul les opérations de maintenance.

- **Implantation du système proposé**

Afin d'évaluer et valider la contribution présentée dans ce mémoire, une phase d'implémentation tout le système proposé s'avère indispensable. Cette phase a pour objectif d'étudier les performances de notre système de mapping et de montrer ses avantages. En outre, ceci nous permettra aussi d'identifier les contraintes et les insuffisances de notre système.

Bibliographie

Bibliographie

- ACM. 2009. Association for Computing Machinery. <http://www.acm.org/>
- Aleksovski Z., Klein M., Kate W. T., Harmelen F. V., 2006. Matching Unstructured Vocabularies using a Background Ontology, 15th International Conference on Knowledge Engineering and Knowledge Management, Hong Kong, China, pp. 182-197.
- Amardeilh F., 2007. Web Sémantique et Informatique Linguistique : propositions méthodologiques et réalisation d'une plateforme logicielle. Thèse de doctorat, Université Paris X –Nanterre.
- Angele J. et Schnurr H., 2005. Do not use this gear with a switching lever! automotive industry experience with semantic guides. In GI Jahrestagung, pages 48– 52.
- Bach T. L., Dien-Kuntz R. et Gandon F., 2004. On Ontology Matching Problems - for Building a Corporate Semantic Web in a Multi-Communities Organization, *Proceedings of ICEIS*, Porto, Portugal, pp. 236-243
- Bach T. L., 2006. Construction d'un web sémantique multi-points de vue. Thèse de PhD, École des Mines de Nice à Sophia Antipolis.
- Bachimont B., 2000. Engagement sémantique et engagement ontologique : conception et réalisation d'ontologies en ingénierie des connaissances. Ingénierie des connaissances. Évolution Récentes et nouveaux défis Paris: Eyrolles, France, pp. 305-323.
- Barwise J. et Seligman J., 1997. Information Flow: The Logic of Distributed Systems. Cambridge University Press, Number 44.
- Berners-Lee T., 1999. Weaving the Web. Harper, San Francisco.

- Berners-Lee T., Hendler J. et Lassila O., 2001. The semantic web: A new form of web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American*, 3: 35–43.
- Bennacer N. et Aufaure M., 2006. www.supelec.fr/d2ri/RR_06_07/RR_06_07_Chap7.pdf
- Blázquez M., Fernández M., García-Pinar J. M. et Gómez p. A., 1998. Building Ontologies at the Knowledge Level using the Ontology Design Environment. <http://ksi.cpsc.ucalgary.ca/KAW/KAW98/blazquez/>
- Bouquet P., Giunchiglia F., Harmelen F., Serafini L. et Stuckenschmidt H., 2004. Contextualizing ontologiesstar, open. *Web Semantic: Science, Services and Agents on the World Wide Web*, 1:325–343.
- Borst W. N., 1997. Construction of Engineering Ontologies. Thèse de doctorat, Center for Telematica and Information Technology, Université de Tweenty, Enschede.
- Bray T., Paoli J. et Sperberg-McQueen C., 2000. Extensible Markup Language (XML) 1.0. <http://www.w3.org/TR/REC-xml>.
- Breslin J., Harth A., Bojars U. et Decker S., 2005. Towards Semantically-Interlinked Online Communities. *Lecture Notes in Computer Science*, 3532: 500–514.
- Bruijn J., Ehrig M., Feier C., Recuerda F. M., Scharffe F. et Weiten M., 2006. Semantic Web Technologies, trends and research in ontology-based systems, chapter Ontology Mediation, Merging, and Aligning, WILEY, pp. 95–113.
- Buchheit M., Donini F., et Shaerf A., 1993. Decidable reasoning in terminological knowledge representation systems. *Journal of Artificial Intelligence Research*, 1: 109–138.
- Caracciolo C., J. Euzenat, L. Hollink, R. Ichise, A. Isaac, V. Malaisé et al., 2008. Results of the Ontology Alignment Evaluation Initiative 2008. www.dit.unitn.it/~p2p/OM-2008/oaiei08_paper0.pdf
- Charlet J., Bachimont B. et Troncy R., 2004. Ontologies pour le Web Sémantique. *Série de la Revue Information - Interaction - Intelligence (I3)*, 4 : 96-100.

- Charlet J., B. Bachimont et R. Troncy, 2005. Le Web sémantique, vol. 4 of Hors série de la revue Information - Interaction - Intelligence, Cépaduès, 4 : 7-20.
- Chawk M., 1996. Modèle de graphes conceptuels et représentation sémantique du langage naturel. www.revue-texto.net/marges/marges/Documents%20Site%206/doc0004_chawk_m/graphic.pdf
- Corcho O., Fernandez-Lopez M. et Gomez P. A., 2002. OntoWeb Technical Roadmap, Version 1.0. <http://www.ontoweb.org>.
- Corcho O., Fernandez-Lopez M. et Gomez-Pérez A., Methodologies, 2003. tools and languages for building ontologies : Where is their meeting point ?. Data & Knowledge Engineering, Elsevier, 46: 41-64.
- Cordier A. Fuchs B. et Mille A., 2006. De l'Ingénierie à l'Apprentissage des connaissances d'adaptation en Raisonnement à Partir de Cas, http://www.sdc2006.org/cdrom/contributions/Cordier_et_al_IC06.pdf
- Cullot N. et F. Jouanot, 2003. Context Comparison for Object Fusion. The 15th Conference on Advanced Information Systems Engineering CAiSE'03. Klagenfurt/Velden, Austria, pp. 536-551.
- D'Aquin M., 2006. Raisonnement à partir de cas décentralisé pour le Web sémantique. <http://hal.inria.fr/docs/00/06/57/13/PDF/daquin-final.pdf>
- D'Aquin M., F. Badra, S. Lafrogne, J. Lieber, A. Napoli et L. Szathmary, 2006. Adaptation Knowledge Discovery from a Case Base. Proceedings of the 17th European Conference on Artificial Intelligence (ECAI-06). Trento, Italy, pp. 795-796.
- Delteil A., 2002. Représentation et apprentissage de concepts et d'ontologies pour le Web Sémantique. Thèse de doctorat, université de Nice Sophia Antipolis (UNSA).
- Djaghoul Y., 2007. Intégration des ressources Web dans un environnement P2P, basée sur les ontologies et la gestion de la confiance. Thèse de doctorat, Université Mentouri de constantine.

- Dieng R., Corby O., Gandon F., Giboin A., Golebiowska J., Matta N. et Ribière M., 2001. Méthodes et outils pour la gestion des connaissances : une approche pluridisciplinaire pour le knowledge management, 2ème édition, Paris: Dunod.
- Doan A., Madhavan J., Domingos P. et Halevy A., 2003. Ontology matching: A machine learning approach. Handbook on Ontologies in Information Systems. Springer-Verlag.
- Doan A., J. Madhavan, P. Domingos et A. Halevy, 2002. Learning to map between ontologies on the semantic web. Proceedings of the Eleventh International World Wide Web Conference. Hawaii, USA, pp. 662-673.
- Dou D., Mcdermott D. et Qi P., 2002. Ontology translation by ontology merging and automated reasoning. <http://cs-www.cs.yale.edu/homes/dvm/papers/DouMcDermottQi02.pdf>
- Ehrig M. et Staab S., 2004. Qom quick ontology mapping. International Semantic Web Conference (ISWC2004), Japan, pp. 683-697.
- Elbyed A., 2009. ROMIE, une approche d'alignement d'ontologies à base d'instances. Thèse de doctorat de l'INSTITUT NATIONAL DES TELECOMMUNICATIONS dans le cadre de l'école doctorale S&I en co-accréditation avec l'UNIVERSITE D'EVRY-VAL D'ESSONNE.
- Euzenat J., M. Mochol, P. Shvaiko, H. Stuckenschmidt, O. Svab, V. Svatek et al., 2006. First results of the ontology alignment evaluation initiative 2006. Proceedings of the ISWC workshop on Ontology Matching. Athens, Greece, pp. 73-95.
- Euzenat J., Isaac, C. Meilicke, P. Shvaiko, H. Stuckenschmidt, O. Svab et al., 2007. Results of the Ontology Alignment Evaluation Initiative 2007. International Semantic Web Conference 2007. Busan, Korea.
- Fensel D., Horrocks I., van Harmelen F. et De S., 2000. Oil in a nutshell. In Proceedings of European Knowledge Acquisition Workshop (EKAW'2000), 1937: 1-16.
- Fürst F., 2002. L'ingénierie Ontologique. Rapport de recherche, Institut de Recherche en Informatique de Nantes, France.

- Gaëlle L., 2002. Etat de l'art: Ontologies et Intégration/Fusion d'ontologies, casanuestra.free.fr/ontologie_fusion_lortal.pdf
- Giunchiglia F., Sshvaiko P. et yatshevich M., 2005. S-match: an algorithm and an implementation of semantic matching. Semantic Interoperability and Integration conference, Schloss Dagstuhl, Germany, pp. 347-365.
- Gomez P.A., 1999. Développements récents en matière de conception, de maintenance et d'utilisation d'ontologies. 3èmes rencontres Terminologie et intelligence artificielle TIA. 19 :9-20.
- Gómez P. A., 1999a. Ontological Engineering: A state of the art. Expert Update, 2: 33-43.
- Gómez P. A., 1999b. Tutorial on Ontological Engineering. <http://www.dsv.su.se/ijcai-99/>
- Gomez P. A. et Benjamins V.R., 1999. Overview of Knowledge Sharing and Reuse Components: Ontologies and Problem-Solving Methods. <http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-18/>.
- Guarino N., et Giaretta P., 1995. Ontologies and Knowledge Bases: Towards a Terminological Clarification. In Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing, Mars N. J. I., Amsterdam: IOS Press. pp. 25–32.
- Guarino N., 1997. Semantic Matching: Formal Ontological Distinctions for Information Organization, Extraction, and Integration. In Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology. Springer Verlag, pp. 139-170.
- Guarino N., 1997a. Understanding, building and using ontologies. International J. Human-Computer Studies, 46: 293-310.
- Guarino N., 1997b. Some organizing principles for a unified top-level ontology. AAAI Spring Symposium on Ontological Engineering. Stanford, CA, pp. 57-63.
- Gruber T., 1992. "Ontolingua: A mechanism to Support Portable Ontologies". <http://www-ksl.stanford.edu/knowledge-sharing/papers/README.html#ontolingua-long>

- Gruber T., 1993. A translation approach to portable ontology specifications. *Knowledge Acquisition Journal*. 5:199 -220.
- Gruber T. et Olsen G.R., 1994. An ontology for engineering mathematics. *Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning*. Bonn, Germany, Morgan Kaufmann, pp. 258-269.
- Hage W., Katrenko S. et Schreiber G., 2005. A Method to combine Linguistic Ontology-Mapping Techniques, *International Semantic Web Conference*, Sardinia, Italy, pp. 732-744.
- Hendler J. et Mcguinness D., 2001. The Darpa Agent Markup Language. <http://www.daml.org>.
- Hernandez N., 2005. Ontologies de domaine pour la modélisation du contexte en recherche d'information. Thèse de doctorat, université Paul Sabatier de Toulouse.
- Horrocks I., 1998. The FaCT system. *Proceeding of the 2nd Int. Conf. on Analytic Tableaux and Related Methods (TABLEAUX'98)*, Lecture Notes in Artificial Intelligence, 1397: 307-312.
- Inaba A., Supnithi T., Ikeda M., Mizoguchi R. et Toyoda J. 2000. An Overview of Learning Goal Ontology. *Proceeding of ECAI2000 Workshop on Analysis and Modelling of Collaborative Learning Interactions*. Berlin, Germany, pp. 23-30.
- Jenhani, O., 2003. Ontologies pour le WEB: relations, construction d'ontologies et méthodes de raisonnement pour la génération de langue naturelle. Rapport ARC INRIA GeNI, France.
- Jin L., Chen W., Hayashi Y., Ikeda M., Mizoguchi R., Takaoka Y. et Ohta M. 1999. An Ontology-Aware Authoring Tool - Functionalstructure and guidance generation. *Proceedinf of AIED1999*, Le Mans, France, pp. 85-92.
- Kalfoglou Y. et Schorlemmer M., 2003. If-map: an ontology mapping method based on information flow theory. *Journal on Data Semantics*, 1:98-127.

- Kassel G., 2002. OntoSpec : une méthode de spécification semi-informelle d'ontologies. In Actes des journées francophones d'Ingénierie des Connaissances, Rouen, France, pp. 75-87.
- Kavouras M., 2003. A unified ontological framework for semantic integration. International Workshop on Next Generation Geospatial Information, Cambridge, UK, <http://www.ntua.gr/ontogeo/publications/boston-kavouras.pdf>
- Kifer M., Lausen G. et Wu J., 1995. Logical foundations of object-oriented and frame-based languages. In Journal of the ACM. 42:741-843.
- Klein M., 2001. Combining and relating ontologies: an analysis of problems and solutions, International Joint Conferences on Artificial Intelligence (IJCAI), Workshop on Ontologies and Information Sharing, Seattle, USA, pp. 53-62.
- Knowledge W., 2004. KnowledgeWeb, Deliverables of KWEB Project. <http://knowledgeweb.semanticweb.org/>.
- Kolli M. et Boufaïda Z., 2004. A Method for the Integration of Ontologies in the Semantic Web. Arab conference international of technology (ACIT'2004). Constantine, Algeria, pp. 665.
- Kolli, M. et Z. Boufaïda, 2009. Detecting semantic relations among ontologies with a CBR system. Proceedings of the IADIS International Conference on Applied Computing, Rome, Italy, pp. 361-368.
- Kolli, M. et Z. Boufaïda, 2010. A description logics formalization for the ontology matching. Proc. Comput. Sci., ELSEVIER, 3: 29-35.
- Kolli, M. et Z. Boufaïda, 2011. Composing semantic relations among ontologies with a description logics. Information Technology Journal, 10 : 1106-1112.
- Kolodner J., 1998. Case-Based Reasoning. Proceeding of the European Workshop on Case-Based Reasoning, EWCBR'98. Dublin, Ireland, pp. 172-183.
- Laublet P., Reynaud C. et Charlet J., 2002. Sur quelques aspects du Web sémantique. Actes des deuxièmes assises du GdR 13. Université de Paris-Sorbonne-CNRS (LaLICC), Université Paris-Sud – CNRS (L.R.I.) & INRIA (Futurs),

- Leake B., A. Kinley et D. Wilson, 1997. Case-based similarity assessment: Estimating adaptability from experience. Fourteenth National Conference on Artificial Intelligence. Menlo Park, California, pp. 674-679.
- Lopez V., Motta E. et Uren V., 2006. Next Generation Semantic Web Applications. First Asian Semantic Web Conference (ASWC), Beijing, China, pp. 24-29.
- Maedche A., Motik B., Silva N. et Volz R., 2002. MAFRA - A Mapping Framework for Distributed Ontologies. In Proc. of 13th European Conference on Knowledge Engineering and Knowledge Management (EKAW), Siquenca, Spain, pp. 60-68.
- Maedche A., 2002. *Ontology Learning for the Semantic Web*. Boston: Kluwer Academic Publishers.
- McGuinness D., Fikes R., Rice J. et Wilder S., 2000. The chimaera ontology environment. <http://ksl.stanford.edu/people/dlm/papers/dmcguinness-aaai00-camera-ready-withcitation.doc>.
- Mellal N., 2007. 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.
- Miller G., 1990. Nouns in WordNet: A Lexical Inheritance System, *International Journal of Lexicography* . 4: 245-264.
- Minsky M., 1975. *A Framework for Representing Knowledge. The Psychology of Computer vision*, New York, USA, pp. 211-277,
- Mizoguchi R. et Ikeda M., 1996. *Towards Ontological Engineering*. Technical Report AI-TR-96-1, I.S.I.R., Osaka University, Japan.
- Mizoguchi R., 1998. *A Step Towards Ontological Engineering*. www.ei.sanken.osaka-u.ac.jp/english/step-onteng.html
- Mizoguchi R., Kozaki K., Sano T. et Kitamura Y. 2000. Construction and Deployment of a Plant Ontology. The 12th International Conference, EKAW2000, Juan-les-Pins, France, pp. 113-128.

- Napoli A., 1997. Une introduction aux logiques de descriptions.
<http://www.inria.fr/rrrt/rr-3314.html>
- Neeches R., Fikes R. E., Finin T., Gruber T. R., Senator T. et Swartout W. R., 1991. Enabling technology for knowledge sharing. *AI Magazine*. 12: 36-56.
- Noy N. et Musen M., 2000. PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. *Seventeenth National Conference on Artificial Intelligence*. Austin, Texas, pp. 450–455.
- Noy N. et Musen M., 2001. Anchor-prompt: Using non-local context for semantic matching. *Workshop on Ontologies and Information Sharing at the Seventeenth International Joint Conference on Artificial Intelligence*, Seattle, WA. pp. 63-70.
- Noy N. et Musen M. A., 2002. Evaluating ontology-mapping tools: requirements and experience,
<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.19.3489>.
- Noy N., 2004. *Semantic Integration : A Survey of Ontology-based Approaches*.
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.133.1384>
- Omelayenko B., 2003. *Rdft: A mapping meta-ontology for web service integration, Knowledge Transformation for the Semantic Web*, IOS Press, Amsterdam, pp. 137–153.
- O'Reilly T., 2005. *O'Reilly Network : What Is Web 2.0 : Design Patterns and Business Models for the Next Generation of Software*.
<http://www.oreillynet.com/lpt/a/6228>.
- OWL API version 3.0.0, 28/01/ 2010,
http://sourceforge.net/projects/owlapi/files/OWL_API_28for_OWL_2.0%29/3.0.0/owlapi-3.0.0.zip/download
- Patel-Schneider P. F. et Horrocks I., 2004. Web ontology language (OWL) abstract syntax and semantics. <http://www.w3.org/TR/2004/REC-owl-semantic-20040210/syntax.html>.
- Passant A., 2009. *Technologies du Web Sémantique pour l'Entreprise 2.0*.
www.theses.paris4.sorbonne.fr/these.passant.pdf

- Pinto H., Gómez-Pérez A. et Martins J., 1999. Some Issues on Ontology Integration. Proceedings of the IJCAI-99 workshop on Ontologies and Problem-Solving Methods (KRR5), Stockholm, Sweden, pp. 1-12.
- Psyché V., Mendes O. et Bourdeau J., 2003. http://sticf.univ-lemans.fr/num/vol2003/psyche-06s/sticf_2003_psyche_06s.htm#Heading56
- Psyché V., 2007. Rôle des ontologies en ingénierie des EIAH : Cas d'un système d'assistance au design pédagogique. Thèse de doctorat, Université du Québec, Canada.
- Rahm E. et Bernstein P., 2001. A survey of approaches to automatic schema matching. The VLDB Journal, 10:334–350.
- RDF. Resource Description Framework Model and Syntax Specification. <http://www.w3.org/TR/REC-rdf-syntax/>, 1999.
- RDFS. Resource Description Framework Schema Specification 1.0. <http://www.w3.org/TR/2000/CR-rdf-schema-20000327/>, 2000.
- Saïd I., 2006. INTEGRATION DES SYSTEMES D'INFORMATION INDUSTRIELS : Une approche flexible basée sur les services sémantique, Thèse Pour obtenir le grade de Docteur de l'Ecole Nationale Supérieure des Mines de Saint-Etienne.
- Shvaiko P. et Euzenat J., 2005. A Survey of Schema-based Matching Approaches, Journal on Data Semantics. 4 :146-171
- Sowa J. 1995a. Top-level ontological categories. International Journal of Human and Computer Studies, 43: 669-685.
- Sowa J. 1995. Distinction, combination, and constraints. Proceedings of the IJCAI Workshop on Basic Ontological Issues in Knowledge Sharing, Menlo Park CA, USA: AAAI Press, pp. 626–633.
- Staab S. et Maedche A., 2000. Axioms are objects too: Ontology engineering beyond the modeling of concepts and relations. Research report 399, Institute AIFB, Karlsruhe.

- Stuckensschmidt H., Harmelen F. V., Serafini L., Bouquet P. et Giunchiglia F., 2004. Using C-OWL for the Alignment and Merging of Medical Ontologies, Proceedings of the First Int. WS. on Formal Biomedical K. R. (KRMed), Whistler, Colorado, pp. 8–101.
- Studer R., Benjamins V. R. et Fensel D., 1998. Knowledge engineering: Principles and Methods. *Data Knowledge Engineering*, 25: 161-197.
- Stumme G. et Maedche A., 2001. FCA-MERGE: Bottom-up merging of ontologies. In Proceedings of IJCAI'01, Seattle (WA), pp. 225–234,
- Swartout B., Patil R., Knight K. et Russ T., 1997. Towards Distributed Use of Large Scale Ontologies. *Spring Symposium Series on Ontological Engineering*, Stanford University, CA. pp. 138-138.
- Thanh Le B., Dieng-Kuntz R. et F. Gandon F., 2004. On Ontology Matching Problems (for building a corporate Semantic Web in a multi-communities organization), 6th International Conference on Enterprise Information Systems, Porto, Portugal, 14-17.
- Uschold M. et Grüninger M., 1996. Ontologies: Principles, Methods and Applications. *Journal of Knowledge Engineering Review*. 11: 5-33.
- Van Heijst G., Schreiber A. et Wielinga B. J., 1997. Using Explicit Ontologies in KBS Development. *International Journal of Human and Computer Studies /Knowledge Acquisition*, 46: 183-292.
- Vanwelkenhuysen J. et Mizoguchi R., 1994. Maintaining the workplace context in a knowledge level analysis. *Proceeding of JKAW'94*, Hatoyama, Japan, pp. 33-47.
- Vanwelkenhuysen J. et Mizoguchi R., 1995. Workplace-Adapted Behaviors: Lessons Learned for Knowledge Reuse. *Proceeding of KB&KS '95*. Enschede, The Netherlands, pp. 270-280.
- Web sémantique et ontologies : notes de cours. 2011. http://www.loria.fr/~coulet/Teaching_files/esial_3_il_plbc_ws_poly.pdf
- Wielinga B. et Schreiber A., 1993. Reusable and sharable knowledge bases: A European perspective. *Proceeding of KB & KS'93*, Tokyo, Japan, pp. 103-115.

Zghal S. et S. Ben Yahia, 2007. SODA: Une approche structurelle pour l'alignement d'ontologies OWL-DL.
www.cril.univartois.fr/spip/publications/zghal_etal_JFO07.pdf

Résumé :

La détection des relations de similitude entre les entités sémantiques (concept et rôle) des ontologies est une des principales difficultés rencontrées durant le processus de l'intégration des ontologies. Dans le web sémantique, ces ontologies sont décrites en logiques descriptions. En effet, les entités sémantiques sont organisées selon le mécanisme de la subsumption dans leurs ontologies. La comparaison des concepts pour déduire les relations de similitude entre eux pourra bénéficier des comparaisons de leurs subsumants ou bien de leurs voisins à travers des petites modifications ou bien sans aucune modification. La réutilisation de la solution d'un problème déjà résolu pour résoudre un autre problème similaire est le principe du raisonnement à partir de cas (RÀPC). Dans ce travail nous proposons un système basé sur le RÀPC pour la détection des relations de similitude entre les entités sémantiques des ontologies qui distingue plusieurs phases : la phase de conception, la phase de production et la phase de maintenance.

Mots clés

Ontologies, relations sémantiques, web sémantique, mapping, raisonnement à partir de cas, subsumption, entités sémantiques.

Abstract

The detection of the semantic relations among the semantic entities (concept and role) of ontologies is one of the principal difficulties faced during the process of the ontology integration. In the Semantic Web, these ontologies are expressed as Description Logics (DLs) knowledge bases. Indeed, the semantic entities are organized according to the mechanism of subsumption in their ontologies. This organization implies the existence of these entities in the intentional definitions of their subsumee. Thus, the detection of the semantic relations among semantic entities can be done through the combination of the semantic relations of their subsumers with small modifications. The reuse of the solution of a problem already solved to solve another similar problem is the principle of case-based reasoning (CBR). The objective of this study is to develop a new system based on a CBR to infer the semantic relations by using the techniques of knowledge discovery from databases (KDD) and the semantic distance.

Keywords

Ontologies, semantic relations, Semantic Web, mapping, CBR, subsumption, semantic entities.