

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université des Sciences et de la Technologie Houari Boumediène
Faculté d'Electronique et d'Informatique



Thèse

Présentée pour l'obtention du grade de

Docteur d'état

EN : INFORMATIQUE

Spécialité : Intelligence Artificielle

par :

Karima ASTOUATI Née AKLI

Thème :

**Systemes coopératifs dans un contexte de
multi-expertises**

Soutenue publiquement le 03/07/2012, devant le jury composé de:

M. M. AHMED-NACER	Professeur, USTHB,	Président
Mme A. AISSANI-MOKHTARI	Professeur, USTHB,	Directrice de thèse
Mme Z. ALIMAZIGHI	Professeur, USTHB	Examinatrice
M. Y. DJOUADI	Professeur, U. TIZI-OUZOU	Examineur
M. H. AZZOUNE	Maître de Conférence, USTHB	Examineur
M. O. NOUALI	Maître de Recherche, CERIST	Examineur

A mes enfants Nawel et Ryane

Remerciements

J'adresse mes sincères remerciements à ma directrice de thèse, professeur Aïcha AISSANI-MOKHTARI, pour son encadrement, sa lecture attentive de cette thèse, pour ses conseils avisés et ses encouragements constants.

Que Mr Mohamed AHMED-NACER Professeur à l'USTHB, trouve l'expression de mon respect et mes remerciements pour l'honneur qu'il me fait d'être président de mon jury de thèse.

Mes remerciements vont également à Mme Zaia ALIMAZIGHI Professeur à l'USTHB, Mr Yacine DJOUADI Professeur à l'UMMTO, Mr Hamid AZZOUNE Maître de conférence à l'USTHB ainsi que Mr Omar NOUALI Maître de recherche au CERIST pour m'avoir fait l'honneur d'accepter de juger ce travail en faisant partie du jury de cette thèse.

Ma reconnaissance et mes vifs remerciements vont au Professeur Yamine AIT AMEUR, qui à l'époque était directeur du LISI-ENSMA à Poitier et actuellement est chercheur à l'IRIT, pour m'avoir reçu dans son laboratoire dans une ambiance très cordiale et avoir mis à ma disposition tous les moyens nécessaires pour la réussite de mon stage. En me mettant en contact avec les membres du laboratoire, en l'occurrence le Professeur Ladjel BELLATRECHE, des perspectives fructueuses se sont ouvertes à moi à travers les différentes discussions que j'ai eu avec eux. Que Mr BELLATRACHE trouve ici l'expression de ma reconnaissance et mes remerciements.

Mes remerciements vont à tous ceux qui, par leurs encouragements et leur aide précieuse, m'ont soutenu tout le long de mon travail.

Je tiens enfin à remercier tous mes collègues du département informatique.

Résumé

Ce travail de thèse présente notre contribution dans le domaine des ontologies tant du point de vue de la représentation des connaissances que celui du problème d'hétérogénéité qu'il pose.

Dans la vision représentation des connaissances, nous nous sommes intéressés à une application de grande envergure, « le web sémantique », plus précisément à la qualité des données géographiques qui a imprégné le monde moderne grâce au world wide web (www). Nous avons montré que les concepts utilisés dans ce cas sont imprécis. Il s'avère que les notions de vrai et de faux sont insuffisantes pour rendre compte de la valeur de certaines de ces affirmations. La nécessité de raisonner à partir de tels concepts implique une logique de l'imprécis généralisant les notions de vrai et de faux. A partir de là, nous avons exploité le formalisme des graphes conceptuels, étendu en sa version floue, pour l'utiliser dans les métadonnées du web sémantique géo spatial. Des règles de passage entre ce formalisme et un langage de métadonnées du web sémantique ont été définies.

Pour le problème de l'hétérogénéité des ontologies, nous avons orienté nos travaux vers deux directions :

- L'alignement des ontologies où un système basé sur la technologie agent a été réalisé. Il se base sur les mesures de similarité (syntaxique, structurelle et sémantique) ;
- La négociation par argumentation, permettant à des agents ayant des ontologies différentes du même domaine de dialoguer pour arriver à une réconciliation ontologique.

Mots clés :

ontologie, représentation de connaissances, graphes conceptuels, incertitude, imprécision, web sémantique, appariement, alignement, mesure similarité, argumentation, dialogue, agent.

Sommaire

Introduction générale.....	13
Cadre de la thèse.....	15
Problématiques	17
Apports de la thèse	21
Organisation du document.....	22
Chapitre 1 : De la représentation des connaissances aux ontologies	24
1.1 Introduction	24
1.2 De l'information à la connaissance	26
1.3 Représentation des connaissances	27
1.4 Les niveaux de représentation des connaissances	28
1.4.1 L'ingénierie des connaissances.....	29
1.4.2 Les ontologies	30
1.5 Etude des ontologies	31
1.5.1 Définitions de l'ontologie	32
1.5.1.1 Point de vue philosophique.....	32
1.5.1.2 Point de vue en ingénierie des connaissances.....	32
1.5.2 De la notion de terminologie à celle d'ontologie.....	33
1.5.2.1 Terminologie.....	34
1.5.2.2 Classification.....	35

1.5.2.3	Thésaurus	36
1.5.2.4	Taxinomie	37
1.5.2.5	Ontologie.....	37
1.5.3	Structuration des ontologies.....	38
1.5.3.1	Les constituants de base.....	38
1.5.3.2	Les propriétés portant sur les concepts	40
1.5.3.3	Les propriétés portant sur les relations	41
1.5.3.4	Les propriétés liant une relation et des concepts.	42
1.6	Les formalismes de représentation	42
1.6.1	Le modèle des Frames	42
1.6.2	Représentation graphique des connaissances - Les réseaux sémantiques	43
1.6.3	Les logiques de description.....	46
1.7	Conclusion.....	47
Chapitre 2 : Web Sémantique et Qualité des données		49
2.1	Introduction	49
2.2	Le Web Sémantique.....	50
2.2.1	Principe du Web Sémantique.....	51
2.2.2	Architecture du Web sémantique.....	52
2.2.2.1	Niveau «Nommage/Adressage ».....	52
2.2.2.2	Niveau Syntaxique	52
2.2.2.3	Niveau Sémantique	52

2.3	Qualité et Web des données.....	54
2.3.1	Qualité des données	54
2.3.2	Intérêt des métadonnées	55
2.4	Métalangage de définition des métadonnées dans le web sémantique	58
2.4.1	RDF.....	58
2.4.4.1	Le modèle de données élémentaire de RDF	59
2.4.1.2	La syntaxe élémentaire de RDF	61
2.4.1.3	Autres possibilités de RDF	63
2.4.1.4	Les insuffisances de RDF	67
2.4.2	RDF Schéma (RDFS)	68
2.4.2.1	Les classes.....	68
2.4.2.2	Les Propriétés.....	69
2.4.2.3	Les contraintes	70
2.4.2.4	La documentation.....	70
2.5	Conclusion	71
Chapitre 3 : Vers une représentation floue des métadonnées dans le Web Sémantique		73
3.1	Introduction	73
3.2	Le modèle des graphes conceptuels flous.....	74
3.2.1	Théorie des ensembles flous	76
3.2.1.1	Concept de sous ensemble flou.....	76
3.2.1.2	Définition d'un sous ensemble flou	76

3.2.1.3 Logique floue	77
3.2.2 Graphes Conceptuels Flous.....	77
3.3 Passage RDF (S) / graphes conceptuels	79
3.3.1 Correspondances entre RDF(S) et les graphes conceptuels.....	80
3.3.2 Passage direct	81
3.3.2.1 Déclarations RDF.....	81
3.3.2.2 Descriptions RDF emboîtées	82
3.3.2.3 RDF Schéma	83
3.3.3 Passage inverse	89
3.3.3.1 Classification des concepts avec introduction du flou	89
3.3.3.2 Les règles de Passage.....	91
3.4 Application –Géo Spatial Sémantique Web-.....	99
3.4.1 L’information Géo Spatial et la sémantique	99
3.4.2 Qu’est ce que le Web Géo Spatial ?.....	100
3.4.3 Qualité de l’information géographique	101
3.4.3.1 Imperfection des données géo spatiales	102
3.4.3.2 Définition de la qualité des données	103
3.4.3.3 Composantes de la qualité	103
3.5 Conclusion.....	105
Chapitre 4 : Appariement des ontologies	106
4.1 Introduction	106

4.2 L'hétérogénéité des ontologies	107
4.2.1 Niveau des langages.....	107
4.2.2 Niveau des ontologies.....	108
4.2.2.1 Conceptualisation.....	109
4.2.2.2 Explication	110
4.3 Appariement, alignement et similarité.....	111
4.3.1 Exemple illustratif.....	111
4.3.2 Définitions des concepts de base	112
4.3.2.1 Appariement, correspondance et alignement.....	113
4.3.2.2 Similarité, dissimilarité, distance et autres mesures	115
4.4 Les fondements de l'appariement des ontologies.....	118
4.4.1 La complexité sémantique des ontologies	118
4.4.2 Techniques d'appariement d'ontologies	122
4.4.2.1 Etude de classifications existantes	122
4.4.2.2 Principe de la classification proposée	126
4.4.3 Processus d'appariement.....	130
4.5 Applications de l'Alignement des ontologies.....	132
4.5.1 Recherche d'information sémantique	132
4.5.2 L'intégration de données/Schémas	133
4.5.3 Dialogue Homme/Machine	134
4.5.4 Evolution des ontologies/versionning.....	134

4.6 Conclusion	134
Chapitre 5 : Agents coopératifs pour l'Alignement d'Ontologies	136
5.1 Introduction	136
5.2 La théorie de l'Argumentation	139
5.2.1 Eléments de base d'un système d'argumentation	140
5.2.1.1 Le langage	140
5.2.1.2 L'argument.....	140
5.2.1.3 La relation de contrariété entre les arguments	140
5.2.1.4 Les conditions d'acceptabilité.....	140
5.2.2 Préférences entre arguments	142
5.3 Système multi-agents à trois niveaux (S3A)	143
5.3.1 Schéma général du système	143
5.3.2 Processus d'appariement.....	145
5.3.2.1 Pré-traitement.....	145
5.3.2.2 Calcul des valeurs de similarité et Construction d'argument	146
5.3.2.3 L'alignement	150
5.3.2.4 Le filtrage.....	151
5.4 Tests et Evaluation.....	154
5.5 Conclusion.....	156
Chapitre 6 : Dialogue argumentatif appliqué au E-recrutement	158
6.1 Introduction	158

6.2 Communication/ Argumentation Inter-Agents	159
6.2.1 Communication inter-agents	160
6.2.2 Le dialogue argumentatif	160
6.2.2.1 Le système de dialogue	161
6.2.2.2 Les types de dialogue multi-agents	162
6.3 Le e-recrutement.....	163
6.3.1 Définition	163
6.3.2 Processus de recrutement électronique	164
6.3.3 Un système multi-agents pour le e-recrutement, pourquoi ?	165
6.4 Présentation du système.....	165
6.4.1 L'architecture globale du système	166
6.4.2 Recueil et expertise	167
6.4.3 Le système de dialogue argumentatif	171
6.5 Réalisation du système	176
6.6 Conclusion	180
Conclusion Générale et Perspectives	181
Bibliographie	185

Introduction générale

L'intelligence artificielle (IA) a vu son importance s'accroître considérablement par sa capacité à s'attaquer à de nouvelles classes de problèmes, différents de ceux traités par l'informatique classique. Ces problèmes relèvent d'activités humaines variées (perception, prise de décision, planification, diagnostic, interprétation de données, compréhension du langage, conception, etc.) et nécessitent l'exploitation d'une grande quantité de connaissances, pour l'essentiel spécifiques du domaine étudié. En effet, depuis les débuts de l'IA et plus particulièrement depuis le Knowledge Level de A. Newell (Newell, 80), la question des connaissances et de leur rationalité occupait une place fondamentale en informatique. Et avant même, l'hypothèse connue sous le nom d'« Hypothèse du système physique de symboles » ou encore « Hypothèse représentationnelle », formulée par les deux pionniers de l'IA, le sus-cité et Herbert Simon (Newell et Simon, 1976) témoignait que la conception moderne de la pensée comme manipulation de représentations est l'approche dominante en IA. Les deux auteurs stipulent dans leur article que: «L'étude de la logique et des ordinateurs nous a révélé que l'intelligence réside dans des systèmes physiques de symboles. [...] Un système de symboles est une collection de structures (patterns) et de processus, ceux-ci pouvant produire, détruire et modifier celles-là. [...] La propriété la plus importante de ces structures est qu'elles peuvent designer des objets, des processus et d'autres structures et que, lorsqu'elles désignent des processus, elles peuvent être interprétées». Cette hypothèse postule que l'intelligence est dans la structure de l'arrangement des symboles et les opérations sur elle, indépendamment du medium utilisé pour les matérialiser, d'où la

possibilité, en principe, de construire un système intelligent (Pratt, 1995). Ceci justifie largement l'intérêt grandissant de la communauté de l'IA, pour l'acquisition et la représentation des connaissances. En effet, pour diverses activités nécessitant une certaine forme d'« intelligence », il est vite apparu la nécessité de disposer d'informations et de capacités de traitement de ces informations. C'est la possession de ces capacités qu'exprime la possession de l'intelligence. Dans le cadre de cette thèse, nous assimilons la représentation de l'intelligence à celle de la possibilité de représenter et traiter l'information incertaine et imprécise.

Puis, avec l'avènement du web et les avancées technologiques, un intérêt particulier de l'IA, pour une structuration du sens et une gestion automatique et intelligente du contenu du web est apparu. Ceci s'inscrit dans le cadre d'une application concrète de grande envergure, le web sémantique (Berners-Lee, & al., 2001). L'enjeu fondamental, dans ce cas, est d'assigner une sémantique aux ressources utilisées, de façon à permettre à des agents logiciels de les interpréter de manière autonome. Les informations ne seront plus juste stockées, mais aussi comprises (Luong, 2007).

Annoncée « technologie du futur », par son créateur, Tim Berners-Lee, cette nouvelle vision du web va permettre l'exploitation et la compréhension des informations disponibles dans les différentes ressources non seulement par les hommes mais également par les machines, les programmes et les agents informatiques offrant ainsi une ouverture à de nouvelles possibilités d'automatisation dans le Web et d'assurer une meilleure collaboration entre les humains et les machines (Berners-Lee, 2002).

Ainsi, pour rendre possible et réelle cette perspective, la description du contenu des ressources doit être à la fois formelle et signifiante. Pour cela, les *ontologies* ont été utilisées. Ce sont des éléments principaux pour la construction de l'infrastructure du Web sémantique (Laublet, & al., 2002). Cette dernière doit :

- Permettre de localiser, d'identifier et de transformer des ressources de manière robuste et saine tout en renforçant l'esprit d'ouverture de web avec sa diversité d'utilisateurs,
- S'appuyer sur un certain niveau de consensus portant, par exemple, sur les langages de représentation ou sur les ontologies utilisées,
- Contribuer à assurer, le plus automatiquement possible, l'interopérabilité et les transformations entre les différents formalismes et les différentes ontologies,

- Faciliter la mise en œuvre de calculs et de raisonnements complexes tout en offrant des garanties supérieures sur la validité,
- Offrir des mécanismes de protection (droits d'accès, d'utilisation et de reproduction), ainsi que des mécanismes permettant de qualifier les connaissances afin d'augmenter le niveau de confiance des utilisateurs.

Mais restreindre le Web sémantique à cette infrastructure serait trop limitatif. Ce sont les applications développées sur cette infrastructure qui font et feront vivre cette vision et qui seront, d'une certaine manière, la preuve du concept. Bien sûr, de manière duale, le développement des outils, intégrant les standards du Web sémantique, doit permettre de réaliser plus facilement et à moindre coût des applications ou des services développés aujourd'hui de manière ad-hoc.

D'un point de vue architectural, le Web sémantique est une infrastructure s'articulant autour de trois composantes majeures :

- les métadonnées, qui permettent de qualifier les flux informels du Web ,
- les ontologies, destinées à représenter, dans un formalisme opératoire, les connaissances utiles à la méta-description ,
- les systèmes de raisonnement et de calcul, qui combinent des données issues de sources hétérogènes pour produire de nouvelles informations, susceptibles d'être plus précises, plus adéquates et moins redondantes.

Cadre de la thèse

Nous allons montrer comment à partir de la thématique « Systèmes Coopératifs dans un contexte de Multi-Expertises », nous avons orienté notre travail de thèse vers les ontologies et le web sémantique. De cette thématique, deux notions importantes émergent:

- *L'expertise* qui demeure indissociable des *experts*. Ces derniers sont des spécialistes qui sont reconnus en général socialement dans une discipline pour un domaine donné. Ils résolvent des problèmes, donc font du raisonnement, en utilisant leurs compétences et donc leurs connaissances. Lorsque, pour un problème donné, les compétences de plusieurs experts sont mises en commun ou en interaction, on dit qu'on est dans un contexte de *multi-expertises*. Nous rappelons deux conceptions de l'expertise: l'expertise *idéale* et l'expertise *automatisable*. L'expertise idéale, concept générique et virtuellement muni d'une majuscule,

est vue par (Prince, 1991) comme un état privilégié de l'activité cognitive humaine dans laquelle la mise en œuvre des connaissances acquises en vue d'une compétence se fait d'une manière naturelle et indiscutable. L'ensemble des qualités d'une expertise idéale est coûteux, parce qu'il nécessite un apprentissage et une pratique consommateurs de ressources, et éphémère, parce que c'est l'expert humain qui est considéré comme le support réel de l'expertise.

Une expertise, selon (Pitrat, 1990) qui se place d'un point de vue "computationnel", contient les connaissances qui permettent d'obtenir de bonnes performances dans un domaine. Elle comporte non seulement ce que l'on a le droit de faire, mais surtout ce qu'il est bon de faire dans certaines situations. L'expérience montre que, dans beaucoup de domaines, une très grande quantité de connaissances est nécessaire pour atteindre de bonnes performances.

Il existe un grand décalage entre l'état d'expertise idéale et l'expertise de l'expert. L'expertise automatisable est réductrice de l'expertise de l'expert. En effet, on ne sait pas acquérir, représenter et modéliser tout ce qui a trait à cette expertise. Dans les années 70 et le début des années 80, l'acquisition des connaissances était focalisée sur une modélisation psychologique ou empirique des connaissances d'un expert dans le but de les coder dans un système expert (Charlet, 2002). On passait directement du modèle cognitif de l'expert au système de représentation interne de la machine (Aussenac-Gilles, & al., 1992). Par contre, dans sa version actuelle, acquérir des connaissances, c'est plus une concentration sur la modélisation conceptuelle du monde, en tentant le plus souvent à partir d'une formulation linguistique du problème comme des transcriptions d'interviews d'experts, des descriptions techniques, des notices de maintenance, etc. ., d'élaborer une représentation plus ou moins formelle du problème. La connaissance est alors organisée en *modèle des données*, statique, et en *modèle de raisonnement*, dynamique (Aussenac-Gilles, & al., 1992). Si le modèle de raisonnement peut être considéré comme étant propre à l'expert, le modèle des données doit être indépendant de l'expert et faire donc l'objet d'un accord à peu près consensuel dans la communauté experte du domaine concerné. C'est justement ce consensus que recherchent les études sur les **ontologies**.

- *La coopération* qui prend des sens différents selon qu'elle concerne les décisions et les actions ou au contraire les facultés de perception et d'analyse (Royer, 1994) dans un système complexe composé d'agents. Dans cette thèse, nous orientons nos travaux vers la seconde forme de coopération en soutenant que cela concerne d'abord le partage du

vocabulaire et par la même des ontologies. Ceci fait apparaître des besoins d'intégration de connaissances hétérogènes, de négociation et de résolution de conflits sémantiques et/ou de croyance.

En effet, les conflits sont inévitables dans les systèmes multi-agents, où les agents ont besoin de communiquer par l'échange de connaissances. Une approche commune pour faire face à ces conflits est l'utilisation de processus de négociation fondée sur l'argumentation.

Problématiques

L'objectif général de mes recherches est l'étude des ontologies, vision web sémantique, tant du point de vue de la représentation des connaissances que celui de leur hétérogénéité. Pour plus de clarté, nous avons séparé cette problématique en deux parties :

Problématique 1 (Représentation des connaissances)

Les ontologies occupent une place de choix dans le domaine de l'ingénierie des connaissances et ont prouvé leur efficacité pour la représentation des connaissances puisqu'elles s'inscrivent dans la continuité de nombreux travaux sur les frames, les réseaux sémantiques -graphes conceptuels- et les logiques de description. Définies comme une spécification d'une conceptualisation d'un domaine de connaissances (Grüber, 1993), elles visent à décrire des concepts, qui sont appréhendés comme des représentations mentales plus ou moins universelles ou comme des catégories a priori largement partagées dans la droite ligne de la philosophie de la connaissance (Guarino, 1998).

Dans cette thèse, nous nous penchons sur le formalisme de représentation « les graphes conceptuels » (Sowa, 1984) qui appartient à la famille des réseaux sémantiques. La connaissance, dans ce cas, est répartie en deux ensembles distincts : le support, qui contient les connaissances terminologiques et s'apparente de ce fait à l'ontologie du système, et les graphes conceptuels proprement dits qui contiennent la connaissance assertionnelle construite en utilisant la terminologie du support. Sa forme la plus générale inclut la négation et a un pouvoir d'expression équivalent à celui de la logique du premier ordre. Pour des raisons de décidabilité, le fragment le plus utilisé est celui des Graphes Conceptuels Simples, qui correspond au sous-fragment de la logique du premier ordre constitué des formules positives, existentielles et conjonctives.

On peut aussi dire, que c'est un formalisme de représentation pivot entre la logique et le langage naturel. Or, ce dernier est connu par son caractère vague et imprécis du fait qu'il emploie des termes et des concepts qui ne peuvent être représentés en logique bivaluée. En d'autres termes, les valeurs de vérité «Vrai» ou «faux» ne sont pas les seules à pouvoir être utilisées. On retrouve aussi bien des valeurs intermédiaires comme « très vrai », « plus ou moins vrai», etc., que des valeurs imprécises, ayant un caractère subjectif, comme « jeune », « éloigné », « plus fréquenté », « grand » etc.

Afin de représenter et traiter cette information, une extension du modèle des graphes conceptuels a été proposée pour prendre en compte ce caractère vague et flou, connu sous le nom de « Graphe Conceptuel Flou » (Cao, 2001). Ces résultats de représentation de concepts en Intelligence Artificielle sont intéressants pour résoudre un problème actuel sur le web sémantique dont la vision s'appuie sur l'utilisation d'un langage de description commune, permettant d'exprimer à la fois des ontologies, rassemblant un ensemble de définitions de concepts et de relations, et d'exprimer des annotations et/ou des métadonnées, sur les ressources du Web, utilisant le vocabulaire de ces ontologies.

Ce langage permettrait aux agents informatiques de comprendre les diverses métadonnées et annotations et de communiquer entre eux, en effectuant des raisonnements sur les concepts. Le langage de base qui semble aujourd'hui s'imposer est RDF (Resource Description Framework) (Manola et Miller, 2004), proposé par le W3C (World Wide Web Consortium). Il est simple et possède en plus d'une syntaxe du type triplet, ou ressource-attribut-valeur, une syntaxe graphique. De plus, il correspond à peu près aux Graphes Conceptuels Simples. L'idée serait de proposer un moyen de passage, à travers un ensemble de règles, des graphes conceptuels vers RDF, langage du web sémantique. Puis, étendre RDF à sa version floue, en exploitant les correspondances entre les deux modèles. Les motivations derrière une telle proposition sont :

- La réutilisation : il serait intéressant d'exploiter des applications déjà développées à partir des graphes conceptuels vers RDF(S) afin de les réutiliser facilement dans le cadre du web sémantique. En d'autres termes, il existe des éditeurs de graphes conceptuels, sur lesquels on applique des opérations afin de produire de nouvelles connaissances. Il serait alors possible de réutiliser ces résultats dans le web sémantique grâce au passage graphe conceptuel (GC) vers RDF(S).

- La facilité d'expression: Lorsqu'on est habitué à écrire des prédicats ou à dessiner des graphes conceptuels représentant des expressions du langage naturel, on peut facilement s'intégrer dans la communauté du web sémantique sans avoir à apprendre le langage RDF(S) par le biais d'interface automatique réalisant ce passage.

- Le raisonnement : les graphes conceptuels sont munis d'un certain nombre d'opérations permettant la déduction de nouvelles connaissances. Réaliser le passage vers RDF(S) peut permettre de nouvelles recherches visant la réalisation d'un tel processus de déduction dans le web pour une plus grande ouverture.

- L'extension : dans le cadre des recherches réalisées par l'équipe de l'INRIA Sophia Antipolis au sein de son projet pluridisciplinaire ACACIA, qui vise à développer des aides méthodologiques et logicielles (i.e. modèles, méthodes, outils) pour l'acquisition, la modélisation et la capitalisation des connaissances, une extension de RDF(S) pour la représentation de connaissances contextuelles basées sur le modèle des graphes conceptuels (Delteil, & al., 2003) a été proposée. Les déclarations dans cette proposition sont toutes certaines, dans le sens où l'information est connue et précise. Il n'existe pas de moyen pour exprimer des déclarations comme « il est vrai que », « Peut-être que ... », ... ou encore le caractère vague ou imprécis des concepts et des relations. Ajouter ce caractère, c'est-à-dire, se situer au niveau des concepts en ajoutant une valeur supplémentaire reflétant le caractère imprécis ou incertain de l'information est un des objectifs de cette partie.

Problématique 2 (Hétérogénéité des ontologies)

Les ontologies peuvent être employées dans le cadre de spécification de services web, d'annotation de ressources (pages web, livres, musique, photos) ou encore de description de flux de données. Il est vraisemblable que différentes sources utiliseront différentes ontologies. Il est alors nécessaire de trouver les correspondances entre ontologies qui vont permettre de traduire les données d'une ontologie à une autre. De même, comme s'accorde à dire la communauté de l'ingénierie ontologique, il serait bénéfique de pouvoir intégrer des ontologies de façon à partager et réutiliser les connaissances de chacune (Noy et Hafner, 1997). Dès lors, un problème majeur se pose, celui de la discordance ontologique. Des ontologies développées séparément, même si elles se rapportent au même domaine ou à des domaines ayant une intersection non vide, sont souvent *hétérogènes*, à la fois au niveau de leur structure et au niveau de la sémantique des données qu'elles contiennent. La structure

implique le formalisme de représentation de la connaissance, et elle varie en fonction du langage choisi. La sémantique est liée à l'étude du contexte dans lequel les concepts sont représentés, et elle est identifiée par une étude sur le contenu des concepts. En effet, les informations peuvent être extraites différemment à partir d'une même ressource par des organisations ou des personnes différentes. Quelqu'un peut dire qu'un « livre est écrit par un auteur » alors qu'un autre, affirmera qu'un « ouvrage est soit une monographie dont l'écrivain est X, soit une revue contenant des articles ». Ceci engendrera un désaccord au niveau de la conceptualisation. Aussi, la terminologie choisie dépend des objectifs fixés et de la subjectivité de la personne qui modélise l'ontologie. Il est possible que dans un même domaine, les conceptualisations, donc les ontologies, soient construites différemment. Les termes utilisés peuvent être identiques, différents, synonymes, etc. L'un peut employer le terme « voiture » dans une ontologie pour dénoter des objets qui sont des véhicules à quatre roues dans le monde réel, quant à l'autre, il peut utiliser le terme « automobile » pour signifier le même objet en question. La différence peut aussi provenir de la granularité dans la spécialisation des concepts. L'un peut considérer que le concept « personne » se divise en deux sous-concepts « homme » et « femme » tandis que l'autre préfère le diviser en trois sous-concepts différents à savoir : « enfant », « adulte » et « personne âgée ».

Toute la question est donc de pouvoir exploiter les ontologies simultanément, c'est-à-dire de travailler dans plusieurs « repères conceptuels » à la fois (Berthier, 2004). L'objectif étant donc de garantir une utilisation simultanée et cohérente de plusieurs ontologies.

Pour aborder ce problème, nous nous plaçons dans le cadre d'une *coopération* décrite comme un processus d'argumentation entre agents, processus nécessaire lors d'une prise de décision. L'argumentation est relativement un nouveau paradigme dans l'IA où ses techniques et ses résultats trouvent une grande gamme d'applications dans les deux branches théorique et pratique. Les conflits de type incohérences de manipulation, dus au problème d'hétérogénéité des ontologies, peuvent trouver une solution dans le dialogue argumentatif basé sur la persuasion.

Mais avant, afin d'établir les liens sémantiques entre les entités similaires des différentes ontologies, nous nous orientons vers les travaux de recherche relatifs à l'appariement des ontologies qui vise non pas à éliminer l'hétérogénéité dans le web sémantique mais plutôt à cohabiter avec. L'ensemble de ces liens sémantiques est dit « alignement ».

Pour ce faire, plusieurs travaux proposant des techniques pour mesurer la similarité des entités des ontologies sont recensés dans la littérature. Cependant, dans le cadre du Web sémantique, les langages de représentation des ontologies n'ont pas tous la même puissance d'expression. Ils ne permettent pas de représenter le même ensemble d'entités. Par conséquent, le choix des techniques appropriées dépendra du langage de représentation des ontologies utilisé. La question est donc de savoir comment guider le choix d'une technique pour un langage donné ?

D'autre part, la profusion de travaux se rapportant à l'appariement des ontologies a suscité plusieurs points de vue sur l'explicitation du processus permettant sa réalisation. Ces travaux décrivent ce processus d'appariement avec des niveaux de détails différents. Une vue plus globale et plus détaillée est alors à construire.

Apports de la thèse

Cette thèse présente quelques contributions dans le domaine de la représentation des ontologies et du problème de leur hétérogénéité :

- Nous avons pris en compte l'aspect imprécision dans les graphes conceptuels en vue de leur utilisation dans le web sémantique. Notre intérêt s'est porté sur les langages du web RDF et RDFS et les graphes conceptuels en définissant des règles de passage de l'un vers l'autre et inversement (Guebaili, & al., 2006a et b).
- Application de ces règles dans les applications géo spatiales en insistant sur l'importance de la prise en compte de l'imperfection des connaissances et le rôle des métadonnées dans ces systèmes (Akli, & al., 2010 ; Akli, & al., 2011).
- Une classification des techniques d'appariement sous forme d'un modèle en couche, qui s'appuie sur les différentes caractéristiques supportées par les langages du web sémantique a été défini (Oulefki et Akli, 2008). Ce travail a été réalisé à partir d'une étude critique de certaines classifications proposées dans la littérature.
- Nous avons conçu et réalisé un système d'alignement d'ontologies basé sur la technologie agents. Chaque agent est dédié à un calcul de similarité (syntaxique, sémantique et structurelle). Des tests ont été faits sur des Benchmarks pour la validation du système. Puis, un dialogue argumentatif s'ensuit pour définir la meilleure correspondance (Guebaili, & al., 2011);

– Dans le cadre d'une application de e-recrutement, un dialogue argumentatif a été défini en se basant sur un premier travail publié dans (Akli, 2006 ; Akli, 2005). Il s'agissait d'un outil d'aide à la gestion des conflits basé sur la méthode de négociation.

Organisation du document

Pour aborder les deux aspects cités précédemment, le mémoire inclue six (6) chapitres. Les trois premiers chapitres sont consacrés à la représentation des connaissances. Quand aux trois derniers chapitres, ils sont liés à la coopération en rapport avec le problème de l'hétérogénéité des ontologies. Un aperçu plus détaillé de ces chapitres est présenté dans ce qui suit :

Chapitre 1 : On donne un aperçu sur les aspects fondamentaux de représentation des connaissances et sur les ontologies. Après une clarification de la notion de « connaissances », l'accent est mis sur celle de « représentation des connaissances » et ses impacts sur les travaux de l'IA. Les paradigmes de représentation des connaissances en rapport avec les ontologies, dans un but de formalisation, sont ensuite présentés. Le formalisme des graphes conceptuels et les logiques de description faisant l'objet d'étude par la suite.

Chapitre 2 : On se concentre sur la présentation du web sémantique en donnant un bref aperçu sur ses principes, son architecture et ses langages (RDF, RDF(S), OWL, SWRL). L'importance des ontologies et des métadonnées pour exprimer cette sémantique dans le web sémantique est ensuite exposée. L'accent est mis sur l'imperfection des connaissances dans cette application et le besoin de sa prise en compte dans la description des métadonnées.

Chapitre 3 : Notre proposition qui consiste à introduire du flou dans le web sémantique géo spatial fait l'objet d'une présentation dans ce chapitre. Le modèle des graphes conceptuels, formalisme de représentation de connaissances proche de RDF, RDF(S) modèle de données de base du web sémantique, étendu en sa version floue, est considéré. Avant, les correspondances entre les deux modèles (Graphes conceptuels et RDF) afin de bien cerner les points communs et les différences entre eux sont répertoriées. Puis, des règles de passage des graphes conceptuels flous vers RDF(S) et vice versa, après avoir défini une classification des concepts avec une dimension supplémentaire pour prendre en compte ce caractère flou, sont présentées.

Chapitre 4 : Il porte sur l'appariement des ontologies et des problèmes d'hétérogénéité que son processus doit résoudre. Pour cela, une classification des techniques de base ainsi qu'un

processus d'appariement sont proposés. Le premier pour servir de guide dans le choix de la technique appropriée pour comparer l'ensemble des caractéristiques supportées par le langage ontologique utilisé et le deuxième permettant de prendre en considération le maximum d'ingrédients pouvant composer un système d'appariement.

Chapitre 5 : On présente notre système d'appariement d'ontologies qui est basé sur le calcul de similarités syntaxique, sémantique et structurelle où chaque technique est réalisée à travers un agent. En sortie, trois alignements dédiés à chaque technique sont obtenus. Le meilleur alignement est ensuite déduit grâce à un processus d'argumentation où le choix des correspondances retenues sera justifié.

Chapitre 6 : On s'intéresse à un dialogue argumentatif dans une application de recrutement électronique (e-recrutement). En premier, on mettra l'accent sur le problème de l'hétérogénéité sémantique rendant ainsi la communication inter-agent impossible dans un contexte ouvert, justifiant largement l'utilisation d'un mécanisme dialogique argumentatif. Les notions relatives à ce dernier feront l'objet d'une présentation, ainsi que le domaine d'application, le e-recrutement. On abordera la conception de notre travail par la présentation des ontologies de recrutement, en adoptant UML comme formalisme de modélisation puis le protocole de dialogue argumentatif pour la communication inter-agent en découle. Le travail de réalisation du dialogue argumentatif terminera ce chapitre.

Conclusion Générale : Nous résumons tous les aspects abordés dans ce mémoire en précisant les perspectives ouvertes par notre travail.

Chapitre 1

De la représentation des connaissances aux ontologies

*Il ne faut pas se demander ce que sont des représentations
ou ce qui se passe quand on se représente quelque chose,
mais comment on se sert du mot représentation.*

Ludwig Wittgenstein, (Investigations
Philosophiques, §370)

1.1 Introduction

L'expression « représentation des connaissances » n'est pas, comme on pourrait le croire, associée depuis son avènement à l'intelligence artificielle (IA). Déjà au XVII^{ème} siècle, dans le projet contemporain de construction d'une machine pensante, une nouvelle conception de la pensée comme manipulation de la représentation a été retenue *par la condition d'établir une distance entre le sujet connaissant et l'objet connu* (Pratt, 1995). Aussi, des noms comme ceux du philosophe mathématicien Allemand G.W. Leibniz (XVII^{ème} siècle), des mathématiciens Britanniques C. Babbage (XIX^{ème} siècle) et A. Turing (XX^{ème} siècle) avaient dans des projets paradigmatiques, dans le contexte du paysage intellectuel de l'époque, mis en interaction chacun à sa manière, le concept de *représentation* avec ceux de formalisation et d'automatisation sans les nommer explicitement.

Puis, aux premières années de l'IA, l'importance des connaissances et de leur représentation dans les systèmes intelligents n'a pas été d'emblée reconnue comme le souligne E. Feigenbaum (Luger et Stubblefield, 1993):

« Le premier principe de l'ingénierie des connaissances (IC) est que le pouvoir de résolution de problème démontré par la performance d'un agent intelligent est principalement la conséquence de sa base de connaissances et seulement secondairement de la méthode d'inférence employée. [...] C'est là un résultat important qui n'a pourtant été compris que récemment en IA. Pendant longtemps l'IA s'est focalisée presque exclusivement sur le développement de méthodes d'inférence ingénieuses; presque n'importe quelle méthode fera l'affaire. »

L'idée de Feigenbaum est que l'intelligence est d'abord une affaire de connaissance, plus que de calcul ou de raisonnement. C'est du reste lui qui invente en 1977 le terme Knowledge Engineering (Ingénierie des Connaissances) en donnant naissance dans les années 90 à un nouvel outil conceptuel pour la modélisation des connaissances : *les ontologies*. Ces dernières rendent compte partiellement mais explicitement (Gondan, 2008): (1) des notions mobilisées par la description d'une réalité (2) des règles contraignant la structure de ces descriptions. Les concepts d'intelligence et de connaissances, dans notre cas les ontologies, sont donc très liés : « Quand on parle d'intelligence on est renvoyé à la connaissance : intelligence et connaissance ont été confondues par de nombreux auteurs et le mot intelligence a servi à couvrir l'ensemble des faits relatifs à la connaissance..... » (Oléron, 1991). On peut dire que la puissance réside dans la connaissance.

La problématique de la représentation des connaissances est donc essentielle en IA, du moins dans le courant symbolique (par opposition au courant connexionniste). En effet, même si on sait que toutes les représentations qui ont la même puissance expressive sont en théorie équivalentes, la structure de la représentation joue un rôle déterminant dans les possibilités de manipulation qu'elle offre en pratique.

Ce chapitre donne un aperçu sur les aspects fondamentaux de représentation des connaissances et sur les ontologies. Après une clarification de la notion de « connaissances », l'accent est mis sur celle de « représentation des connaissances » et son impact sur les travaux de l'IA. Les paradigmes de représentation des connaissances en rapport avec les ontologies, dans un but de formalisation, sont ensuite présentés. Il faut toutefois noter que notre ambition n'est pas de proposer une analyse détaillée de tous les usages dont il est fait en ce domaine.

1.2 De l'information à la connaissance

Les concepts de connaissance et d'information sont associés depuis près d'un siècle. En effet, depuis les années 30, plusieurs chercheurs de tous les horizons des sciences (physiques, mathématiques, sociales et humaines) à la recherche de solutions à des problèmes qui semblaient insolubles, ou du moins d'une grande complexité à l'époque, ont eu à user, interpréter ou encore manipuler la connaissance sous l'angle de celle de l'information. Dans ce cas, la valeur des *connaissances* est liée au contexte selon lequel *connaître* est assimilé à un acte défini formellement comme une inférence qu'elle soit *déductive* si l'accent est mis sur la dimension justificative de l'analyse épistémologique ou *inductive* selon qu'on déplace l'analyse épistémologique vers les aspects psycho-cognitifs et individuels, voire personnels de la structuration des connaissances (Robillard, 2006). Cela sous entend que nous attendons des connaissances qu'elles se comportent comme guides des inférences (Kayser, 1997).

La définition de la connaissance est un exercice difficile car c'est une notion abstraite qui implique plusieurs aspects : les actions, les informations, le contexte, l'expérience etc. Pour faire la distinction entre *donnée*, *information* et *connaissance*, nous utilisons une représentation pyramidale (voir figure 1.1) permettant de rendre compte du lien existant entre ces trois notions :

➤ *Les données* se trouvent au premier niveau de la pyramide. Elles sont constituées de bits, d'octets, de caractères, etc. Ce sont des signaux non interprétés. *La donnée* transporte l'information. Elle est en générale mesurable. Exemple : « Une balle mesure 100 grammes ».

➤ *L'information* est une interprétation de *la donnée*. Elle correspond à une donnée conceptualisée. Elles sont statiques et indépendantes du contexte. Exemple: « Il jette une balle de 100 grammes à une hauteur de 10 mètres. ».

➤ *La connaissance* utilise l'information dans le cadre d'actions, dans un but précis. Les actions peuvent être la prise de décisions, la création de nouvelles informations, etc. Elle correspond à l'appropriation et l'interprétation des informations par les hommes. Elles sont inscrites dans l'action et sont étroitement dépendantes de son contexte. Exemple: « loi de Newton, l'accélération terrestre ... ».

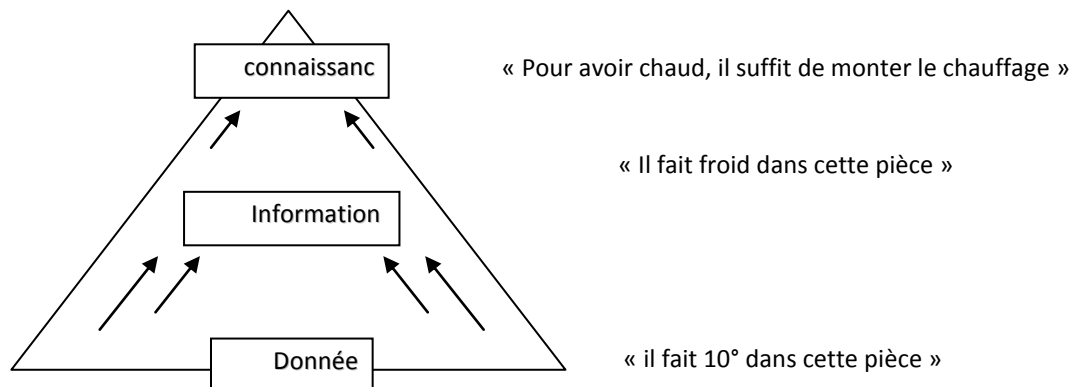


FIG. 1.1 — Pyramide (Donnée, information et connaissance)

Trois (3) caractéristiques de la connaissance, du point de vue épistémologique, ont été retenues (Bachimont 1996, Ganascia 1998, Kayser 1997) et résumées ainsi :

- La connaissance est prise dans un sens technique rendu nécessaire par le fait que dans le domaine de l'intelligence artificielle, nous cherchons *in fine* à construire des artefacts informatiques.
- Il y a connaissances et représentation des connaissances quand les manipulations symboliques effectuées par la machine via des programmes, prennent un sens et une justification pour les utilisateurs interagissant avec ces programmes.
- La pensée repose sur la médiation externe du signe. La technique (ici l'informatique), par ses outils et ses capacités de mémorisation, permet alors à l'homme de se constituer des connaissances qui évoluent et s'accroissent et n'existeraient pas sans cela.

Ainsi, on peut conclure que *l'interprétation humaine* et *l'outil informatique* confèrent respectivement à la connaissance son existence et son support.

« La connaissance est une information qui a été triée, mise en forme, interprétée et transformée. » (Feigenbaum, et McCorduck, 1984).

1.3 Représentation des connaissances

Afin de mieux appréhender la notion de « représentation des connaissances », nous nous sommes intéressées d'une manière chronologique à certains travaux qui font référence en la matière. Il s'agit du « Handbook of Artificial Intelligence » de Barr et Feigenbaum (1982), de

l'ouvrage de Kayser (1997) sur « la représentation des connaissances » et de celui de Sowa (2000) sur le « knowledge representation » :

- Un système de représentation des connaissances est une combinaison de *structures de données* et de *procédures d'interprétation* qui, si elles sont correctement utilisées dans un programme, conduiront à un *comportement cognitif*. (Barr et Feigenbaum, 1982).
- Représenter des connaissances propres à un domaine particulier consiste à décrire et coder les *entités* du domaine sous une forme *exploitable* par un système informatique, pour mener à bien des *raisonnements* et résoudre des problèmes dans le domaine étudié (Kayser, 1997).
- La représentation des connaissances est l'application de *la logique* et *l'ontologie* dans un but de construire des *modèles calculables* pour certains domaines (Sowa, 2000).

La première définition est significative de l'imprécision du concept et de la discipline qui s'y rattache. C'est la raison pour laquelle, nous ne la retenons pas pour la suite de notre travail. Par contre, les définitions de Kayser et Sowa nous semblent plus pertinentes et peuvent s'adapter à notre travail de thèse si on apporte quelques modifications pour mieux appréhender cette notion par la suite.

Définition :

La représentation des connaissances propre à un domaine particulier est l'application de la *logique* et *l'ontologie* pour mener à bien des *raisonnements* c'est à dire trouver les conséquences implicites de la connaissance représentée explicitement. *La logique*, qui façonne l'application, va permettre d'assurer la cohérence de la base de connaissances ; *l'ontologie* assurera une bonne définition des termes et des symboles, leur évitant ainsi d'être confus et/ou sources de confusion.

1.4 Les niveaux de représentation des connaissances

Deux niveaux de représentation des connaissances ont été distingués (McCarthy et Hayes, 1969): le niveau heuristique et le niveau épistémologique. Le premier niveau correspond à l'encodage plus procédural des connaissances. Il introduit les structures de données pour représenter les *objets* et *programmes* qui permettent de simuler les *processus*. Le deuxième niveau est consacré à la représentation déclarative des connaissances en mode logique et il est

réservé à la connaissance qui est liée *aux objets* et aux *processus*. Ce niveau correspond aux techniques de « l'ingénierie des connaissances » qui englobe la conception d'applications d'aide "intelligente" à l'utilisateur, et cherche à définir des principes, des méthodes et des outils pour guider le recueil, l'analyse et la structuration des connaissances et plus particulièrement de l'ontologie. En effet, c'est pour ce niveau épistémologique que McCarthy introduira le premier, selon Sowa, le terme de *ontology* (McCarthy, 1980).

1.4.1 L'ingénierie des connaissances

L'idée d'ajouter des connaissances, c'est à dire du sens, au dessus des données manipulées dans les machines est à la base de l'Ingénierie des Connaissances (IC) qui émerge dans les années 90. Domaine actif des recherches menées en intelligence artificielle, l'IC est une science pluridisciplinaire qui se situe au carrefour de plusieurs réflexions (Charlet, & al., 2000): *la linguistique* pour étudier la formulation linguistique des connaissances, *la terminologie et l'ontologie* pour dégager les concepts, *la psychologie* pour élaborer les méthodes d'élicitation et les modèles d'assistance, *la logique* pour élaborer les modèles formels, *l'informatique* pour mettre en œuvre les modèles (opérationnalisation), etc. Ses domaines d'applications sont nombreux et variés. On peut citer l'acquisition de connaissances à partir de corpus textuels, la recherche d'information dans des bases de données ou sur le web, l'indexation, la réalisation de mémoire d'entreprise, l'annotation documentaire, la maintenance de thésaurus, etc. où la conception et la réalisation de systèmes à base de connaissances (SBC) fait partie des principaux travaux qui s'articulent autour de ses domaines. Ils représentent un modèle de calcul pour certains domaines d'intérêt où les symboles servent de substituts pour les artefacts des domaines du monde comme les objets physiques, les événements, etc. (Sowa, 2000). Un système à base de connaissances maintient la base de connaissances qui stocke les symboles du modèle de calcul sous forme de déclaration sur le domaine et effectue un raisonnement par la manipulation de ces symboles. De tels systèmes nécessitent en particulier une *représentation* adéquate des connaissances mises en jeu ainsi que des mécanismes efficaces d'exploitation de ces connaissances ou de raisonnement. Deux étapes importantes se distinguent : l'acquisition et la représentation des connaissances.

– *La première étape*, longtemps, sans se réduire seulement à cela, était focalisée sur une modélisation psychologique ou empirique des connaissances d'un expert dans le but de les

coder dans un système expert (Charlet, 2002). Par contre, dans sa version actuelle, elle se concentre davantage sur la modélisation conceptuelle du monde, en tentant le plus souvent, à partir d'une formulation linguistique du problème comme des transcriptions d'interviews d'experts, des descriptions techniques, des notices de maintenance, etc., d'élaborer une représentation plus ou moins formelle du problème. On a renoncé alors à passer directement du modèle cognitif de l'expert au système de représentation interne de la machine (Aussenac-Gilles, & al., 1992). Il s'agit de constituer un modèle de connaissance organisé en *modèle des données*, statique, et en *modèle de raisonnement*, dynamique. Si le modèle de raisonnement peut être considéré comme étant propre à l'expert, le modèle des données doit être indépendant de l'expert et faire donc l'objet d'un accord à peu près consensuel dans la communauté experte du domaine concerné. C'est justement ce consensus que recherchent les études sur les ontologies. L'IC utilise un nouveau modèle de représentation des connaissances, nommé ontologie, qui regroupe des hiérarchies de concepts à la Aristote, des relations entre ces concepts et des propriétés logiques qui expriment la sémantique des concepts et des relations.

– *La deuxième étape* est l'application de la logique et l'ontologie dans un but de construire des modèles calculables pour certains domaines (Sowa, 2000). Sowa note que la représentation des connaissances est improductive si la logique et l'ontologie qui façonnent son application dans un certain domaine ne sont pas claires: « sans logique, la représentation des connaissances est vague, et sans aucun critère pour déterminer si les déclarations sont redondantes ou contradictoires et sans ontologie, les termes et les symboles sont mal définis, confus et source de confusion ». Plusieurs paradigmes de représentation des connaissances ont été développés permettant de formaliser les connaissances d'un domaine, puis de mettre en œuvre des raisonnements sur ces représentations. Parmi ces paradigmes, nous pouvons citer : les langages des frames, les réseaux sémantiques, et les logiques de description qui feront l'objet de présentation dans le paragraphe 6.1.

1.4.2 Les ontologies

Dans sa démarche de définition des concepts manifestés par les termes, ces études cherchent précisément à trouver les délimitations acceptables par l'ensemble des experts du domaine. Et au bout du compte, l'ontologie contient les éléments conceptuels à partir desquels peuvent s'élaborer des raisonnements individuels, qui, exprimés par des termes

reconnus du domaine, sont compréhensibles par tout membre de la communauté. Dans un but d'améliorer le partage des connaissances et la réutilisation dans des applications différentes, les ontologies ont donc été proposées comme un mécanisme de spécification (Neches, & al., 1991).

Autour des ontologies, il s'est développé une véritable ingénierie visant à proposer des concepts, des méthodes, des outils et des langages dédiés à la définition, au développement et au déploiement d'ontologies (Fürst, 2002). Il s'agit de *l'ingénierie ontologique* qui a émergé de l'ingénierie des connaissances considérée comme le domaine de prédilection du développement d'expertise en conception de systèmes à base de connaissances. Malgré le fait que l'ingénierie des connaissances ait contribué à accroître cette expertise en l'organisant dans une perspective computationnelle, certains membres de la communauté de l'IA ont éprouvé le besoin de passer à une ingénierie s'appuyant plus solidement sur les fondements théoriques et méthodologiques améliorant ainsi la conception de systèmes intelligents.

Une ontologie fournit un système de concepts fondamentaux c'est-à-dire un système de connaissances qui constitue les connaissances d'arrière plan d'une base de connaissances. Elle offre aussi une conceptualisation du monde cible ainsi qu'une base solide sur laquelle des bases de connaissances partageables sont plus largement utilisables qu'une base de connaissances conventionnelles.

Les ontologies constituent un apport à l'Ingénierie des connaissances dans la mesure où elles visent à développer et à mettre en œuvre des techniques pour la construction de modèles de connaissances conceptuels, pour répondre à une tâche définie.

Les ontologies sont aujourd'hui au cœur de la plupart des systèmes de représentation de connaissances. Elles visent à fournir des représentations à travers lesquelles les machines peuvent représenter explicitement la sémantique des informations. Elles sont utilisées pour permettre aux utilisateurs (humains/systèmes) d'un domaine de modéliser celui-ci, afin qu'il soit compréhensible par les utilisateurs (humains/systèmes) d'un autre domaine. Le modèle ainsi obtenu est utilisable informatiquement, c'est-à-dire qu'il est facile à représenter et à programmer.

1.5 Etude des ontologies

Afin de comprendre le concept d'ontologie, il est intéressant de donner les différentes définitions du mot, présenter ses constituants et montrer comment réfléchir à représenter les connaissances c'est-à-dire décider quels sont les constituants à considérer.

1.5.1 Définitions de l'ontologie

Plusieurs définitions du terme ontologie ont été proposées. Elles dépendent du domaine d'intérêt qu'il soit philosophique, littéraire ou informatique. Nous en reprenons les principales dans cette section.

1.5.1.1 Point de vue philosophique

Historiquement, l'Ontologie (avec "O") est d'abord une notion philosophique qui s'intéresse à la nature et à l'organisation de la réalité. Elle correspond à ce qu'Aristote appelait la Philosophie première, *protè philosopha*, c'est-à-dire *la partie de la métaphysique qui s'intéresse à l'être en tant qu'être*, par opposition aux philosophies secondes qui s'intéressent à l'étude des manifestations de l'être (les *étants*) (Garf, 1996). Il s'agit de capturer l'essence d'un être indépendamment de son existence.

1.5.1.2 Point de vue en ingénierie des connaissances

Dès les années 70, la notion d'ontologie existait, sans être nommée, dans différents systèmes de représentation de connaissances : c'est la TBox des logiques de description, où l'on décrit les types de termes qui existent dans notre représentation et leurs caractéristiques ; c'est le support des graphes conceptuels, où l'on décrit des hiérarchies de multi-héritage entre des types de concepts ou des types de relations ; ce sont enfin les schémas des « Frames » et les classes des langages de représentation par objet.

Puis en 1980, John McCarthy, le même qui en 1956, à la célèbre conférence de Dartmouth, présentait le concept d'« Intelligence Artificielle », associa le mot « ontologie » à l'informatique. Pour cela, il s'est inspiré des écrits du philosophe et logicien Willard Quine (Bénel, 2011). Pour lui, les créateurs de systèmes intelligents basés sur la logique doivent « énumérer tout ce qui existe, construire une ontologie de notre monde » (Smith et Welty, 2001).

Petit à petit, l'usage du terme se répandait dans le domaine de l'intelligence artificielle dès les années 90 avec les travaux de Grüber et son équipe à Stanford (Grüber, 1993). La définition qu'il a proposé «*une ontologie est une spécification explicite d'une conceptualisation* » a fait consensus. Il précise qu'elle est faite dans le contexte du partage et de la réutilisation des connaissances, et que ce qui importe est ce pourquoi (quel usage) une ontologie est faite. Pour mieux la comprendre, une explication plus détaillée est donnée dans (Guarino et Giaretta, 1995). Il explicite la notion de conceptualisation en la définissant comme l'identification par des termes et/ou des symboles, des concepts du domaine et des relations existantes entre ces concepts. Mais cette conceptualisation est forcément basée sur une vision partielle des connaissances d'un domaine. Par exemple, on peut dégager des concepts et des relations permettant de décrire un triangle. Mais cette conceptualisation ne sera peut être plus valable pour décrire un carré. Il convient donc de bien définir le domaine de connaissances que l'on veut conceptualiser.

Depuis, plusieurs définitions de l'ontologie ont été proposées. En 1997, une définition modifiant légèrement celle de Gruber est proposée : «*une ontologie est définie comme étant une spécification explicite et formelle d'une conceptualisation partagée* » (Borst, 1997). Dans cette définition, le terme *conceptualisation* signifie un modèle abstrait d'un phénomène basé sur l'identification de concepts significatifs. Le terme *explicite* signifie que l'ensemble des concepts utilisés et leurs contraintes d'utilisation sont définis d'une façon explicite. L'adjectif *formel* précise que l'ontologie construite doit être lisible par un ordinateur. Enfin, le terme *partagée* montre qu'une ontologie fournit un vocabulaire conceptuel commun et une compréhension partagée par la communauté visée. A partir des travaux précédents, nous proposons la *Définition* :

Une ontologie, dont l'objectif est de faciliter la communication entre acteurs (humains \longleftrightarrow machines), est définie comme une conceptualisation d'un domaine partagée. Les concepts et les relations sont formalisés à l'aide d'un langage compréhensible par une machine et la signification de son vocabulaire de termes doit être définie aussi de manière formelle.

1.5.2 De la notion de terminologie à celle d'ontologie

Lors de l'expression orale ou écrite, le mot constitue un maillon essentiel de la chaîne de compréhension. Un mot mal compris, ou auquel on accorde un sens différent de celui qui lui

est assigné par l'auteur de l'énonciation, pose problème dans le cheminement logique du raisonnement et dans l'échange d'informations. L'idée n'est pas ici d'imposer le sens d'un mot, mais d'indiquer clairement le sens assigné aux *notions*. Afin de lever l'ambiguïté par rapport au vocabulaire utilisé en rapport aux ontologies, nous précisons le sens de certains termes récurrents.

1.5.2.1 Terminologie

Une terminologie présente l'ensemble des termes qui selon le Larousse sont particuliers à une science, un domaine ou un art, puis, étendu à un groupe de personnes ou à un individu selon l'office de la langue française. Une définition plus précise est proposée: « Les terminologies sont des listes de termes d'un domaine ou sujet donné représentant les concepts ou notions les plus fréquemment utilisés ou les plus caractéristiques, cette liste étant ou non structurée » (Lefèvre, 2000).

La terminologie, considérée comme science, s'intéresse au recensement des concepts d'un domaine et des termes qui les désignent pour faciliter l'échange de connaissances dans une langue et d'une langue à l'autre. Pour cela, il faut, d'une part, normaliser et figer l'expression des concepts du domaine en fixant les termes qui les désignent et, d'autre part, rendre compte de l'agencement relatif des concepts recensés (Wuster, 1981).

Ainsi se profilent les trois sommets du triangle aristotélicien (voir figure 1.2) : la chose, c'est-à-dire *l'objet du monde* (par exemple le chat Félix), le signe, c'est-à-dire *le terme*, la chaîne de caractère, ou la photo qui désigne cet objet particulier (par exemple « Félix », le terme qui désigne ce chat parmi tous les chats) et le concept ou *l'objet symbolique* c'est à dire l'idée du chat Félix.

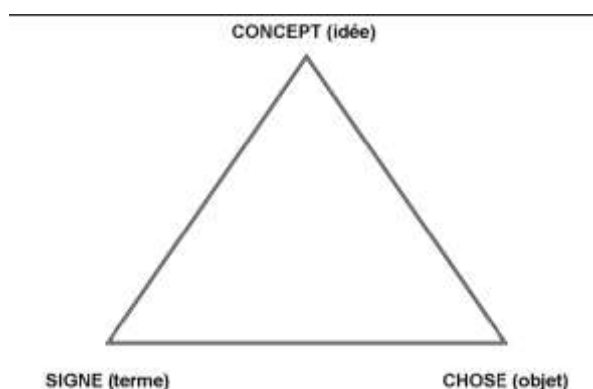


FIG. 1.2 — Triangle aristotélicien simplifié

Les concepts présents dans la terminologie peuvent être reliés par des relations qui témoignent, par exemple, d'un rapport de spécialisation-généralisation (relation « est-un ») comme le montre la figure 1.3.

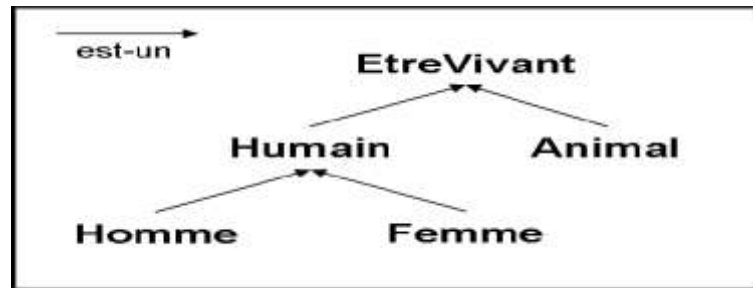


FIG. 1.3 — Relation de spécialisation-généralisation

Une terminologie modélise ainsi un système de concepts sous la forme d'un système de termes normalisés. Le principal intérêt des terminologies est de réduire, voire de supprimer, l'ambiguïté. En effet, puisque par définition, une terminologie de référence spécifie une norme pour un domaine donné, alors le sens de chaque terme est figé et il n'existe qu'une interprétation.

Il existe des terminologies de natures diverses adaptées aux différents objectifs de traitement de l'information : classification pour le recueil de données, nomenclature pour la description d'observations et thésaurus pour la recherche d'information. Cela dit, dans (Lefèvre, 2000), une distinction est faite entre les terminologies et les thésaurus et avance que les thésaurus possèdent une structure de réseaux sémantiques et qu'en cela ils ne peuvent pas être inclus dans les terminologies. Nous précisons les notions de classification, nomenclature et thésaurus dans les sections suivantes.

1.5.2.2 Classification

Une classification est l'action de distribuer par classes et par catégories. Dans (Bourigault, & al., 2004), une définition plus complète mais allant dans le même sens est donnée: « une classification est la répartition systématique en classes, en catégorie d'êtres, de choses ou de notions ayant des caractères communs notamment afin d'en faciliter l'étude. C'est aussi le résultat de cette opération».

Une classification consiste donc à partitionner l'ensemble des objets pour les distribuer en classes et sous-classes constituées d'éléments de plus en plus semblables.

La définition de classes plus spécifiques à l'intérieur de classes plus générales, hiérarchise la classification. Par exemple la classification internationale des maladies et la classification commune des actes médicaux sont de bons exemples de classifications hiérarchiques dans le domaine médical bien qu'elles n'aient pas le même niveau de profondeur.

1.5.2.3 Thésaurus

Un thésaurus est un ensemble structuré de termes d'un vocabulaire, par exemple les termes techniques utilisés et représentés de façon normalisée par des descripteurs ou des mots clés. Les termes sont organisés de manière conceptuelle et reliés entre eux par des relations sémantiques. C'est donc un ensemble organisé de termes, choisi pour leur capacité à faciliter la description d'un domaine et à harmoniser la communication et le traitement de l'information. Chaque terme, appelé descripteur, est aussi peu ambigu que possible et il est préféré à des termes voisins ou synonymes, qualifiés de non-descripteurs, et cela pour faciliter les échanges significatifs (Baneyx, 2007).

En pratique, le thésaurus forme un répertoire alphabétique pour l'analyse du contenu, le classement et donc l'indexation de documents, sachant que dans de nombreux cas, les thésaurus proposent également une définition des termes utilisés. En mode consultation et exploitation des données, le thésaurus devient un instrument de recherche : disposant des vocabulaires et règles de l'indexation, l'utilisateur peut optimiser ses requêtes.

Un thésaurus s'élabore comme un sous-ensemble du vocabulaire usuel et d'au moins un vocabulaire spécialisé. C'est un vocabulaire contrôlé puisqu'il résulte d'un long processus de tri des mots, appellations et expressions utilisées de manière informelle dans un domaine particulier. Il s'agit d'une démarche pragmatique de rationalisation des termes descriptifs.

Pour construire le thésaurus, les termes ainsi identifiés sont inventoriés, comparés, mis en relation et finalement hiérarchisés pour rendre compte des traits essentiels du domaine. Cette hiérarchie s'appuie sur une typologie : chaque terme appartient à une catégorie qui le situe par rapport à tous les autres termes retenus et qui fixe de cette manière sa priorité d'emploi. La hiérarchie des termes peut tout à fait être différente d'un thésaurus à un autre et même, sous réserve d'incohérence, dans un usage ou un autre du même thésaurus. Il demeure toujours une dimension arbitraire dans l'étape de hiérarchisation, soit dans le choix des termes, soit dans leur position hiérarchique bien qu'il existe des normes pour guider l'élaboration des thésaurus.

Finalement, en partant du niveau le plus haut correspondant au domaine du thésaurus, nous trouvons en premier les subdivisions majeures représentant les composantes du domaine - subdivisions souvent nommées microthésaurus , puis pour chaque subdivision, la hiérarchie propre aux descripteurs.

Les thésaurus et les classifications permettent de traduire un message dans un vocabulaire normalisé. Lorsqu'il y a transmission d'information, l'émetteur code le message en fonction d'un langage et du contexte d'énonciation. L'interprétation correcte par le récepteur suppose l'emploi du même langage et la connaissance du contexte. Or le contexte conditionne le codage comme on peut le voir à travers cet exemple : Dans le cas d'un malade hospitalisé pour chimiothérapie d'un cancer ayant développé une aplasie, le dossier sera codé selon l'étiologie cancéreuse par un épidémiologiste mais sous la rubrique aplasie si on s'intéresse à la charge en soins (Baneyx, 2007).

1.5.2.4 Taxinomie

Le mot taxinomie vient du grec. Il est composé de (taxis) et (nomos) qui signifient respectivement « classement », « ordre »pour le premier et « loi » , « règle » pour le deuxième (wikipédia). Il s'agit de la partie de la biologie visant à établir une classification systématique des êtres vivants. Le Petit Robert définit la taxinomie comme étant (1) l'étude théorique des bases, lois, règles, principes d'une classification ; (2) une classification d'éléments. Toutes les classifications se présentent sous la forme d'un arbre (classement arborescent, depuis une racine incluant tous les êtres existants ou ayant existé, jusqu'aux individus. Chaque nœud de l'arbre définit un taxon, qui groupe tous les sous-taxons qu'engendre un nœud.

1.5.2.5 Ontologie

L'ontologie fixe une sémantique aux objets primitifs de la représentation d'un domaine. Pour cela, il convient d'identifier quelles sont les notions élémentaires qui, assemblées et combinées, vont constituer l'ensemble des connaissances représentées pour le domaine en question. L'ontologie est un artefact informatique qui doit être opérationnel. Elle devient, dans ce but, un ensemble de concepts et de relations spécifiques du domaine et atteint un niveau d'abstraction supplémentaire.

1.5.3 Structuration des ontologies

Les ontologies sont basées sur l'utilisation d'un certain nombre de composantes auxquelles des propriétés spécifiques peuvent être associées.

1.5.3.1 Les constituants de base

Les connaissances traduites par une ontologie sont véhiculées à l'aide d'un ensemble de cinq sortes de composants principaux (Fürst, 2002): les concepts, les relations, les fonctions, les axiomes, les règles.

A. Les concepts

Un concept représente un objet ou une notion du domaine. C'est une représentation de l'esprit qui abrège et résume une multiplicité d'objets empiriques ou mentaux par abstraction et généralisation des traits communs identifiables (Mellal, 2007). Il peut être divisé en trois parties: Un terme (ou plusieurs) représentant son nom, des attributs et des instances. Les attributs, également appelés l'intension du concept, correspondent à des caractéristiques, des spécificités particulières, attachées à un concept et permettent de le définir de manière unique dans le domaine (Luong, 2007). Leurs valeurs sont littérales, i.e., de type primitif, comme une chaîne de caractères ou un nombre entier. Par exemple, le concept « Auteur » peut avoir les attributs tels que : avoir un « nom, » une « date de naissance », une « adresse », etc. D'autre part, les instances, aussi appelées extension du concept, représentent les objets manipulés à travers le concept. Par exemple, "Mouloud Feraoun" est l'instance du concept « Auteur ».

Dans (Dieng, & al., 1998), les auteurs affirment l'existence d'une dualité entre l'intension et l'extension : à des intensions incluses $I1 \subset I2$ correspondent des extensions incluses $E1 \supset E2$. Par exemple, la notion de "voiture" ($I2$) inclut la notion de "véhicule" ($I1$) et l'extension $E2 = \{ \text{la voiture d'Ali, la voiture de Souad} \}$ de $I2$ est incluse dans l'extension $E1 = \{ \text{le camion de Mohamed, la voiture d'Ali, la voiture de Souad, le vélo de Rahim} \}$ de $I1$.

Par ailleurs, il est remarqué que deux concepts peuvent partager la même extension sans pour autant avoir la même intension. C'est le cas des concepts d'« étoile du matin » et d'« étoile du soir », qui désignent tous les deux Vénus. De plus, des concepts partageant la même extension mais pas la même intension peuvent être désignés par le même terme. Ceci correspond à des points de vue différents sur un même objet. Par exemple, les chiens peuvent

être considérés comme des animaux de compagnie dans le domaine de la botanique, ou comme des ressources culinaires dans le domaine de la gastronomie chinoise.

Les concepts manipulés dans un domaine de connaissance sont organisés au sein d'un réseau de concepts. L'exemple du concept « Table » montre que cette notion ne peut se définir qu'en utilisant d'autres concepts comme « Meuble », « Plateau » et « Pied ».

B. Les relations

Une relation représente un type d'interaction entre les concepts d'un domaine. Elle est caractérisée par un terme (voir plusieurs) représentant son nom et une signature qui précise son domaine et son co-domaine (Luong, 2007). Le domaine spécifie les concepts susceptibles d'initialiser une relation tandis que le co-domaine sert à contraindre les domaines de ses valeurs. Par exemple, une relation désignée par le terme « Ecrire » peut indiquer une relation entre les concepts « Auteur » et « Livre » dans laquelle « Auteur » est le domaine et « Livre » est le co-domaine.

Lors du développement d'une ontologie, décider entre la représentation d'une notion sous forme d'un concept ou d'une relation peut parfois s'avérer nécessaire. Par exemple, l'écriture d'un texte peut être vue comme un concept et le fait qu'une personne écrive sera exprimé en disant que c'est en relation avec le concept « Ecriture », concept lui-même en relation avec un « Texte ». Le choix dépend alors essentiellement de l'usage des termes dans le domaine, et des liens qu'ils entretiennent avec d'autres concepts ou relations dans l'ontologie.

La relation de subsomption « est-un » (IS-A) a un statut particulier car elle structure la hiérarchie ontologique. Un concept C1 subsume un concept C2 si toute propriété sémantique de C1 est aussi une propriété sémantique de C2. Ainsi, l'extension d'un concept subsumé est forcément plus réduite que celle du concept qui le subsume. Cependant, la relation de subsomption n'est pas la seule relation qui permet de structurer la hiérarchie ontologique. Certains domaines, comme celui de la médecine par exemple, branche anatomie, la relation de méronymie, « *partie-de* » ou « *(part-of)* » est plutôt utilisée. Enfin, il est à noter que tout comme les concepts, les relations sont organisées de manière hiérarchisée à l'aide de la propriété de subsomption.

C. Les fonctions

Une fonction est un type particulier de relations dans lesquelles le nième élément de la relation est défini de manière unique à partir des n-1 éléments précédents (Chamoun, 2006).

Formellement, les fonctions sont définies ainsi : $F : C_1 \times C_2 \dots \times C_{n-1} \rightarrow C_n$. Comme exemple de fonctions binaires, il y a la fonction mère-de ou carré-de, comme fonction ternaire, le prix d'une voiture usagée sur lequel on peut se baser pour calculer le prix d'une voiture d'occasion en fonction de son modèle, de sa date de construction et de son kilométrage.

D. Les axiomes

Les axiomes représentent les intensions des concepts et des relations du domaine et, de manière générale, les connaissances n'ayant pas un caractère strictement terminologique (Staab & Maedche, 2000). Ils sont des expressions qui sont toujours vraies. Leur inclusion dans une ontologie peut avoir plusieurs objectifs (Hernandez, 2006): définir la signification des composants, des restrictions sur la valeur des attributs ou les arguments d'une relation.

Dans le domaine de la géométrie par exemple, l'axiome de Hilbert « Sur une droite, il y a au moins deux points » est une propriété de cardinalité : la relation d'appartenance d'un point à une droite porte une propriété de cardinalité minimum de 2 par rapport à la droite.

E. Les règles

Les règles sont un mécanisme d'inférence de connaissances. Dans le domaine des ontologies, les règles s'appliquent sur des faits qui représentent les concepts et les relations d'un domaine. Une règle est une contrainte explicite sur des comportements, fournissant un support à ces faits et à la conduite des activités. Par exemple, si Leïla est la fille de Naïma et que Nassima est la sœur de Naïma, alors on peut déduire que Leïla est la nièce de Nassima sans avoir besoin de le spécifier explicitement, grâce à la règle générale :

$(X = \text{fille de } Y) \text{ et } (Y = \text{sœur de } Z) \Rightarrow (X = \text{nièce de } Z)$, qui définit, le terme « nièce ».

1.5.3.2 Les propriétés portant sur les concepts

Il est possible d'associer aux concepts un certain nombre de propriétés qui peuvent porter aussi bien sur leur extension que sur leur intension. Nous citons ici d'une manière non exhaustive certaines d'entre elles (Fürst, 2002) :

- *La généralité*. Un concept est générique s'il n'admet pas d'extension. Par exemple, la vérité est un concept générique;

- *L'abstraction*. Un concept est abstrait si toute instance de ce concept est aussi instance d'un de ses concepts fils. Par exemple, dans une hiérarchie comportant les concepts « Homme » et « Femme » qui sont fils du concept « Humain », le concept « Humain » est abstrait;

- *La rigidité*. Un concept est rigide si toute instance de ce concept reste instance dans tous les mondes possibles. Par exemple, « Humain » est un concept rigide, par contre, « Etudiant » 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. Par exemple, « Etudiant » est un concept anti-rigide car l'étudiant est avant tout un humain.

En plus de ces propriétés intrinsèques, nous pouvons aussi citer les propriétés inter-concepts qui portent sur des propriétés extrinsèques aux concepts. Il s'agit principalement de (Guarino et Giarretta, 1995; Fürst, 2002):

- *L'équivalence*. Deux concepts sont équivalents s'ils ont la même extension. Par exemple, « Etoile du matin » et « Etoile du soir » ;

- *La disjonction* ou *l'incompatibilité*. Deux concepts sont disjoints, si leurs extensions sont disjointes. Par exemple, « Homme » et « Femme » sont disjoints dans la mesure où aucune instance de l'un n'est instance de l'autre.

1.5.3.3 Les propriétés portant sur les relations

On retrouve des propriétés intrinsèques comme la *cardinalité* et les *propriétés algébriques* telles que la symétrie, la réflexivité, la transitivité et l'antisymétrie. De plus, des propriétés *inter-relations* peuvent exister comme (Fürst, 2002):

- *L'incompatibilité*. Dans le cas où deux relations ne peuvent pas lier les mêmes instances de concepts. Par exemple, les relations "être rouge" et "être vert" sont incompatibles;

- *L'inverse*. Lorsqu'une relation R_1 lie l'instance I_1 à I_2 , alors qu'une autre relation R_2 lie l'instance I_2 à I_1 . On dit que R_1 et R_2 sont inverses. Par exemple, les relations *fils-de* et *père-de* sont inverses ;

- *L'exclusivité*. Lorsque l'une lie des instances de concepts, que l'autre ne lie pas, et vice-versa. L'exclusivité entraîne l'incompatibilité. Par exemple, l'appartenance et la non appartenance sont exclusifs.

1.5.3.4 Les propriétés liant une relation et des concepts.

Deux propriétés liant une relation et des concepts existent. Elles sont définies comme suit (Kassel, 2002):

- *Le lien relationnel.* Il existe un lien relationnel entre une relation R et deux concepts C_1 et C_2 si, pour tout couple d'instances des concepts C_1 et C_2 , il existe une relation de type R qui lie les deux instances de C_1 et C_2 . Un lien relationnel peut en outre être contraint par une propriété de cardinalité, ou porter directement sur une instance de concept. Par 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 C_1 , et toute relation de type R liant C_1 , les autres concepts liés par la relation sont d'un type imposé. Par 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 », qui est un concept subsumé par « Aliment ».

1.6 Les formalismes de représentation

Les connaissances d'un domaine sont formalisées à l'aide d'un langage permettant d'une part leur exploitation automatique et calculable et d'autre part l'interprétation par des utilisateurs (Bachimont, 2007) : la première veut garantir la calculabilité et l'effectivité des inférences et la seconde l'intelligibilité des représentations. Plusieurs types de langages émergent selon qu'ils se basent ou pas sur les logiques.

1.6.1 Le modèle des Frames

Le concept de schéma a été repris en IA sous la forme de *frames*, vers 1975 par M. Minsky (Minsky 1975). Les *Frames* sont des entités regroupant de façon structurée l'ensemble des connaissances relatives à un objet, un concept, une situation typiques. Un *frame* est composé d'un ensemble d'attributs (*slots*) correspondant aux diverses notions relatives au concept représenté dont les valeurs sont parmi un ensemble de *facettes* (facets). Une autre façon de présenter ces attributs est de les considérer comme des *relations* binaires entre classes dont le premier argument est appelé *domaine* (domain) et le deuxième *portée* (range).

Un ensemble de *frames* concernant un même thème forment un système traduisant la hiérarchie des objets de ce thème et les relations existant entre ces objets. Le mécanisme d'héritage de propriétés permet à un objet du système d'hériter des attributs des objets supérieurs dans la hiérarchie.

1.6.2 Représentation graphique des connaissances - Les réseaux sémantiques

Les réseaux sémantiques ont été développés par (Quillian, 1968) pour représenter la sémantique du langage naturel. Ce formalisme a une représentation de graphe dont les nœuds représentent les entités (concepts et individus) et ils sont connectés par des arcs étiquetés. Il existe des arcs de propriétés et des arcs IS-A. Les premiers affectent des propriétés à des entités et les seconds introduisent les relations hiérarchiques entre les entités. L'expressivité est privilégiée dans ce cas. Des interfaces graphiques sont utilisées permettant ainsi de manipuler des connaissances représentées à l'aide de structures de données *ad hoc*. En général, un manque de *caractérisation sémantique* précise est constaté. Le même réseau peut être interprété de façon différente, selon la sémantique intuitive que chaque utilisateur attache aux étiquettes du réseau. Des problèmes d'inconsistance et d'incomplétude se posent alors.

Pour pallier ces problèmes, l'utilisation d'un formalisme basé sur une notation de graphes pour la logique est préconisée. Il s'agit du modèle des graphes conceptuels (Sowa, 1984) où une association entre expressivité et effectivité de calcul est réalisée.

A l'origine, on parlait de graphes conceptuels pour désigner un système de représentation de la sémantique du langage naturel et on les concevait pour simplifier le passage du et au langage naturel. Ils ont été développés en premier par le philosophe et logicien C.S. Peirce (Peirce, 1882).

Les premiers graphes de Peirce étaient simples. Mais, ils n'étaient pas suffisamment généraux pour représenter toute la logique (Sowa, 1991). En 1883, il a développé sa notation linéaire, qui avec un minimum de modification, a été utilisée dans le calcul des prédicats moderne. En 1897, Peirce a proposé une extension de ces graphes pour représenter toute la logique du premier ordre, et par la suite, la logique modale. Son système, appelé 'graphes existentiels', constitue le fondement logique pour les « graphes conceptuels ».

Le terme ‘graphe conceptuel’ désignait à la fois un modèle de base, défini d’une manière assez précise, et appelé parfois ‘graphes conceptuels simples’, et des extensions, ou des idées d’extension, du modèle de base définies de manière plus ou moins formelles (Munier et Chein, 1996).

Parmi les points positifs ou les avantages derrière l’utilisation des graphes conceptuels, on peut citer :

- Facilité de représentation graphique: « un bon dessin vaut mieux qu’un long discours ». Cette représentation basée sur l’objet, les relations entre les divers objets sont claires et visibles.
- Un modèle de type graphe étiqueté est intéressant d’un point de vue informatique pour diverses raisons. De nombreux algorithmes efficaces ont été développés pour résoudre divers problèmes sur les graphes.
- Un graphe conceptuel est ‘plus facile à lire’ que la formule logique qui lui est associée.
- D’un point de vue algorithmique, les inférences se calculent par des algorithmes de graphes et les algorithmes polynomiaux déjà construits. De plus, la projection, qui est l’opération fondamentale et qui correspond à l’implication logique, est facilement visualisable. Un utilisateur peut donc suivre visuellement un raisonnement.
- Facilité d’extension des capacités descriptives du modèle des graphes conceptuels simples.

Le modèle des graphes conceptuels (GC) simples est décrit dans (Sowa, 1984) et formalisé dans (Chein et Mugnier, 1992). J.F. Sowa propose que sa théorie soit un langage universel de représentation des connaissances pour tout système intelligent. Il se compose de deux (02) niveaux:

- *Le niveau terminologique*, appelé aussi support, qui contient le vocabulaire de base sur lequel vont être construites les connaissances assertionnelles (ou graphes conceptuels). Ces connaissances peuvent être considérées comme un ensemble de concepts liés entre eux par des relations. Le vocabulaire est réparti en trois ensembles : les types de concepts, les types de relations, et les marqueurs individuels instanciés des concepts. Chacun de ces ensembles est muni d’une relation d’ordre partiel, définissant la relation de spécialisation/généralisation.
- *Le niveau assertionnel* qui correspond à la connaissance assertionnelles (les faits) dans une base de connaissances. Ce sont des ‘graphes conceptuels’ qui n’ont de sens que relativement à un support donné et ils sont composés de deux sortes de sommets : des sommets concepts qui représentent les concepts –entités, attributs, états ou événements– et

des sommets relations qui donnent la nature des liens existant entre les concepts. Chaque sommet a une étiquette qui doit respecter certaines règles.

De plus, le modèle est doté de l'opération de projection qui permet de comparer les graphes conceptuels selon la relation de spécialisation. L'opération de projection est un morphisme de graphes autorisant une restriction sur les étiquettes des sommets. Elle permet le calcul effectif de la relation de spécialisation, notée \leq , qui est un pré-ordre partiel sur l'ensemble des graphes conceptuels : $G' \leq G$ si et seulement si il existe une projection de G dans G' . L'opération de projection permet donc de déterminer si un graphe conceptuel est plus spécifique qu'un autre graphe conceptuel. Dans une optique base de données, l'opération de projection peut être utilisée pour trouver l'ensemble des réponses à un graphe conceptuel requête, qui est l'ensemble des graphes conceptuels de la base plus spécifiques que le graphe requête.

Afin de définir un graphe conceptuel, des langages ont été développés comme CGIF (*Conceptual Graph Interchange Form*), OCGL (*Ontology Conceptual Graphs Language*) (Fürst, 2004). Le premier langage permet de représenter concrètement les graphes conceptuels en traduisant chaque graphe dans une représentation logique équivalente. Tandis que le second langage consiste principalement à spécifier d'une part le vocabulaire conceptuel du domaine considéré et d'autre part la sémantique de ce vocabulaire à l'aide d'axiomes. En plus de cet apport (le langage), une facilité d'utilisation a été constatée. Ceci a incité les concepteurs d'applications à les utiliser par exemple dans l'acquisition des connaissances, la recherche d'informations et le raisonnement sur la connaissance conceptuelle (Mellal, 2007).

Outre les travaux théoriques sur le modèle des GC, des plates-formes de gestion des GC ainsi que des outils dédiés à l'acquisition des connaissances ont été développés, comme par exemple COGITo (Guinaldo et Haemmerlé, 1997), CGKEE: Conceptual graph knowledge engineering environment (Lukose, 1997), CGKAT - conceptual Graphs Knowledge Acquisition Tool- (Martin, 1997), CoGITaNT - (Conceptual Graphs Integrated Tools Allowing Nested Typed graphs) (Genest, & al., 1998), ou TooCoM (*Tool to Operationalize an Ontology with the Conceptual Graph Model*) dédié à l'édition et à l'opérationnalisation d'ontologies de domaine (Fürst, 2003) et qui s'appuie sur la plate-forme de manipulation de graphes conceptuels CoGITaNT. On peut aussi citer le moteur de recherche sémantique Corese, développé à l'INRIA Nice Sophia Antipolis, qui est basé sur les graphes conceptuels.

Il implante RDF/S et SPARQL et il est développé dans l'équipe Acacia puis Edelweiss. Plus précisément, il implante la sémantique de RDF et RDFS.

1.6.3 Les logiques de description

Dans ce cas, l'effectivité des calculs est garantie au détriment de l'expressivité. Parmi ces représentations, on trouve les logiques de description (L.D.) (ou logiques terminologiques) qui ne sont qu'une variante de la logique du premier ordre. Chaque logique diffère des autres par l'ensemble de constructeurs utilisés.

Les logiques de descriptions, permettent de représenter les connaissances relatives à un domaine de référence à l'aide de « descriptions » qui peuvent être des concepts, des rôles et des individus (Napoli, 1997). Les concepts modélisent des ensembles d'individus et les rôles des relations entre individus. Un concept correspond à une entité générique d'un domaine d'application et un individu à une entité particulière, instance d'un concept.

Dans ce formalisme, un concept, correspondant à une entité générique d'un domaine d'application, permet de représenter un ensemble d'individus. Ces derniers correspondent à des entités particulières, instances d'un concept. Tandis qu'un rôle représente une relation binaire entre individus. Concepts, rôles et individus obéissent aux principes suivants (Napoli, 1997):

- Un concept et un rôle possèdent une description structurée, élaborée à partir d'un certain nombre de constructeurs. 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.
- Les connaissances sont prises en compte selon plusieurs niveaux : la représentation et la manipulation des concepts et des rôles relèvent du niveau terminologique ; la description et la manipulation des individus relèvent du niveau factuel ou niveau des assertions. Le niveau terminologique est aussi qualifié de TBox et le niveau factuel de ABox.
- La relation de subsomption, Is-a, permet d'organiser concepts et rôles par niveau de généralité : intuitivement, un concept C subsume un concept D si C est plus général que D au sens où l'ensemble d'individus représenté par C contient l'ensemble d'individus représenté par D. Une base de connaissances se compose alors d'une hiérarchie de concepts et d'une (éventuelle) hiérarchie de rôles.

– Les opérations qui sont à la base du raisonnement terminologique sont la classification et l'instanciation. La classification s'applique aux concepts, le cas échéant aux rôles, et permet de déterminer la position d'un concept et d'un rôle dans leurs hiérarchies respectives ; la construction et l'évolution de ces hiérarchies est ainsi assistée par le processus de classification. L'instanciation permet de retrouver les concepts dont un individu est susceptible d'être une instance.

Les différentes logiques peuvent être enrichies d'autres constructeurs : la disjonction de concepts, la cardinalité sur les rôles, la quantification existentielle typée et la conjonction de rôles, la restriction de cardinalité, l'introduction de description et l'introduction d'exclusion, etc. Le formalisme des logiques de description a donné lieu à de nombreuses implémentations et à un certain nombre d'applications dans le domaine du WS pour le développement d'ontologies notamment. Pour plus de détails, une synthèse complète est proposée dans (Napoli, 1997).

1.7 Conclusion

L'axe de recherche « représentation de connaissances » occupe une place prédominante en IA. Nous avons montré à travers ce chapitre, l'importance que revêtent la connaissance et son impact sur l'intelligence que l'on veut simuler sur une machine. Des systèmes experts, aux bases de connaissances, la vision d'acquisition des connaissances a changé. De la modélisation psychologique ou empirique des connaissances d'un expert, on est passé à une modélisation conceptuelle du monde. Ceci, a justifié largement l'utilisation d'un nouveau modèle de représentation des connaissances, l'ontologie. C'est un modèle de données qui est indépendant de l'expert et qui doit faire l'objet de consensus.

Ses domaines d'applications sont nombreux et variés. On peut citer la recherche d'information dans des bases de données ou sur le web, l'indexation, la réalisation de mémoire d'entreprise, l'annotation documentaire, etc.

Dans la suite de notre travail, nous nous intéressons au web qui apparaît comme une large base de ressources à accès distribué. La recherche d'information sur ce web, par des moteurs de recherche, a vite montré ses limites et a encouragé l'utilisation des *métadonnées* dont les termes employés appartiennent à une ontologie. C'est ainsi que s'est succédé de nombreuses propositions pour l'ajout des métadonnées permettant d'ajouter au web la sémantique qui lui

manquait. Il s'agit du « web sémantique » comme l'a nommé Tim Berners-Lee (Berners-Lee, & al., 2000), fondateur du W3C.

La manipulation des ressources du Web par les machines requiert l'expression ou la description de ces ressources. Des langages ont donc été définis par le W3C: RDF (Resource Description Framework) qui permet d'exprimer les données et métadonnées, et qui dispose d'un modèle abstrait, RDF Schéma, de ce qui est décrit grâce à l'expression d'ontologies. Cependant, les termes employés, empruntés à la langue naturelle, sont souvent imprécis et incertains. Le choix d'un formalisme de représentation de connaissances s'impose pour prendre en compte cet aspect d'imperfection de la connaissance. Les graphes conceptuels répondant parfaitement à cette exigence sont considérés dans les chapitres suivants où un intérêt particulier est donné à la représentation de métadonnées dans le web sémantique.

Chapitre 2

Web Sémantique et Qualité des données

We live in an age where everything is on the net but nothing is on the net.

(Australia All Over 2007)

2.1 Introduction

La première génération du Web a permis d'«ouvrir» le réseau internet au commun des mortels en donnant naissance à un outil de navigation simple à utiliser. Il s'est révélé un fantastique outil pour la production, la publication et la diffusion de l'information. Des ressources de types très variés existent se résumant en sources d'information plus ou moins structurées (bases de données, documents XML), sources homogènes du point de vue de leur niveau de structuration mais néanmoins diverses. Par exemple, si l'on considère des textes pas ou peu structurés, il peut s'agir de dépêches, de documents issus d'un traitement de texte, de blogs, de wikis, etc. Le problème de l'hétérogénéité sémantique se pose alors.

La recherche uniquement par mots-clés ne permet de résoudre qu'une partie de l'énorme complexité du problème de la recherche d'information. L'accès, l'extraction, l'interprétation et le maintien de l'information sont laissés à l'utilisateur (Maedche, 2002). En effet, il est très fréquent qu'une recherche ne nous retourne pas uniquement les documents dont nous avons besoin. Ceci, va contraindre l'utilisateur à fouiller le texte en question pour trouver ce qu'il

recherche. Si on considère le terme *tour*, par exemple, les moteurs de recherche ne détectant que des suites de caractères, vont générer une recherche liée à l'outillage, le bâtiment, le voyage ou le cyclisme.

Ces critiques ont été soulevé par le W3C et son président Tim Berners-Lee. Ils ont constaté que le web s'est transformé en un véritable fourre-tout informationnel et présente diverses insuffisances décrites ainsi : absence ou faiblesse d'une véritable description des ressources, la non-exploitation de la sémantique des liens hypertextes par les machines, les limites des outils de recherche, l'impossibilité de lever les ambiguïtés liées aux langues ou aux cultures, et aux contextes, etc. Le Web d'aujourd'hui est essentiellement destiné à être lu par des humains. Il ne peut être manipulé de façon intelligente par des programmes informatiques.

Pour pallier ces insuffisances, une « couche » sémantique est rajoutée dans la représentation du contenu des ressources du Web, ce qui devra les rendre aisément accessibles aussi bien par l'homme que par la machine. De là, est née l'expression « web sémantique » dont l'ambition est d'associer une représentation formalisée au contenu des ressources du web pour structurer leur contenu informatif, créant un environnement dans lequel des agents logiciels sillonnant les pages web pourront effectuer des tâches sophistiquées pour les utilisateurs (Berners-Lee, & al., 2001).

Le Web Sémantique (WS) entend donc proposer des solutions pour accroître la précision et le rappel des moteurs de recherche, à travers l'annotation du contenu des ressources Web. Cette annotation est mise en œuvre à l'aide de concepts et de relations ontologiques, compréhensibles et exploitables à la fois par l'homme et la machine. En associant une couche descriptive aux pages Web classiques, le WS rend possible l'évolution des données vers des connaissances et marque le début d'une nouvelle étape de développement de l'Internet.

Dans ce qui suit, notre intérêt s'est porté sur RDF (Ressources Description Framework) (Manola et Miller, 2004), métalangage de métadonnées du WS. Après un aperçu sur le WS en précisant son principe et son architecture, nous relevons l'importance de prise en compte de l'aspect imperfection de la connaissance de type incertitude et imprécision.

2.2 Le Web Sémantique

Comme précisé précédemment, Tim Berners-Lee (Berners-Lee, & al., 2001) est à l'origine du WS qui fait référence à la vision du Web de demain comme une extension du Web actuel.

Des fonctions avancées, pour la collaboration Homme/Machine en vue de partage et de raisonnement sur le contenu de grands volumes d'informations, doivent être supportées par cette proposition. Concrètement, le WS est une infrastructure permettant l'utilisation de connaissances formalisées en plus du contenu informel du Web actuel, même si aucun consensus n'existe sur jusqu'où cette formalisation doit aller.

2.2.1 Principe du Web Sémantique

Attacher des annotations à toutes les ressources disponibles (documents, pages Web, services...) sur le Web en vue de rendre explicite leur contenu sémantique est le principe du WS. L'interprétation des annotations, donc la sémantique, est précisée entre des agents logiciels, des machines ou voire des humains grâce aux ontologies. La figure 2.1 donne une vision très intéressante de l'utilisation des métadonnées dans le WS.

Des pages web sont annotées à partir de connaissances disponibles dans une ou plusieurs ontologies qui ont pour objectif de normaliser la sémantique des annotations. Ces dernières, regroupées en entrepôts de métadonnées deviennent utiles pour des agents de recherche d'information, faisant appel à des moteurs d'inférence permettant de déduire de nouvelles connaissances formelles des annotations. Ceci montre toute l'importance que revêtent les métadonnées dans le WS, très important pour nous. C'est dans ce cas qu'on se place.

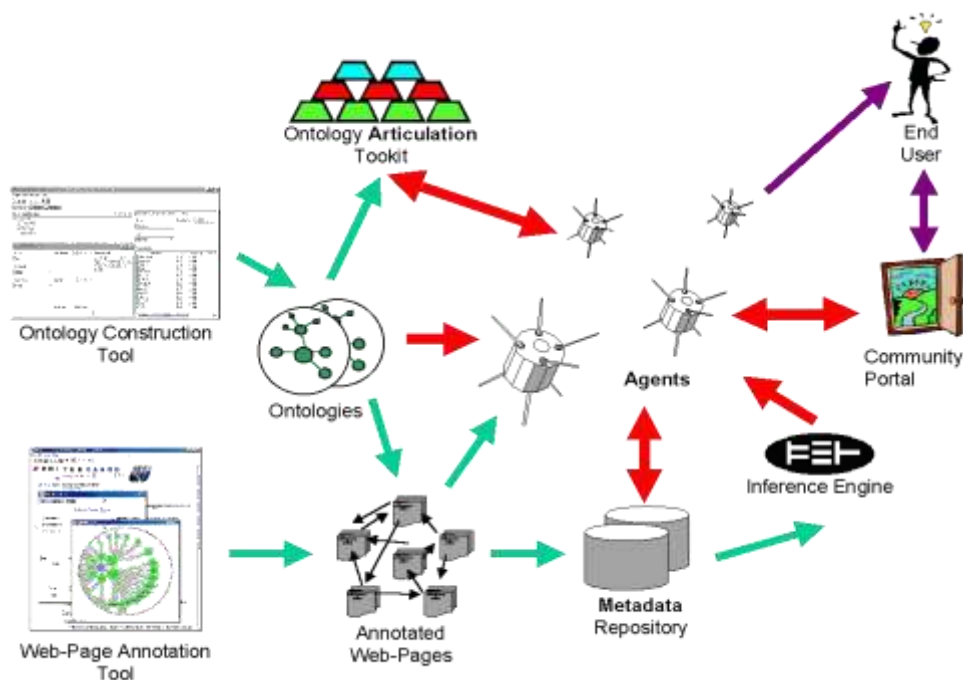


FIG. 2.1 — Vision du Web Sémantique (Hacid et Reynaud, 2003)

2.2.2 Architecture du Web sémantique

La vision courante du WS est souvent représentée dans une architecture comportant trois niveaux (voir figure 2.2), le niveau d'adressage, le niveau syntaxique et le niveau sémantique :

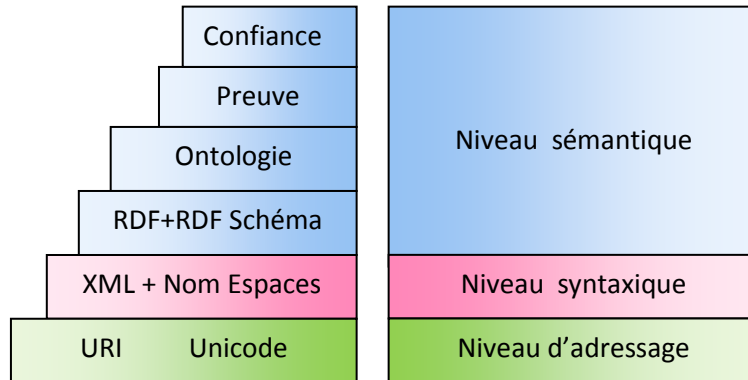


FIG. 2.2 — Architecture du Web Sémantique (Berners-Lee, & al., 2001)

2.2.2.1 Niveau «Nommage/Adressage »

On retrouve deux notions, l'**Unicode** qui est un encodage textuel universel pour échanger des symboles, et l'**URI** (*Uniform Resource Identifier*) qui fournit un adressage standard universel permettant d'identifier de manière unique et non ambiguë les ressources du Web telles que les pages Web, les adresses email, les images, etc. Un exemple d'URI est l'URL (*Uniform Resource Locator*) traditionnelle qui identifie les ressources via une représentation de leur mécanisme d'accès (par exemple: <http://www.usthb.dz>).

2.2.2.2 Niveau Syntaxique

Le niveau syntaxique est établi par l'utilisation de **XML** (eXtensible Markup Language). Ce dernier fournit un ensemble de règles pour la création de vocabulaires qui structurent à la fois les documents et les données sur le Web. Il utilise l'**espace de nommage** (namespace) afin d'identifier les noms des balises (tags) utilisées dans les documents XML. Les schémas XML sont ensuite utilisés comme une méthode autorisant la composition de vocabulaires XML. Bien que XML soit une syntaxe puissante et flexible pour les documents semi structurés, il n'impose aucune contrainte sémantique à la signification de ces documents.

2.2.2.3 Niveau Sémantique

Afin de donner une organisation plus structurée des informations présentes sur le Web à travers une description sémantique des données fournies par XML, **RDF** (*Resource Description Framework*) est utilisé. C'est un standard permettant la mise en place des annotations. Il permet d'affirmer des relations entre les ressources, et de représenter l'information sous forme de triplets < sujet, prédicat, objet >, dont les éléments peuvent être des URIs, des variables ou des littéraux. L'information représentée par RDF est conservée principalement sous forme de déclarations RDF.

RDF Schéma permet ensuite de décrire les hiérarchies de concepts et les relations entre les concepts. Par exemple, dans un document RDF Schéma décrivant un site dédié aux animaux, on peut préciser que le concept souris est un sous-concept du concept mammifère. On peut également créer une relation entre chat et souris pour indiquer que les chats chassent les souris. Dans le contexte qui nous intéresse, RDF et RDF Schéma répondent largement à nos besoins en rapport avec la représentation des métadonnées. Une présentation plus détaillée est donnée par la suite.

Cependant, si on veut manipuler des termes complexes, utiliser des mots qui dénotent le même concept en utilisant des termes différents (*souris* et *mouse* par exemple), ou encore des concepts différents avec le même identifiant (*souris* pour désigner un animal ou un périphérique d'ordinateur), la signification sémantique des données XML représentée par RDF ou RDF Schéma, est insuffisante pour assurer une bonne distinction entre les différents concepts. Ce problème peut être résolu grâce à l'utilisation des ontologies.

La couche **Ontologie** décrit des sources d'information hétérogènes et distribuées en définissant le consensus du domaine commun et partagé par plusieurs personnes et communautés.

La couche **Preuve** a pour but de prouver la pertinence de l'information retournée par les couches de plus bas niveaux. Une des façons de le faire est de garder trace des sources d'information et des raisonnements effectués.

Le Web est un environnement très ouvert et dynamique. De ce fait, toute personne est en mesure d'éditer et de publier des informations de façon très simple. La couche **Confiance**, dans cette architecture, a pour objectif d'évaluer la fiabilité de l'information et des raisonnements. Cette couche repose sur les signatures numériques, le cryptage des données et sur la fiabilité des sources d'information (agents de confiance, certifications, etc.)

2.3 Qualité et Web des données

Le web des données (Linked Data) a comme objectif de publier des descriptions représentées en RDF dont les URIs identifient des documents web, des objets du monde réel et des relations les reliant d'une part et d'interconnecter ces jeux de données d'autre part. Or, ces données, de nature parfois imparfaite, peuvent engendrer des problèmes de qualité des données. Ceci a suscité de l'intérêt de la part des communautés du web et de l'ingénierie des connaissances pour cette problématique.

2.3.1 Qualité des données

Depuis ces dernières années, l'espace de données semi structurées, rendues disponibles sur le Web, est de l'ordre de milliards d'assertions. Elles proviennent de sources différentes dont le nombre est en croissance fulgurante. Dans cet espace de données, des liens sémantiques peuvent être établis entre les données, permettant aux robots d'exploration, aux navigateurs ou aux applications de naviguer parmi les sources de données et d'agréger les informations provenant de sources différentes. Pourtant, dans un environnement ouvert comme le Web, des URIs différentes sont créées régulièrement pour identifier le même objet. Les liens entre URI peuvent être configurés manuellement mais, les données étant nombreuses, certaines approches s'intéressent à la génération automatique de liens entre sources de données RDF. De plus, même si des vocabulaires reconnus existent, permettant de représenter les données sur le Web (FOAF, DublinCore, ...), ces vocabulaires évoluent et sont souvent insuffisants pour certains domaines d'applications qui développent leur propre schéma (ou ontologie). Se pose alors le problème de l'intégration de données liées malgré l'hétérogénéité des vocabulaires utilisés. Ces données liées (ou les liens) peuvent être imprécises et manquantes, périmées, fausses ou soumises à des restrictions d'usage et certaines approches s'intéressent à la provenance des données ou à leur qualité. Dans ce domaine, plusieurs initiatives sont menées. Au niveau français, il y a par exemple le projet DataLift (<http://datalift.org>) qui s'intéresse entre autres à la conversion des données en RDF en rapport avec la ou les ontologies sélectionnées. Au niveau international, le projet LOD2 (<http://lod2.eu>), dont l'activité WS a pris ces trois dernières années de l'ampleur avec la publication généralisée des données structurées comme RDF, aborde des thématiques de recherche liées au WS comme l'amélioration de la cohérence et la qualité des données publiées sur le Web et l'établissement

de la confiance sur le Web de données. On trouve aussi le projet Planet Data (<http://planet-data.eu>) dont l'attention des chercheurs est progressivement passée de l'organisation et la représentation des données à la liaison et la composition de l'énorme quantité de données disponibles sur le Web. Tous ces projets ont pour objectif de résoudre les problèmes engendrés par la masse de données liées disponibles comme la qualité des données qui peut trouver solution dans les travaux sur les métadonnées puisqu'elles constituent actuellement les bases fondamentales du "nouveau" web en matière d'indexation, d'identification, de classification, d'authentification, de conservation, d'intégrité et de gestion des droits, de distribution, de recherche et de récupération des ressources numériques.

2.3.2 Intérêt des métadonnées

Un des grands principes du web sémantique est qu'il est nécessaire d'associer à ses ressources des informations exploitables par des agents logiciels afin de favoriser leur exploitation (Prie et Garlati, 2003). Associer par exemple une notice comprenant des champs : Auteur, Date de création, Date de modification, Mots-clés, à une page web permettrait à un agent logiciel de les utiliser avec leur sens au lieu de considérer cette page comme comprenant un texte seulement. Ces informations sont les métadonnées qui correspondent donc à la description de documents, mais aussi de personnes, de relations entre personnes, de droits sur des œuvres, de données géo-spatiales, d'enquêtes, de chacun des termes d'un thésaurus, etc.

Cependant, les communautés qui conçoivent, créent décrivent des systèmes informatiques dans le contexte du web considèrent la données différemment. Par exemples, dans certains cas la donnée sur laquelle porte la métadonnée est considérée comme ayant le même statut de donnée formalisée, traitable par un système informatique. Dans d'autres, elle est interprétable que par un être humain, et seul la métadonnée en permet le traitement automatique. Dans ce qui suit, une clarification de cette notion, au sens de cette thèse, est donnée.

Une métadonnée n'est rien d'autre qu'un ensemble de données structurées décrivant des ressources physiques ou numériques, assurant l'interopérabilité de l'information et sa gestion. L'ensemble des informations documentant une ressource quelconque est appelé « jeu de métadonnées ». Des ressources de types très variés existent comme des ouvrages, des bases de données, des documents XML, des pages web, des dépêches, des documents issus d'un traitement de texte, des blogs, des wikis, etc.

Une étude extraite de (Dhérent, 2005) montre que les métadonnées sont loin d'être une invention récente. Bien avant l'avènement du web sémantique, des métiers concernés par le management et l'interopérabilité des données comme par exemple les SIG (Système d'Information Géographique) géo spatiale en particulier avaient besoin d'informations détaillées des fichiers produits afin d'être lues par des machines. Ensuite, les métiers de gestion de l'information et du document, notamment patrimonial les ont adoptées mais avec des extensions assez diverses. Les spécialistes des SIG géo spatial parlent dans ce cas de catalogue de métadonnées où on retrouve des informations sur le contenu des données, l'information historique sur les données, comment les données ont été obtenues, et ainsi de suite. Par conséquent, les métadonnées facilitent l'accès aux données pour les applications par la description des détails sur les données.

Les métadonnées peuvent être *externes* ou *internes* aux ressources. Dans le premier cas, elles sont sous forme papier, stockées dans une base de données ou dans un catalogue. Dans ce cas, elles peuvent courir le risque d'être perdues lors de migration d'un format à un autre. Par exemple, lorsqu'on écrit un document Office, qui est une ressource au sens considéré auparavant, on lui associe une métadonnée externe correspondant à « propriétés » (voir figure 2.3). Elle est livrée en même temps que la ressource mais elle ne migre pas en même temps qu'elle. Dans le deuxième cas, elles sont encapsulées dans la ressource, comme par exemple, l'entête d'un document dans une DTD :

```
<ead><eadheader>...</eadheader></ead>
```

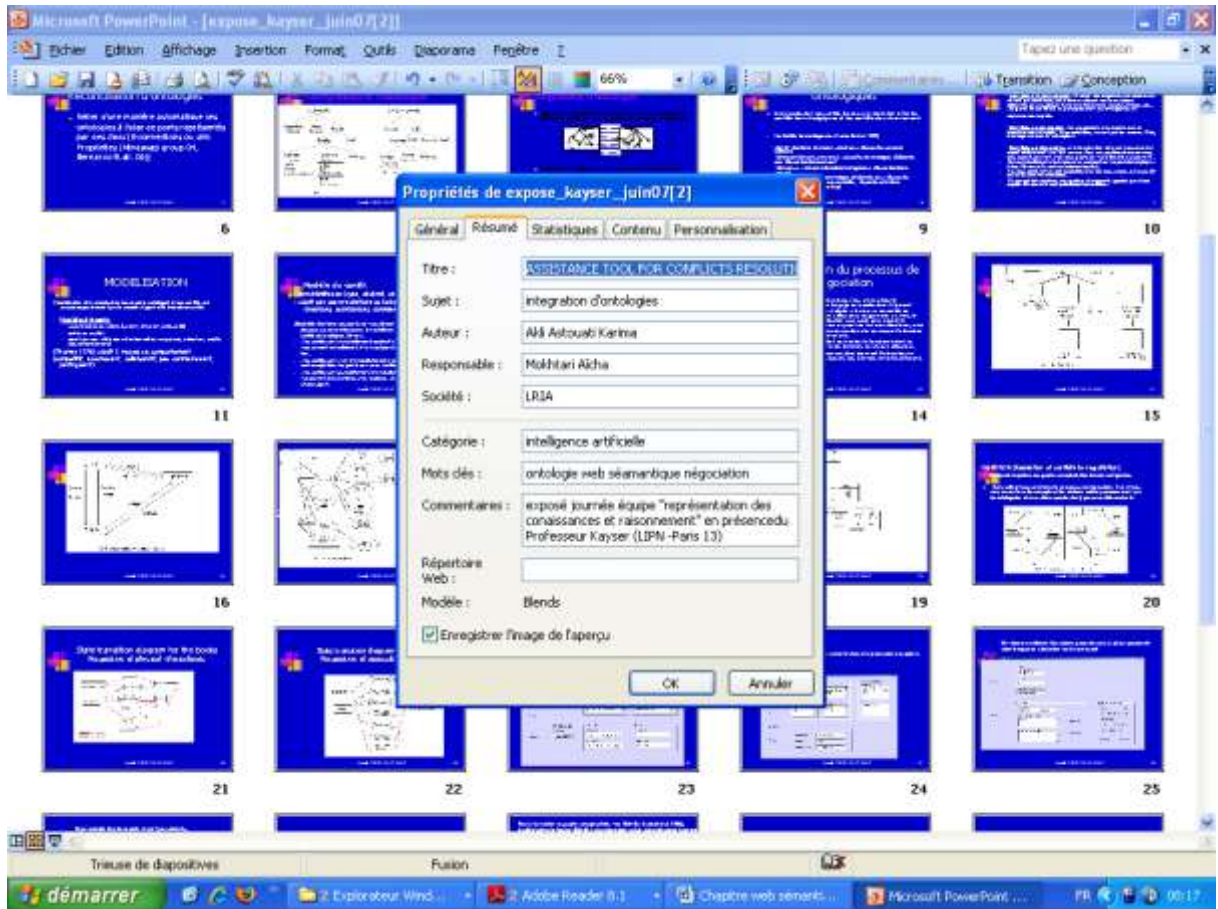


FIG. 2.3 – Métadonnée externe pour un document office

Deux raisons font que les métadonnées telles représentées dans le web traditionnel posent un problème d'interprétation :

- *Des jeux de métadonnées différents*: Divers organismes d'une même activité utilisent parfois des jeux de métadonnées différents qui rendent complexe l'interopérabilité, voire interdisent des actions communes.

- *Absence de description d'un jeu, polysémie sémantique*: Un élément de métadonnées peut avoir des interprétations très différentes selon les cas. Par exemple, « auteur » pour une image pourrait être interprété soit comme l'auteur du tableau représenté par l'image numérique soit comme l'auteur de l'image numérique. Un même vocabulaire dans une même syntaxe ne garantit pas toujours qu'on se comprenne.

Afin de pallier ces problèmes, des solutions sont apportées:

- Les métadonnées doivent être décrites et enregistrées, respectées au maximum les normes et les standards existants et être sous forme d'un modèle conceptuel de données comme par exemple RDF.

- On doit ajouter de la sémantique pour donner plus de sens aux données circulant sur le web. Ceci est réalisé par la définition d'un métalangage permettant de réutiliser les métadonnées structurées provenant de divers jeux. Les termes et les relations d'un sujet précis associés aux métadonnées sont décrits avec précision par des ontologies. Les ressources sont identifiées grâce à un adressage unique l'URI. C'est l'URL (Universal Resource Locator) qui les rend accessibles sur le web.

- On doit tenir compte de l'aspect imperfection des données (incertitude, imprécision,...).
- On doit regrouper des données très diverses en provenance de sources très éparses.
- On doit travailler à l'indexation hiérarchisée de toutes les ressources du web pour une utilisation optimale.

2.4 Métalangage de définition des métadonnées dans le web sémantique

RDF, métalangage de définition de métadonnées, propose d'associer à toute ressource du Web un ensemble de descriptifs qui la caractérisent au mieux. Cependant, il est nécessaire de dériver des langages de définition de corpus homogènes de métadonnées. C'est la fonction d'un métalangage de définition de modèles (RDF-Schéma), souvent appelés "vocabulaires". Comprendre un schéma RDF particulier signifie comprendre la sémantique de chacune des propriétés présentées dans la description de la ressource. Pour plus de précision, RDF et RDFS sont présentés dans ce qui suit :

2.4.1 RDF

RDF est un modèle conceptuel (métalangage) normalisé par le W3C. Il est particulièrement *destiné à la représentation de métadonnées* pour une ressource web, comme le titre, la date de modification de la page Web, droit d'auteur et information sur la licence concernant un document Web. Cependant, en généralisant le concept de 'ressource web', RDF peut être aussi utilisé pour représenter des informations concernant des entités identifiables

sur le web, ou ne peuvent être directement retrouvées à travers le Web, comme les livres d'une bibliothèque, les classes de cours d'une école, etc.

Exemple : les documents incluant des informations concernant des articles disponibles d'un commerce en ligne (spécifications, prix, et disponibilité) ou description des préférences d'un utilisateur de Web.

RDF est plus adapté aux situations où l'information doit être traitée par diverses applications que d'être simplement affichée pour les utilisateurs. Il offre un cadre commun pour exprimer des informations échangeables entre les applications sans perte de sens. La possibilité d'échanger des informations entre les applications veut dire que l'information peut être disponible pour d'autres applications que celle productrice (originaire) de l'information.

RDF est basée sur l'idée d'identifier les choses en utilisant des URIs et en décrivant les ressources en termes de propriétés et de valeurs de propriété. On peut dire aussi que RDF offre les possibilités suivantes (Manola et Miller, 2004) :

- Une interopérabilité entre les applications qui échangent des informations que les machines peuvent comprendre.
- Des facilités pour permettre le traitement automatique des ressources Web.
- Les métadonnées de RDF peuvent être utilisées dans une variété de champs d'application.

Exemple :

- Dans la recherche de ressources, offrir un meilleur mécanisme de recherche dans les catalogues pour décrire les contenus et les relations entre ces derniers.
- Dans un site particulier, page ou bibliothèque numérique par des agents intelligents pour faciliter le partage de connaissance.

2.4.4.1 Le modèle de données élémentaire de RDF

Le modèle de données RDF est dédié à la description des ressources du web. Dans ses principes de base, il utilise trois types d'objets : une « ressource » (resource) décrite par des « propriétés » (properties) ; l'association d'une ressource à une propriété par une valeur de propriété est une « déclaration » (statement).

- *Ressources*. Toute chose décrite par RDF est appelée ressource. Une ressource peut être une page web entière, une partie d'une page web (un élément HTML ou XML spécifique à

l'intérieur du document source), une collection de pages (un site Web complet), un objet qui n'est pas directement accessible via le net (un livre imprimé). Les ressources sont toujours normées par des URIs avec des identifiants optionnels.

- *Propriétés* : Une propriété est un aspect, une caractéristique, un attribut, ou une relation spécifique utilisée pour décrire une ressource. Chaque propriété possède une signification spécifique. Elle définit les valeurs permises, les types de ressources qu'elle peut décrire, ainsi que les relations qu'elle entretient avec les autres propriétés.

- *Valeur* : Peut être soit une simple chaîne de caractères, soit une ressource.

Une '*Déclarations RDF*' est un triplet <ressource, propriété, valeur> où la valeur est soit une ressource, soit un littéral.

Soit l'exemple suivant : Une personne nommé 'Mohamed' a crée une page Web particulière. La façon directe de faire cette déclaration en langage naturel est sous la forme d'une simple déclaration comme : <La page web '<http://www.example.org/index.html>' a 'un créateur' dont la valeur est 'Mohamed'>.

Les parties de cette déclaration sont utilisées dans le but de décrire:

- ✓ La ressource, la page web dans ce cas ;
- ✓ Une propriété spécifique du sujet, créateur ;
- ✓ L'information que la déclaration apporte, c'est la valeur du créateur.

Dans cette déclaration, l'URL '<http://www.example.org/index.html>' de la page Web est utilisée pour son identification. Le terme 'Créateur' est utilisé pour identifier la propriété dont la valeur est identifiée par « Mohamed ».

Des informations additionnelles peuvent être exprimées sur cette page comme sa date de création ou la langue utilisée :

<http://www.example.org/index.html> a une date-création dont la valeur est 16 Août 1999.

<http://www.example.org/index.html> a pour langue l'Arabe.

RDF utilise une terminologie particulière pour préciser des parties variées d'une déclaration <Sujet, Prédicat, Objet> :

- Le sujet de cette déclaration est la page Web <http://www.example.org/index.html>

- La propriété ou la caractéristique du sujet, spécifiée par la déclaration est appelée Prédicat: créateur, date-création, ou le langage.
- La valeur de cette propriété est appelée Objet : John Smith, anglais, 16 Août 1999.

Les déclarations RDF sont schématisées un graphe où la représentation de ces éléments se fait par une forme d'ellipse pour la ressource, un arc pour le prédicat (propriété) et le rectangle pour l'objet. Le graphe correspondant à la déclaration ci-dessus est schématisé par la figure 2.4 :

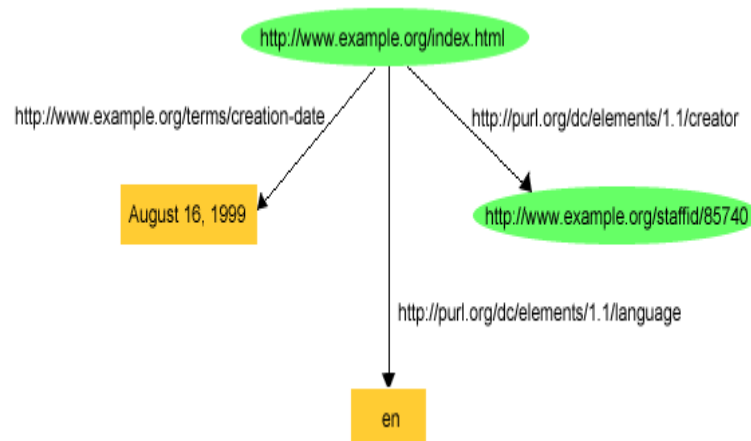


FIG. 2.4 – différentes déclarations concernant la même ressource

Cependant, le langage naturel est suffisant pour la communication entre des humains, au moment où le but est de rendre ces types de déclaration traitable par une machine. Pour cela, il est nécessaire d'avoir :

- ✓ Un système d'identificateurs traitable par les machines pour identifier le sujet, le prédicat ou l'objet dans une déclaration sans aucune possibilité de confusion avec un identificateur qui paraît similaire et qui peut être utilisé par quelqu'un d'autre sur le Web.
- ✓ Un langage traitable par une machine pour représenter ces déclarations et les échanger entre machines.

2.4.1.2 La syntaxe élémentaire de RDF

Pour enregistrer et échanger les graphes, RDF offre une syntaxe basée sur XML appelée RDF/ XML. Il est à noter que RDF/ XML contient des URI, les propriétés et leurs valeurs.

Comme pour HTML, RDF/ XML est traitable par les machines. En utilisant les URI, il peut assembler des pièces d'information à travers le web. Cependant, différent de l'hypertexte

conventionnel, les URIs de RDF peuvent faire référence à n'importe quel objet sur le web ou dans le mode réel qui ne peuvent être directement accessibles à travers le web comme les voitures, les affaires, les gens, etc. Les propriétés RDF elles-mêmes possèdent des URI, pour identifier précisément les rapports qui existent entre les objets (items) liés. La représentation du triplet `<example.org/index.html, example.org/terms/creation-date, "August 16, 1999">` (voir figure 2.5) RDF/XML donne :

```
<? Xml version ="1.0"?>
<rdf: RDF xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
        xmlns:extermns='http://www.example.org/terms/'>
<rdf:Description rdf: about='http://www.example.org/index.html'>
        <extermns :creation-date> August 16, 1999</extermns :creation-date>
</rdf:Description>
</rdf:RDF>
```

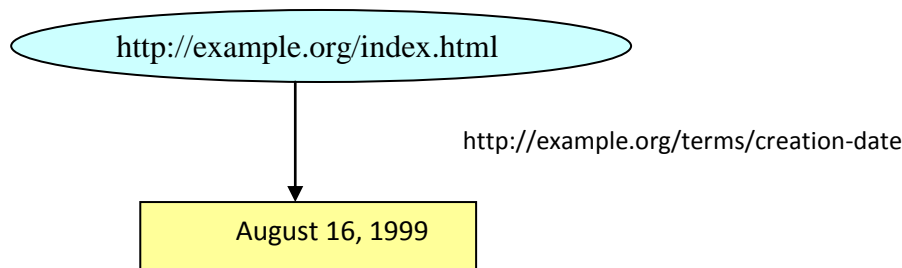


FIG. 2.5 – Exemple d'un schéma d'une déclaration RDF

Chaque élément RDF est un simple emballage. Celui-ci marque les frontières au sein d'un document XML entre lesquelles le contenu est explicitement défini à l'intérieur d'une instance de modèle de donnée RDF.

Dans cet exemple, on peut retrouver une définition d'un espace de nom qui a été déclaré grâce à la notation *xmlns* qui permet de référencer chaque définition de la syntaxe de RDF. Chaque instruction commençant par « RDF » fera référence à cet espace de nom. Le second, appelé *extermns*, permet de rendre disponible par exemple un ensemble de termes.

Concernant `<rdf:Description rdf: about='http://www.example.org/index.html'>`, 'Description' peut être vue simplement comme un lieu pour contenir l'identification de la ressource en cours de description. Typiquement, il y aura une ou plusieurs déclarations faites pour une ressource. 'Description' fournit un moyen de donner une seule fois le nom de la ressource pour plusieurs déclarations.

Lorsque l'attribut 'about' est spécifié, les déclarations dans 'Description' se réfèrent à la ressource dont l'identifiant est déterminé par 'about'.

Dans la portion de code : `<externs:Créateur rdf:resource=""> http://www.w3.org/staffId/85740</tt>>/>`, 'créateur' fait partie du vocabulaire de l'espace de noms défini par 'externs. Ce code signifie que le créateur de la ressource 'http://www.example.org/index.html' a pour identificateur 85740.

2.4.1.3 Autres possibilités de RDF

RDF offre d'autres possibilités comme incorporer des types et des propriétés pour représenter des groupes de ressources et des déclarations RDF. Il permet aussi d'insérer des fragments écrits en XML comme des valeurs de propriétés et les types de données XML. Cela est réalisé grâce aux conteneurs, aux collections RDF et la réification.

A. Les Conteneurs

C'est souvent le besoin de décrire un groupe d'objets, comme par exemple, les auteurs d'un livre ou les étudiants d'un cours ou encore les composants d'un logiciel que la notion d'ensemble a été introduite. RDF offre une variété de types prédéfinis et des propriétés pour décrire de tels groupes.

Un conteneur est une ressource qui contient des objets appelés 'Membres' qui peuvent être des ressources ou des littéraux. On trouve `rdf:Bag`, `rdf:Seq` et `rdf:Alt`.

'**Bag**' représente un groupe de ressources ou littéraux dont les membres peuvent être dupliqués. Il n'y a pas de signification dans l'ordre attribué aux membres.

Exemple :

Soit la déclaration: <Dans le cours de logique, les étudiants sont Ali, Mohamed, Lila et Réda>. Le cours doit être décrit en donnant sa propriété `s:étudiants` (dans un vocabulaire approprié). Sa valeur est un conteneur de type `rdf:bag` (représentant le groupe des étudiants). Dans ce cas, il n'y a aucune signification attribuée à l'ordre. En RDF/ XML, on écrit :

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:s="http://example.org/students/vocab#">
```

```

<rdf:Description rdf:about="http://example.org/cours/logique">
  <s:étudiants> <rdf:Bag>
    <rdf:li rdf:resource="http://example.org/students/Ali"/>
    <rdf:li rdf:resource="http://example.org/students/Mohamed"/>
    <rdf:li rdf:resource="http://example.org/students/Lila"/>
    <rdf:li rdf:resource="http://example.org/students/Réda"/>
  </rdf:Bag>
</s:étudiants>
</rdf:Description>
</rdf:RDF>

```

La valeur de la propriété ‘étudiants’ est formulée comme un ‘Bag’. La forme généralisée ‘rdf:li’ permet de définir les différentes ressources du conteneur. Les conteneurs peuvent avoir des identifiants.

Séquence ou ‘Seq’ représente un groupe de ressources ou littéraux pouvant contenir des duplications. L’ordre attribué aux membres est significatif.

Alternative ou ‘Alt’ représente un groupe de ressources ou littéraux qui sont alternatifs (typiquement pour une valeur unique d’une propriété). Par exemple, une ‘Alt’ peut être utilisée pour décrire des translations alternatives de langages pour le titre d’un livre, ou pour décrire une liste de sites Internet alternative dans lesquels la ressource peut être retrouvée. Une application utilisant une propriété dont la valeur est un conteneur ‘Alt’ peut choisir n’importe quel membre du groupe comme approprié.

Exemple :

Pour illustrer un conteneur Alt, représentons la phrase ‘le code source pour X11 peut être retrouvé dans ‘ftp.exemple.org.ftp1’ ou ‘ftp1.exemple.org’ ou ‘ftp2.example.org’. Elle peut en RDF/XML comme suit:

```

<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:s="http://example.org/packages/vocab#">
  <rdf:Description rdf:about="http://example.org/packages/X11">
    <s:DistributionSite>
      <rdf:Alt>
        <rdf:li rdf:resource="ftp://ftp.exemple.org"/>

```

```

        <rdf:li rdf:resource="ftp://ftp1.example.org"/>
        <rdf:li rdf:resource="ftp://ftp2.example.org"/>
    </rdf:Alt>
</s:DistributionSite>
</rdf:Description>
</rdf:RDF>

```

Il n'existe pas de méthode pour fermer un conteneur, ou pour s'assurer qu'il y a tous les membres du conteneur. Le conteneur exprime seulement qu'un certain nombre de ressources fait partie de l'ensemble, mais aucune façon pour exprimer qu'il n'en n'existe pas d'autres. Donc, quand un graphe peut décrire certains de ces membres, il n'existe pas de façon d'exclure la possibilité qu'un autre graphe décrive des membres additionnels. RDF offre des supports pour décrire des graphes contenant seulement des membres spécifiques. Pour cela on définit les collections.

B. les Collection RDF

Une collection RDF est un groupe d'objets représenté comme une structure de liste dans un graphe RDF. Cette liste est construite en utilisant un vocabulaire de collection prédéfinie consistant en un type 'rdf : List', une propriété 'rdf :First' pour la première ressource et 'rdf :rest' pour le reste de la liste en excluant la tête, ainsi que la ressource 'rdf :Nil' pour fermer le groupe.

Pour illustrer cela, on considère la phrase <les étudiants dans le cours de logique sont uniquement Ali, Mohamed et Lila>. chaque membre de la collection, comme 'S :Ali' est un objet de la propriété 'rdf :First' dont le sujet est une ressource qui représente une liste. Ce membre est lié au reste de la liste par la propriété 'rdf :rest'. La ressource 'rdf :Nil' représente une liste vide. En RDF/ XML, on a :

```

<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:s="http://example.org/etudiant/vocab#">
    <rdf:Description rdf:about="http://example.org/cours/logique">
        <s:etudiant rdf:parseType="Collection">
            <rdf:Description rdf:about="http://example.org/etudiants/Ali"/>
            <rdf:Description rdf:about="http://example.org/etudiants/Mohamed"/>

```

```

    <rdf:Description rdf:about="http://example.org/étudiants/Lila"/>
    </s:students>
  </rdf:Description>
</rdf:RDF>

```

C. La Réification

Les applications ont besoin quelques fois de décrire des déclarations RDF en utilisant d'autres déclarations d'ordre supérieur. RDF offre un vocabulaire incorporé dédié pour le faire. La description d'une déclaration en utilisant ce vocabulaire est appelée 'La Réification d'une déclaration'. Le vocabulaire d'une réification RDF consiste en un type 'rdf:statement', des propriétés 'rdf:subject', 'rdf:predicate' et 'rdf:object', dont les significations sont :

- *Predicate* : La propriété 'predicate' identifie la propriété originale dans la déclaration modélisée. Sa valeur est une ressource qui représente la propriété spécifique dans la déclaration originale.

- *Subject* : La propriété 'subject' identifie la ressource décrite par la déclaration modélisée. La valeur de la propriété 'subject' est la ressource pour laquelle la déclaration originale a été faite.

- *Object* : La propriété 'object' identifie la valeur de la propriété dans la déclaration modélisée. La valeur de la propriété 'object' est l'objet dans la déclaration originale.

- *Type* : La valeur de la propriété type décrit le type de la nouvelle ressource. Toutes les déclarations réifiées sont des exemples de 'RDF:Statement'. Elles ont une propriété type dont l'objet est 'RDF:Statement'. La propriété type est généralement utilisée pour déclarer le type de n'importe quelle ressource.

Exemple :

Soit à déclarer la ressource identifiée par 'exproductions:triple12345'. Son sujet se réfère à la ressource identifiée par 'exproducts:item10245'. Son prédicat est identifié par la ressource 'externs:Weight'. Son objet a pour valeur le littéral typé '2.4' ^{xsd}; decimal. Dans ce qui suit, on déclare cet exemple par son code XML/RDF :

```

<?xml version="1.0"?>
  <!DOCTYPE rdf:RDF [<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">]>
  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:dc="http://purl.org/dc/elements/1.1/"

```

```

xmlns:externs="http://www.example.com/terms/"
xml:base="http://www.example.com/2002/04/products">
<rdf:Description rdf:ID="item10245">
  <externs:weight rdf:datatype="&xsd;decimal">2.4</externs:weight>
</rdf:Description>
<rdf:Statement rdf:about="#triple12345">
  <rdf:subject rdf:resource="http://www.example.com/2002/04/products#item10245"/>
  <rdf:predicate rdf:resource="http://www.example.com/terms/weight"/>
  <rdf:object rdf:datatype="&xsd;decimal">2.4</rdf:object>
  <dc:creator rdf:resource="http://www.example.com/staffid/85740"/>
</rdf:Statement>
</rdf:RDF>

```

2.4.1.4 Les insuffisances de RDF

Le langage RDF permet de définir formellement des métadonnées pour une large catégorie de données. A travers sa syntaxe écrite principalement en XML, les applications peuvent comprendre une grande partie de la traduction des données. Le point faible de ce langage est qu'on ne peut pas définir, des liens de spécialisation ni d'instanciation. C'est ce qui a poussé à son extension connue sous le nom *RDF Schéma* (RDFS).

Exemple: Si on veut formuler en RDF la déclaration suivante : « un livre ou un article d'un magazine possède un titre ou un sujet ».

Il est nécessaire de définir une classe '*ex : Livre*' ou '*ex :MagazineArticle*' et les propriétés '*ex :Sujet*' ou '*ex :Titre*' pour la décrire. Ceci n'est pas permis dans RDF. Aussi, Il n'est pas possible de déclarer qu'un magazine est un ensemble d'articles ou qu'un livre est un ensemble de chapitres. De plus, deux notions désignant le même élément comme, '*Sujet*' et '*Titre*', ne sont pas pris en compte dans RDF .

RDF n'offre aucun moyen pour exprimer de telles définitions. Pour répondre aux besoins des différentes applications, son extension RDFS décrit ces types de classes et de propriétés tout en gardant le même vocabulaire que celui de RDF.

2.4.2 RDF Schéma (RDFS)

RDFS a pour but d'étendre RDF en décrivant plus précisément les ressources utilisées pour étiqueter les graphes. Pour cela, il fournit un mécanisme permettant de spécifier les classes dont les ressources sont des instances et les propriétés. RDFS s'écrit toujours à l'aide de triplets RDF, en définissant la sémantique de nouveaux mots-clés comme par exemple:

- `<ex:Vehicule rdf:type rdfs:Class>` : la ressource 'ex:Vehicule' a pour type 'rdfs:Class'. C'est donc une classe;
- `<snf:TER8153 rdf:type ex:Vehicule>` la ressource 'snf:TER8153' est une instance de la classe 'ex:Vehicule' définie précédemment ;
- `<snf:Train rdfs:SubClassOf ex:Vehicule>` où la classe 'snf:Train' est une sous-classe de 'ex:Vehicule'. Toutes les instances de 'snf:Train' sont donc des instances de 'ex:Vehicule' ;
- `<ex:Localisation rdf:type rdfs:Property>` affirme que 'ex:localisation' est une propriété (une ressource utilisable pour étiqueter les arcs) ;
- `<ex:localisation rdfs:range ex:Ville>` affirme que toute ressource utilisée comme extrémité d'un arc étiqueté par 'ex:localisation' est une instance de la classe 'ex:Ville'.

RDFS ajoute à RDF les possibilités de définir des hiérarchies de classes et de propriétés dont l'applicabilité et le domaine de valeurs peuvent être contraintes à l'aide des attributs 'rdfs:domain' et 'rdfs:range'. A chaque domaine applicatif, on peut associer un schéma identifié par un préfixe particulier et correspondant à une URI. Les ressources instances sont ensuite décrites en utilisant le vocabulaire donné par les classes définies dans ce schéma. Les applications peuvent alors leur donner une interprétation opérationnelle.

2.4.2.1 Les classes

- RDFS définit un certain nombre de ressources comme classes de base de son vocabulaire :
- '**rdf:Resource**' : tout ce qui est décrit par une expression RDFS est appelée ressource. C'est une instance de la classe 'rdf:Resource'.
 - '**rdf:Property**' : c'est un sous ensemble des ressources RDF. On l'utilise pour étiqueter les arcs d'un graphe RDF.
 - '**rdfs:Class**' : l'étape de base dans tout type de description de processus est l'identification des objets à décrire. RDFS se réfère à ces types d'objets par la notion de *classe*.

Une classe RDFS correspond à un *concept générique*, à un *type* ou à une *catégorie*. On peut l'assimiler à la notion de classe dans les langages de programmation orientés objet comme dans Java. Les classes RDF peuvent décrire différents types d'objets comme les pages web, les personnes, les types de documents, les bases de données ainsi que les concepts abstraits.

2.4.2.2 Les Propriétés

En plus des possibilités de décrire des classes spécifiques d'objets, la communauté d'utilisateurs a aussi besoin de décrire des propriétés spécifiques qui caractérisent leurs classes. Ce sont des instances de la classe **'rdf:Property'**. RDFS offre un mécanisme d'expression des relations entre les classes et leurs instances ou super-classe.

- **'rdf:Type'** : associe à une classe un type précis, et indique que cette ressource possède toutes les caractéristiques qu'un membre de cette classe doit avoir. En respectant les règles d'héritage et d'instanciation, une ressource peut être instance de plusieurs supers classe.
- **'rdfs:subClassOf'** : est une propriété transitive restreinte à la classe **'rdfs:Class'**. Une classe peut être une sous-classe de plusieurs classes. Cependant, une classe ne peut jamais être déclarée comme étant une sous-classe d'elle-même.
- **'rdfs:subPropertyOf'** : est une instance de **'rdf:Property'**. Elle est employée pour spécifier qu'une propriété est une spécialisation d'une autre. Une propriété ne peut jamais être instance d'elle-même.
- **'rdfs:seeAlso'** : c'est un moyen de fournir des informations additionnelles sur la ressource sujet. Elle peut être spécialisée en utilisant **'rdfs:subPropertyOf'** afin d'indiquer la nature des informations que l'objet source a sur la ressource sujet.
- **'rdfs:isDefinedBy'** : c'est une sous-propriété de **'rdfs:seeAlso'**. Elle sert à indiquer la source qui a défini la ressource sujet.

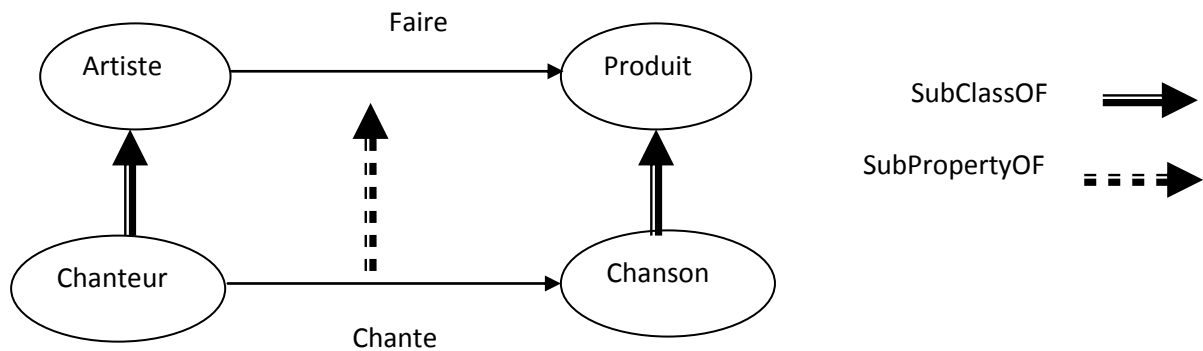


FIG. 2.6 – Schéma utilisant les propriétés SubClassOf et SubPropertyOf.

2.4.2.3 Les contraintes

RDFS fournit un mécanisme de contraintes, qu'on résume en :

- **'rdfs:ConstraintResource'** : elle permet de définir une sous-classe de 'rdf:Resource' dont les instances sont les concepts de RDFS impliqués dans l'expression des contraintes. Le but de cette classe est de fournir un mécanisme de définitions de contraintes associées à un modèle RDF.
- **'rdfs:ConstraintProperty'** : c'est une sous-classe de 'rdf:Property'. Ses instances sont des propriétés utilisées pour spécifier des contraintes. Cette classe est une sous-classe de 'rdfs:constraintResource' et correspond au sous-ensemble de cette classe représentant les propriétés.
- **'rdfs:range'** : c'est une instance de 'ConstraintProperty'. Elle est utilisée pour indiquer à quelles classes les valeurs d'une propriété appartiennent. La valeur d'une propriété 'range' est toujours une classe. Les contraintes 'range' sont seulement appliquées aux propriétés.
- **'rdfs:domain'** : c'est une instance de 'ConstraintProperty' qui est utilisée pour indiquer aux classes sur quels membre une propriété peut être utilisée.

2.4.2.4 La documentation

Les concepts atomiques sont désignés au premier niveau par des commentaires et des labels :

- **'rdfs:comment'** : il est utilisée pour ajouter des commentaires pour décrire une ressource ;
- **'rdfs:label'** : elle permet d'ajouter un nom à une ressource.

Exemple d'utilisation de RDF- RDFS :

On veut définir une classe 'Etudiant', comme sous-classe de la classe 'Personne'. On déclare ensuite une propriété 'Matricule' qui s'applique sur les objets de la classe 'Etudiant'

et prend ses valeurs dans la classe 'Etudiant-Matricule'. Elle est représentée en RDF- RDFS comme suit :

```
<rdfs:Class rdf:ID='Etudiant'>
    <rdfs:subClassOf rdf:resource='#Personne'>
</rdfs:Class>
<rdf:Property rdf:ID='Matricule'>
    <rdfs:domain rdf:resource='#Etudiant'>
    <rdfs:range rdf:resource='#Etudiant-Matricule'>
</rdf:Property>
```

L'instanciation d'un nouvel étudiant se fait :

```
<Etudiant id='1236'>
    <rdfs:label> Matricule de l'étudiant </rdfs:label>
    <rdfs:comment> les matricules des étudiants sont codés séquentiellement
</rdfs:comment>
</Etudiant>
```

2.5 Conclusion

Nous avons montré à travers ce chapitre que la manipulation des ressources du Web par les machines requiert leur expression ou leur description par des métadonnées afin de soutenir des applications dans leur traitement du contenu. A cet effet, des langages, disposant d'un modèle abstrait de ce qui est décrit grâce à l'expression d'ontologies, sont définis. Il s'agit de RDF et RDFS, langages ontologiques de premier niveau, qui répondent bien à l'exigence du traitement du contenu des ressources du web. Actuellement, ils suscitent beaucoup d'intérêt de la part de la communauté du Web. Avec les travaux sur le Linked Data, l'importance de saisir la qualité et la pertinence d'une ressource, qui n'est pas forcément exprimée d'une manière certaine, pour répondre au besoin d'un utilisateur externe, a été mise en évidence. Aussi, les concepts utilisés pour exprimer cette qualité peuvent s'avérer imprécis. Toutefois, en vertu de la sémantique classique, RDF ne peut représenter cette imperfection de l'information.

Pour y remédier, nous nous sommes intéressés dans le chapitre suivant au modèle des graphes conceptuels GCs, étendu en sa version floue, FCGs, pour prendre en compte l'aspect

imperfection des connaissances. Ce choix a été dicté par le fait que sa représentation graphique est proche de celle du modèle de données de RDF. Puis, nous avons mis en évidence des règles de passage entre FGCs et RDF / RDFS pour les appliquer par la suite au web géo spatial.

Chapitre 3

Vers une représentation floue des métadonnées dans le Web Sémantique

*Il lui arrive assez souvent de ne pas achever ses phrases,
ce qui donne à sa pensée une sorte de flou poétique.*

(Les Faux-Monnayeurs (1925) André Gide)

3.1 Introduction

Nous partons du principe que les expressions du langage naturel utilisées peuvent parfois être imparfaites. On retrouve des connaissances floues de nature vague, imprécise et incertaine. Des travaux ont déjà été entrepris afin de prendre en compte ce type d'information en utilisant les graphes conceptuels (Sowa 1984) et les principes de la logique floue (Zadeh, 1965). Ceci a donné naissance aux Graphes Conceptuels Flous (Genest 2000 ; Cao 2001 ; Thomopoulos & al., 2003).

Notre objectif, dans ce cas, est de ramener une telle représentation sur le web sémantique pour ce genre d'information. Partant d'une étude (Corby & al., 2000), (Berners-Lee, 2001) reconnaissant la similitude entre le modèle de représentation des graphes conceptuels avec celui du modèle de représentation de données d'un des langages du web sémantique à savoir RDF (Manola et Miller, 2004), nous allons proposer un moyen de passage qui permette à travers un ensemble de règles d'utiliser les points forts des graphes conceptuels dans le web

sémantique. Nous sommes parties de l'idée d'interopérabilité dans le web sémantique qui ne saurait être assurée que si l'on sait établir des relations entre divers formalismes, c'est à dire définir une relation entre des formalismes hétérogènes.

En effet, dans le monde de l'informatique, il n'existe pas de place pour le mot *abandonner* un programme ou un logiciel déjà créé. Il serait intéressant d'exploiter les nombreuses applications en traitement du langage naturel déjà développées à partir des graphes conceptuels, afin de les réutiliser dans le cadre du web sémantique grâce au passage graphe conceptuel (GC) / RDF(S). De plus, lorsque l'on est habitué à écrire des prédicats ou à dessiner des graphes conceptuels représentant des expressions du langage naturel, on peut facilement s'intégrer dans la communauté du web sémantique sans avoir à apprendre le langage RDF(S) par le biais d'interface automatique réalisant ce passage. La prise en compte d'expression du langage naturel ayant fait l'objet d'étude dans le formalisme des graphes conceptuels, a motivé notre choix de ce formalisme. Dans le cadre des recherches réalisées au sein du projet ACACIA, il a été mis en évidence les similarités entre RDF et les graphes conceptuels. Une extension de RDF(S) pour la représentation de connaissances contextuelles basées sur le modèle des graphes conceptuels (Delteil & al., 2003), a été proposée. Les déclarations dans cette proposition sont toutes certaines, dans le sens où l'information est connue et précise. Il n'existe pas de moyen pour exprimer des déclarations comme « il est vrai que », « Peut-être que ... », ... ou encore le caractère vague ou imprécis des concepts et des relations. L'objectif de ce chapitre est d'ajouter ce caractère flou, c'est-à-dire se situer au niveau des concepts en ajoutant une valeur supplémentaire reflétant le caractère imprécis de l'information.

Le détail des différentes règles de passage RDF/GC (Akli, & al., 2010) est décrit dans ce chapitre en l'illustrant par des données forestières de la forêt de Bainem en Algérie (Akli, & al., 2011), issues d'une application en cours de développement. Avant, nous détaillons les notions de base relatives au modèle des graphes conceptuels flous.

3.2 Le modèle des graphes conceptuels flous

Dans ses activités de la vie courante ou professionnelle, chacun est en permanence confronté à l'imperfection des connaissances et des données sur lesquelles il raisonne. Un

grand nombre de termes sont d'ailleurs utilisés dans les langues naturelles pour exprimer les divers aspects de cette imperfection qui sont de nature différentes (Bouchon 1995) :

- *L'incertitude* concerne un doute sur la validité d'une connaissance. Elle peut être aussi présente lorsqu'on donne intentionnellement des informations erronées ou encore, lors d'une difficulté dans l'obtention ou la vérification de la connaissance. Fiabilité relative par l'intermédiaire d'observateur peu sûr de lui ou susceptible de commettre des erreurs engendre aussi de l'incertitude, comme « Je crois que la voiture est proche ».

- *L'imprécision* correspond à une difficulté dans l'énoncé de la connaissance, lorsqu'elle est numérique mal connue, comme « La taille d'un adulte est entre 1.5 mètres et 2 mètres » ou exprimée en langage naturel pour qualifier une caractéristique de façon vague, comme « Température douce, grand appartement ». Dans ce cas, des catégories aux limites mal définies sont utilisées. On parle d'information floue.

- *L'incomplétude* correspond à l'absence de connaissances ou à des connaissances partielles, comme, « généralement, il travaille tous les jours ».

Ces trois types de connaissances sont dépendantes les unes des autres. Les incomplétudes entraînent des incertitudes et les imprécisions peuvent être associées à des incomplétudes.

Exemple

Soit l'information suivante : « on cherche une personne d'environ 30 ans ». On ne peut pas dire que c'est la même chose que « on cherche une personne de 30 ans » ou encore « on cherche une personne dont l'âge est entre 29 ans et quelques mois et 30 ans et quelques mois ». Le traitement d'une telle information nécessite la définition du terme « environ ».

Dans ce qui suit, nous nous sommes intéressés à l'imprécision des connaissances et aux graphes conceptuels flous. Le graphe conceptuel et la logique floue sont deux formalismes répondant davantage au langage naturel. Dans leur combinaison, le graphe conceptuel fournit une structure des formules qui est près du langage naturel tandis que la logique floue (Zadeh, 1965) fournit une méthodologie pour traiter certains mots.

Afin de mieux appréhender la notion de graphe conceptuel flou, une connaissance des ensembles flous et la théorie associée (Zadeh, 1965) est nécessaire. Quelques définitions de base sont données.

3.2.1 Théorie des ensembles flous

Afin de représenter la notion d'imprécision, la théorie des ensembles flous (Zadeh, 1965) se base sur le principe de graduation d'appartenance à un ensemble abandonnant ainsi les contraintes classiques de l'appartenance à un ensemble. Le postulat de cette théorie est de considérer une graduation, définie sur $[0, 1]$, sur l'adéquation à la notion.

Ainsi, la théorie des ensembles flous peut être considérée comme une adaptation de la théorie des ensembles.

3.2.1.1 Concept de sous ensemble flou

Dans le but d'éviter le passage brusque d'une classe à une autre et d'autoriser des éléments à une appartenance partielle à l'une ou l'autre, le concept de sous ensemble flou a été introduit.

La définition de tel sous ensemble vient pour répondre au besoin de représenter des connaissances imprécises exprimées en langage naturel ou obtenues avec des instruments d'observation.

L'idée principale derrière le caractère graduel des sous ensemble flou est que « plus on se rapproche de la caractérisation typique de la classe, plus l'appartenance à cette classe est forte ». La notion de sous ensembles flous permet le traitement des :

- Catégories aux limites mal définies (comme « ancien », « jeune », ...).
- Situations intermédiaires entre le tout et rien (« presque noir », « presque plein »).
- Passages progressifs d'une propriété à une autre (« jeune » à « vieux », de « proche » à « éloigné »).
- Valeurs approximatives (« environ 30 ans »).
- Classe en évitant l'utilisation arbitraire de limites rigides (Difficile de dire qu'une personne de 20 ans est jeune et qu'une personne de 21 ans est moins jeune).

3.2.1.2 Définition d'un sous ensemble flou

Un sous-ensemble flou A sur un domaine de définition X est défini par une fonction d'appartenance μ_A de X dans $[0, 1]$, qui associe à chaque élément x de X , son degré d'appartenance à A :

$$\mu_A : X \rightarrow [0, 1]$$

Remarques :

- Dans le cas où μ_A ne peut prendre que l'une des deux valeurs 0 et 1, l'ensemble A est un sous ensemble classique de X.
- Le support du sous-ensemble flou A est le sous-ensemble de X sur lequel $\mu_A > 0$.
- Le noyau du sous-ensemble flou A est le sous-ensemble de X sur lequel $\mu_A = 1$.

3.2.1.3 Logique floue

Comme la théorie des ensembles flous est une théorie ensembliste qui diffère de la théorie des ensembles, une logique particulière a été développée afin de pouvoir raisonner sur ces ensembles. Il s'agit de la logique floue née en 1965 des travaux de Lotfi A. Zadeh (Zadeh 1965). Eprouvant le besoin de formaliser la représentation et le traitement de connaissances imprécises ou approximatives, il pense au traitement des systèmes d'une grande complexité dans lesquels sont, par exemple, présent des facteurs humains. Plusieurs règles d'inférence ont été proposées dont la généralisation du modus ponens. C'est l'équilibre entre la représentation de l'imperfection et le traitement simple que cherche la logique floue.

La logique floue est l'un des cadres dans lequel sont traitées des imprécisions et des incertitudes, et le cadre dans lequel sont traitées des connaissances numériques et des connaissances exprimées symboliquement par des qualifications du langage naturel.

Des t-normes et t-conormes ont été proposées pour définir le ET et le OU logique. Plus particulièrement, si on considère deux ensembles flous F et G définis sur E, de fonctions d'appartenance f et g, alors selon Zadeh F ET G est un ensemble flou $F \wedge G$ dont la fonction d'appartenance $f \cap g$ est définie comme étant le minimum des deux fonctions d'appartenance, soit :

$$\forall x \in E, (f \cap g)(x) = \min(f(x), g(x))$$

De manière analogue, F OU G, noté $F \vee G$, est défini par le maximum des fonctions d'appartenance :

$$\forall x \in E, (f \cup g)(x) = \max(f(x), g(x))$$

3.2.2 Graphes Conceptuels Flous

En ce qui concerne l'introduction du flou dans les graphes conceptuels, plusieurs travaux ont été menés pour le faire.

L'introduction de la théorie des ensembles flous dans le modèle des graphes conceptuels a été d'abord étudié par Morton (Morton, 1987), puis prolongée par plusieurs travaux comme ceux de (Wuwongse et Manzano, 1993 ; Cao, 1999). Il a ajouté, dans la première extension, des référents flous (fuzzy referents), des opérateurs flous (Fuzzy operations) aux graphes conceptuels en appliquant des incertitudes perceptuelles, linguistiques et propositionnelles équivalent à objet, attribut et information. (Wuwongse et Manzano, 1993) a étendu la théorie des graphes conceptuels par la fuzzification sur les relations conceptuelles. Il a défini les règles d'inférence utilisées dans les graphes conceptuels flous. Cependant, (Wuwongse et Cao, 1996) ont montré des inconvénients dans les travaux des deux extensions citées ci-dessus. Dans (Wuwongse et Manzano, 1993) le degré de compatibilité (Compatibility degree) d'un concept ou d'une relation conceptuelle est défini comme un nombre réel dans un intervalle $[0, 1]$, et une autre raison c'est que leur définition de la projection des GCFs implique des résultats déraisonnables.

L'exemple qui a été donné pour montrer les inconvénients dans les travaux cités précédemment est le suivant :

Un concept d'entité flou représenté par $[T : x|c]$ où « T » est un type conceptuel, « x » est un référent et « c » la compatibilité de « x » (par rapport à T).

Soient $A : [T_a : a|c_a]$ et $B : [T_b : a|c_b]$ deux concepts tels que : T_a est plus spécialisé que T_b et $c_a \leq c_b$. Selon leur définition de projection floue, alors $[T_a : a|c_a]$ est une projection de $[T_b : a|c_b]$, c'est-à-dire, qu'il existe une implication du concept A au concept B. Mais cette dérivation n'est pas raisonnable. Soit l'exemple suivant : « Lila n'est pas un garçon. Elle est une fille ». Selon leur définition du concept d'entité floue, on peut représenter cette phrase par : $[Garçon : Lila|0]$. Mais, ce concept peut entraîner $[Garçon : Lila|1]$, qui exprime « Lila est un garçon ». Ce qui génère un conflit.

Pour pallier ces derniers, un ensemble de primitives est proposé (Wuwongse et Cao 1996) :

- Ensemble de concepts flous S_{TYP} .
- Ensemble de relations floues S_{REL} .
- Ensemble de référents flous S_{REF} .
- Ensemble de mesures floues S_{MEA} qui permet de représenter les attributs comme la couleur, la vitesse et bien d'autres.
- Ensemble de valeurs de vérité floues S_{COM} : Ensemble défini sur un espace des multivaleurs logiques, constitué de 3 classes :

- ✚ Faits positifs : T (very True, fairly True, True);
- ✚ Faits négatifs: F (very False, False, Fairly False);
- ✚ Faits inconnus: U (unknown).

S_{COM} est alors l'union des sous ensembles T, F et U.

De cela, un graphe conceptuel flou n'est autre qu'un graphe conceptuel représenté par un 4-uplets : (V_C, V_R, E, P_C) où :

- V_C : ensemble de nœuds de concepts d'entité et d'attributs, tel que :
 - ✚ Un concept d'entité flou est un triplet sur $S_{TYP} \times S_{REF} \times S_{Com}$ noté $[T : X|c]$ où T : type conceptuel, x : référent et c : comptabilité de x à T. Ce concept possède un dual noté $[\text{Univ} \setminus T : \bar{c}]$, Univ est type de concept générique
 - ✚ Un concept d'attribut flou est un triplet sur $S_{TYP} \times S_{REF} \times S_{MEA}$ noté $[T : x@m]$ où T : Type conceptuel, x : référent et m est la mesure de x.
- V_R : Ensemble de nœuds de relations conceptuelles floues : chacune définie sur $S_{REL} \times S_{Com} \times \mathbb{N}$ noté par $[T|c, n]$, où : T : est un type de relation conceptuelle. c : est la compatibilité de la relation au type T et n : le nombre de référent participant à la relation.
- E : Ensemble des arêtes orientées du graphe.
- P_C : Partition de tous les nœuds de concept par une relation de coréférence.

Dans (Wuwongse et Cao 1996) une proposition a été retenue concernant un type flou définit comme composé du type de base et d'une valeur de vérité floue.

3.3 Passage RDF (S) / graphes conceptuels

Le modèle des graphes conceptuels est un formalisme de représentation proche du modèle de données d'un des langages du web sémantique, à savoir RDF(S). Mais, le langage naturel est connu pour être vague et imprécis. Alors, une extension du modèle des graphes conceptuels pour la prise en compte des informations floues a été proposée, appelé « Les graphes conceptuels floues ».

Dans cette section, notre proposition (Akli, & al., 2010) de règles de passage graphes conceptuels flous \longleftrightarrow RDF(S) est détaillée. Mais avant, l'extension de la classification des concepts ajoutant une dimension supplémentaire pour prendre en compte le caractère flou des

concepts sont présentée. On utilisera des concepts et des relations en rapport avec des données de la forêt de Baïnem en Algérie.

Pour le faire, on commencera par répertorier les correspondances entre les deux modèles afin de bien cerner les points communs et les différences entre ces derniers.

3.3.1 Correspondances entre RDF(S) et les graphes conceptuels

Après une étude détaillée de RDF(S) et les graphes conceptuels nous avons répertorié les points communs suivants :

- Les connaissances terminologiques et assertionnelles sont séparées, dans les deux modèles. La hiérarchie de concepts et de relations des graphes conceptuels est équivalente à celle de classes et de propriétés de RDF et réciproquement.
- Dans les deux modèles, les connaissances assertionnelles sont positives, conjonctives et existentielles.
- Dans les deux modèles, nous retrouvons la représentation graphique qui facilite la visualisation de la connaissance à modéliser :
 - ✚ Un arc étiqueté par une propriété « p » dans un graphe RDF, correspond à un sommet relation de type « p » dans un graphe conceptuel.
 - ✚ Un sommet « ressource » identifié par une URI est traduit par un sommet de concept individuel des graphes conceptuels, dont le marqueur est l'identificateur de la ressource, et dont le type de concept est une classe à laquelle est rattachée la ressource par « rdf:type » dans le graphe RDF.
- Les ressources anonymes de RDF(S) sont représentées par des concepts génériques dans les graphes conceptuels. Le type est connu grâce à sa liaison avec une classe de type « rdf:type ».

Il existe cependant quelques différences que nous mentionnons dans ce qui suit :

- L'arité, associée aux relations, dans RDf est de deux et elle est de type n-aire dans les graphes conceptuels. Les ressources intermédiaires (ressources anonymes) peuvent être utilisées pour relier les ressources par des propriétés binaires.

- Les propriétés RDFS peuvent avoir plusieurs domaines grâce à « rdfs : domaine », tandis que les relations des graphes conceptuels ont un domaine unique. Cette multiplicité de domaines d'une propriété RDF peut être traduite dans le modèle des graphes conceptuels par une relation dont le domaine est le concept le moins général des types de concepts traduisant les domaines de la propriété.

Ces différences ne sont en rien un obstacle pour assurer le passage, entre les graphes conceptuels, RDF et RDFS. En effet, ce sont les points communs qui ont été exploités pour définir les règles de passage. On commencera par le passage le plus évident qui est le passage d'un graphe représentant une déclaration RDF(S) vers un graphe conceptuel.

3.3.2 Passage direct

Après avoir cité les similitudes et les différences entre les deux modèles, il paraît naturel de s'intéresser :

- Aux déclarations RDF pour les traduire en graphes conceptuels.
- A la hiérarchie de classes apparaissant dans un schéma RDF, et qui correspond à une hiérarchie de type de concepts dans un graphe conceptuel.
- A la hiérarchie de propriétés apparaissant dans un schéma RDF et qui correspond à une hiérarchie de type de relations dans un graphe conceptuel.

3.3.2.1 Déclarations RDF

Les déclarations RDF de base sont de la forme <Sujet, Prédicat, Objet> qu'on peut écrire *Prédicat (Sujet, Objet)*. Chaque ressource décrite peut être déclarée dans le modèle des graphes conceptuels comme un concept de type 'ressource' dont le référent est l'URI associée à cette ressource dans la déclaration RDF. Les propriétés des ressources (prédicats des déclarations RDF) sont traduites par des relations conceptuelles.

Exemples:

1. Déclaration, <http://www.usthb.dz/542, Nom, 'Ali'>, qui s'écrit en RDF/ XML :

```
<rdf:Description about =''http://www.usthb.dz/542''>
```

```
  <Nom> Ali </Nom>
```

```
</rdf:Description>
```

Le graphe conceptuel associé:

[Resource : 'http://www.usthb.dz/542'] → (Nom) → [Littéral : Ali]

2. Deux déclarations, <http://www.usthb.dz/542, Nom, 'Ali'> et <http://www.usthb.dz/542, Adresse, 'El Alia'>, s'écrivent en En RDF/ XML :

```
<rdf:Description about ='http://www.usthb.dz/542'>
```

```
<Nom> Ali </Nom>
```

```
<Adresse> El Alia </Adresse>
```

```
</rdf:Description>
```

Le graphe conceptuel associé:

[Resource : http://www.usthb.dz/542] → { → (Nom) → [Littéral : Ali]

→ (Adress) → [Littéral : El Alia] }

On peut donc considérer une description RDF comme une instance de type de concepts « Resource » dans les graphes et les propriétés associées comme les relations liées à ces concepts. Le référent de ce concept de type « ressource » est l'URI correspondante à cette ressource.

3.3.2.2 Descriptions RDF emboîtées

Dans des cas particuliers, les déclarations concernant une ressource peuvent avoir des propriétés (prédicats) dont les valeurs ne sont pas atomiques. Le graphe conceptuel correspondant à la déclaration de cette ressource contient à l'intérieur une autre déclaration de concept de type 'Ressource'. En d'autres termes, on parle de graphe emboîté.

Exemple : En RDF/ XML :

```
<rdf:Description about ='http://www.usthb.dz/542'>
```

```
<Nom> Ali </Nom>
```

```
<Address rdf:Description about ='http://www.usthb.dz/ali'>
```

```
<Num> 32 </Num>
```

```
<Rue> El Alia </Rue>
```

```
<Ville> Alger </Ville>
```

```
</Adresse>
```

</rdf : Description>

Graphe conceptuel associé:

[Resource : 'http://www.usthb.dz/542]-

{ → (Nom) → [Littéral : Ali]

→ (Adresse) → [[Resource : 'http://www.usthb.dz/Ali] → (Num) → [Littéral :32]

→ (Rue) → [Littéral : El Alia] → (Ville) → [Littéral : Alger]] }

Dans le cas de description emboîtée, la translation consiste alors en la création d'un concept typé 'Ressource' pour chaque description de ressource emboîtée.

La ressource emboîtée <http://www.usthb.dz/Ali> est liée par le biais de la relation 'Adresse' à la ressource incorporée '<http://www.usthb.dz/542>'.

3.3.2.3 RDF Schéma

Les descriptions RDF peuvent être typées suivant l'ontologie prédéfinie appelée RDF Schéma par des déclarations précédées par des noms de domaines. Dans ce cas, un concept d'un type associé au domaine correspondant est créé dans les Graphes Conceptuels.

Exemple : En RDF/XML :

```
<rdf : RDF xmlns:rdf=""http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```
  Xmlns:inst=""http://www.usthb.dz/departement#">
```

```
<inst: Local rdf:about ""http://www.usthb.dz/departement/local1234>
```

```
  <inst :type> Laboratoire </inst :type>
```

```
  <inst :Num> Dix </inst :Num>
```

```
</inst :Local>
```

```
</rdf : RDF>
```

Graphe conceptuel associé:

[Local :<http://www.usthb.dz/département/local>]-

{→ (type) →[littéral :Laboratoire]

→(Num) → [Littéral : Dix]}

Il est à noter qu'afin de prévoir les conflits de noms pour les différents schémas, on suppose que les concepts ainsi que les relations doivent être préfixés de la même façon, par le même domaine.

A . Les classes RDFS : Elles peuvent être modélisées comme des types de concepts dans les graphes conceptuels.

A.1. Concept de type Resource : Le sommet de la hiérarchie de type de concept est la « Ressource » du fait que tout dans RDF est une « Resource ». Donc, toute déclaration de type `<rdfs :Class ref : ID ='C1'/>` est translatée dans les graphes conceptuels en : type Concept C1 \leq_c Resource pour dire que « C1 » est un sous type du type « Ressource ».

A.2. Les Sous-classes : Il existe une correspondance claire et directe entre 'rdf :subClassOf' et 'sous_type' dans les GCs.

Exemple : En RDF/XML :

```
<rdfs :Class rdf :ID ='Enseignant'>
  <rdfs :subClassOf rdf :resource ='Personnel_ Université'/>
</rdfs :Class>
```

La classe des enseignants est une sous classe de la classe 'Personnel_ Université'. Puisque la classe est modélisée par un type de concept, la représentation d'un sous_type correspond à la définition d'une classe dont le type est un sous_type de la classe mère.

Graphe conceptuel associé :

Type concept Enseignant \leq_c Personnel _ université.

B. Les propriétés et sous propriétés

B.1. Les propriétés : Une propriété s'applique à une ou plusieurs classes par 'Domain' et prend ses valeurs dans une seule classe spécifiée par 'Range'. Les propriétés sont translatées en des relations dans les graphes conceptuels. La signature de la relation est comme suit : type-relation 'Nom' (Domaine, range). En d'autres termes, le domaine représente la source de la relation et le 'range' spécifie la destination de la relation.

Exemple : En RDF/XML

```
<rdf :Property Id='Enseigne'>
```

```

<rdfs :domain rdf :resource =''#Enseignant ''/>
<rdfs :range rdf :resource =''#Etudiant ''/>
</rdf : Property>

```

La signature de la relation 'Enseigner' est la suivante : Type-relation Enseigner (Enseignant, Etudiant).

Dans RDF, une propriété peut avoir plusieurs classes comme domaine. Car la signature d'une relation conceptuelle est unique. Le passage d'une propriété RDF avec plusieurs domaines vers les relations conceptuelles ne peut se faire directement. Des solutions existent. Il s'agit de :

- Remplacer le domaine de la relation par une relation ancêtre, si les relations de différents domaines en possèdent une.
 - Pour toutes les classes du domaine de la propriété, remplacer par une classe abstraite définie pour ce but,
 - Remplacer par le sommet de la hiérarchie des concepts qui relie toutes les ressources.
- La propriété doit être permise pour toutes les classes.

Exemple : En RDF/XML

```

<rdf : Property ID='Enseigner'>
<rdfs : domaine rdf :resource =''#Enseignant_ Permanent ''/>
<rdfs :domaine rdf :resource =''#Enseignant_ vacataire ''/>
<rdfs :range rdf :resource =''#Etudiant ''/>
</rdf :Property>

```

L'une des solutions consiste à remplacer les deux domaines par une super classe des deux classes représentant le domaine.

Graphe conceptuel associé :

Si Concept type Enseignant :

Concept type Enseignant _ Permanent <_c Enseignant.

Concept type Enseignant _ Vacataire <_c Enseignant.

Type relation Enseigner (Enseignant, Etudiant).

B.2. Les sous propriétés: Afin de raffiner les propriétés existantes, comme dans la définition des sous classes, les ‘sous propriétés’ sont employées.

Si « ρ_1 », « ρ_2 » sont deux propriétés et si $\rho_1 < \rho_2$ et $\rho_1(\text{URI}, C)$ alors $\rho_2(\text{URI}, C)$. En d’autres termes, si une URI possède une valeur « C » par « ρ_1 » et si « ρ_1 » est une sous propriété de « ρ_2 », alors l’URI possède la même valeur par la propriété « ρ_2 ».

Exemple : En RDF/XML

```
<rdf : Property ID='Enseigner'>
  <rdfs : domaine rdf :resource ='#Enseignant ''/>
  <rdfs :range rdf :resource ='#Etudiant ''/>
</rdf :Property>
<rdf : Property ID='Donne-TD'>
  <rdfs : subProperty rdf :resource ='#Enseigner ''/>
</rdf :Property>
```

En termes de graphes conceptuels, une sous propriété RDF est traduite en un type de relation qui est un sous type de relation traduisant la sous propriété.

Grappe conceptuelle associée:

Type relation Enseigner (Enseignant, Etudiant)

Type relation Donne -TD $<_r$ Enseigner.

C. Translation des propriétés supplémentaires de RDF(S)

C.1. La réification : C’est une déclaration à propos d’une déclaration. RDF offre un vocabulaire incorporé pour décrire des déclarations RDF. Ce vocabulaire consiste en un type ‘rdf:Statement’ et des propriétés ‘rdf:Predicate’ qui identifie la propriété originale, ‘rdf:Subject’ qui identifie la ressource décrite dans la déclaration modélisée, ‘rdf:Object’ qui identifie la valeur de la propriété et ‘rdf:type’ qui représente le type de la nouvelle ressource.

Exemple : En RDF/XML

```
<rdf : Description>
  <rdf :Subject rdf :resource : 'http://www.usthb.dz/département/local1234' />
  <rdf :Predicate : rdf :resource = 'http://www.usthb.dz/département/ type' />
  <rdf : Object rdf :resource = 'http://www.usthb.dz/département/laboratoire' />
```

```

    <rdf :type      rdf :resource      ='http://www.w3.org/1999/02/22-rdf-syntax-
ns#Statement' />
  </rdf :Description>

```

Le modèle des graphes conceptuels est muni de moyen d'expression des déclarations imbriquées à savoir les graphes emboîtés via le type de concept 'Statement' et le référent est un type de graphe représentant la déclaration originale.

Graphe conceptuel associé :

```

[Statement :[Local :http://www.usthb.dz/département/local1234]→(type)→[Littéral:Laboratoire]]

```

C.2. Les Containers: La représentation des ensembles est faite dans RDFS grâce aux containers. Ils offrent un moyen de définition de groupe quand la valeur du triplet est un ensemble.

Il existe trois types de containers: « bag », « sequence » et « Alternative ». ils sont tous munis de la propriété <rdf :li> qui permet d'énumérer les composants de l'ensemble dans chacun des cas.

Exemple : En RDF/XML

```

<inst: Local rdf :about ='http://www.usthb.dz/département/locaux' />
  <inst: type>
    <rdf:Bag>
      <rdf :li> laboratoire </rdf :li>
      <rdf :li> Salle-TD </rdf :li>
      <rdf :li> Bureau </rdf :li>
    </rdf :Bag>
  </inst : type>
</inst :Local>

```

Les types des locaux sont: 'laboratoires', 'salle – TD' et 'bureau'. Chacun d'entre eux est précédé par <rdf :li> qui peut être vu comme une propriété qui est translatée à son tour en un type de relation dans les GCs dont la signature est « rdf :li (Container, Resource) ».

Graphe conceptuel

[Local : 'http://www.usthb.dz/département/locaux'] → (type) → [Bag] –
 {→ (rdf:li) →[Laboratoire]
 → (rdf:li) → [Salle-TD]
 → (rdf:li) → [Bureau]}

Il est à noter qu'on a représenté [Bag] sous forme d'un type de concept qui est un sous type du type de concept [Container].

De cette façon, le type de concept abstrait (qui n'a pas d'instances directes) [Container] possède trois sous types qui sont : [Bag], [Séquence] et [Alternative].

C.3. 'Container' et 'aboutEach': Quand une déclaration concerne un groupe d'éléments, l'utilisation de 'aboutEach' sera très utile. Dans l'exemple précédent si on veut dire que les trois types de locaux sont disponibles à l'institut d'informatique, on écrit alors en *RDF/XML* :

```
<inst: Local rdf :about =''http://www.usthb.dz/département/locaux''/>
  <inst: type>
    <rdf:Bag ID:'TYPE'>
      <rdf :li> laboratoire </rdf :li>
      <rdf :li> Salle-TD </rdf :li>
      <rdf :li> Bureau </rdf :li>
    </rdf :Bag>
  </inst : type>
</inst :Local>
<rdf: Description aboutEach:#TYPE''>
  <inst:Dispo> INFORMATIQUE </inst:Dispo>
</rdf :Description>
```

Graphe conceptuel associé :

[Local : 'http://www.usthb.dz/département/locaux'] → (type) → [Bag] –
 {→ (rdf:li) →[Laboratoire] →(Dispo)→[Informatique]
 → (rdf:li) → [Salle-TD] →(Dispo)→[Informatique]
 → (rdf:li) → [Bureau] →(Dispo)→[Informatique] }

Le passage de RDF(S) vers les graphes conceptuels est un passage plus ou moins évident du fait qu'il est basé sur les ressemblances entre les deux modèles. Cette évidence a lieu parce qu'on se base plus sur une compréhension des déclarations RDF(S) plutôt que sur celle des déclarations exprimées en langage naturel. Le passage inverse est bien sur moins évident.

3.3.3 Passage inverse

Afin de réaliser le passage des graphes conceptuels flous vers RDF(S) cas par cas, les règles de passage doivent prendre en compte la classification des concepts proposée par (Sowa, 1991) et que cette même présentation peut être étendue pour la prise en charge du caractère flou des concepts donnés dans (Wuwongse et Cao 1996) :

On changera légèrement cette classification selon que le référent soit connu ou pas, qu'il corresponde à un seul ou plusieurs élément(s).

3.3.3.1 Classification des concepts avec introduction du flou

Concept générique : Un concept est dit générique si son référent n'est pas connu. Dans ce cas, le concept est une interprétation d'un objet existant de l'univers de discours mais il est non identifiable.

Exemple : [Personne], [université]

Du fait que le référent soit inconnu, la compatibilité du référent au type de concept dépend essentiellement de celui-ci. On peut donc dire que la compatibilité est inconnu ($U \subset S_{COM}$). On peut alors l'écrire : [Personne : * |Unknown], [Université : * |Unknown].

Concept individuel : Un concept est dit individuel, si le référent est connu. Le concept est une interprétation d'un objet identifiable et représenté par le référent.

Exemple : [GrandHomme : Réda| vrai] où Réda désigne un grand homme.

Dans ce cas, le référent est connu. La compatibilité du concept à son type conceptuel est aussi connue. Ce concept est alors soit un concept d'entité flou, soit un concept d'attribut flou suivant que celui-ci puisse être mesuré ou non.

Concept ensembliste : Dans les cas où le référent n'est pas un seul élément mais un ensemble, la compatibilité de l'ensemble représentant le référent au type conceptuel dépend aussi du concept (Sowa, 1991). Le degré de compatibilité ou la mesure de compatibilité

(suivant le cas) sont les mêmes pour tous les éléments de l'ensemble. Les cas possibles de ces référents sont:

- *Ensemble générique.* Le référent peut être un ensemble dont les éléments ne sont pas identifiables.

Exemple : [Personne :{*}] désigne plusieurs personnes. Dans un graphe conceptuel flou, cette écriture sera alors [Personne : {*}|C] où C est la compatibilité de l'ensemble au type conceptuel, qui peut être « Unknown » aussi.

- *Ensemble de cardinal spécifié.* On peut préciser le nombre d'éléments faisant référence à un concept.

Exemple : [Personne :{*}@5 |C] pour dire cinq Personnes ayant une compatibilité C à ce concept.

- *Un ensemble fini en extension.* On peut énumérer les éléments en les écrivant dans un ensemble (entre accolade).

Exemple : [PersonnePresent : {Ali, Reda, Zoheir}|C] pour dire que les personnes présentes sont : Ali, Réda et Zoheir et sont compatibles avec un degré C.

- *Un ensemble partiellement spécifié.* C'est un ensemble dont on ne connaît pas tous les éléments.

Exemple : [Personne :{Ali,*}|C] pour dire « Ali et bien d'autres ».

- *Les ensembles collectifs.* Ces ensembles sont utilisés pour dire que le reste du graphe conceptuel est appliqué à l'ensemble des éléments.

Exemple : Pour dire qu'il y a trois personnes qui possèdent des enfants, on écrit [Personne : Col{*}@3|C] → (Possède) → [Enfant]

- *Les ensembles respectifs.* Ils sont utilisés pour exprimer le fait que chaque élément de l'ensemble entretient une relation avec un élément d'un autre ensemble.

Exemple : [Personne : Resp{Ali, Réda}|C1] → (Frèrede)→[Fille:Resp{Nawel, Ratiba}|C2]. [Personne : Resp{Ali, Réda }| C1] sont respectivement les frères de Nawel et Ratiba.

Remarque :

Dans les exemples, ci-dessus, nous avons considéré des concepts précis où $C = \text{Vrai}$. Le formalisme ainsi modifié permet de représenter les concepts et les relations vagues et imprécises dans les différents cas. Notre but est de ramener une telle représentation au web sémantique. Pour cela, il est nécessaire de rapporter les points communs entre les Graphes Conceptuels flous et RDF(S) et de donner par la suite l'ensemble des propositions pour le passage des graphes conceptuels vers RDF(S).

3.3.3.2 Les règles de Passage

Il paraît clair que les deux types de représentation (RDF, graphes conceptuels classiques et flous) présentent des points communs surtout en ce qui concerne les concepts.

Un concept dans les graphes conceptuels, classiques et flous, correspond à une ressource RDF ou à une classe RDF(S) ayant un type. La hiérarchie des concepts est définie grâce à «RDFS :subClassOf». D'un autre côté, une relation dans les graphes conceptuels devient une propriété RDF dont la précision de la source et la destination est faite grâce à «rdfs :Domain» et «rdfs :range».

Dans les graphes conceptuels flous, les concepts sont soit des concepts d'entité floue, soit des concepts d'attribut flou. La représentation des deux doit donc être différente.

- ***La représentation des concepts d'entité floue dans RDF(S)***

La reformulation de la signification de cette notation de concept $[T :x|c]$ est la suivante : Le concept, dont *le référent* est x , a une compatibilité c avec *un type de concept* T .

Le concept est une ressource RDF dont le type est la Super-classe associée et le référent est une instance possible. La compatibilité peut être représentée par une propriété associant une valeur à une instance. A ce niveau, il faut définir une classe pour le type de concept. Le référent est alors une instance de ce concept.

Exemple : Soit le concept $[\text{Homme} : \text{Réda}|\text{Vrai}]$. Dans ce cas, on va simplement créer une classe Homme. Réda ne sera alors qu'une instance de la classe Homme.

La compatibilité est de 3 types. La super-classe Compatibilité a trois (03) sous-classes qui sont classe positive, classe négative et classe inconnue. En RDF/ XML on écrira :

```
<rdfs : Class rdf :ID="ConceptEntitFlou" >
```

```
<rdfs : Class rdf :ID="Compatibility" >
```

```

<rdfs:Class rdf:ID='Positive'>
  <rdfs:subClassOf rdf:resource=''#Compatibility'/'>
</rdfs:Class>
<rdfs:Class rdf:ID='Negative'>
  <rdfs:subClassOf rdf:resource=''#Compatibility'/'>
</rdfs:Class>
<rdfs:Class rdf:ID='Unknown'>
  <rdfs:subClassOf rdf:resource=''#Compatibility'/'>
</rdfs:Class>

```

Le degré de compatibilité du référent au concept est connu grâce à une relation « Estcompatible » comme suit :

```

<rdfs:Property rdf:ID='Estcompatible'>
  <rdfs:Domain rdf:resource=''#ConceptEntitFlou'/'>
  <rdfs:Range rdf:resource=''#Compatibility'/'>
</rdfs:Property>

```

Sachant que la classe «compatibilité» possède trois sous classes à savoir «positive», «négative» et «unknown», on définit alors pour chaque cas une propriété.

```

<rdfs:Property rdf:ID='PositiveCompatibilité'>
  <rdfs:subPropertyOf rdf:resource=''#Estcompatible'/'>
  <rdfs:Domain rdf:resource=''#ConceptEntitFlou'/'>
  <rdfs:Range rdf:resource=''#Positive'/'>
</rdfs:Property>
<rdfs:Property rdf:ID='NegativeCompatibilité'>
  <rdfs:subPropertyOf rdf:resource=''#Estcompatible'/'>
  <rdfs:Domain rdf:resource=''#ConceptEntitFlou'/'>
  <rdfs:Range rdf:resource=''#Negative'/'>
</rdfs:Property>
<rdfs:Property rdf:ID='InconnuCompatibilité'>
  <rdfs:subPropertyOf rdf:resource=''#Estcompatible'/'>
  <rdfs:Domain rdf:resource=''#ConceptEntitFlou'/'>

```

```
<rdfs:Range rdf:resource =''#Unknown''>
</rdfs:Property>
```

- ***La représentation des concepts d'attribut flou dans RDF(S)***

Pour les concepts d'attributs flous, on peut appliquer la même idée. Seulement, les mesures de compatibilité ne sont plus des valeurs qui appartiennent à la classe compatibilité, mais une classe qui contient des mesures subjectives.

A ce niveau, on a vu que la représentation de la compatibilité est facile par RDF(S). Il est important, maintenant, de spécifier comment représenter les types de concepts et les relations déjà définis dans la section précédente.

A. Les Concepts

A-1. Concept générique : Un concept est générique, si son référent n'est pas connu. En RDF/XML, on peut définir une classe pour les concepts génériques et instancier par la suite les objets de cette classe par référent défini.

Exemple : Soit le concept générique [Personne]. En RDF, on peut l'écrire comme suit :

```
<rdf:Description rdf:about =''http://www.université.dz/Personne''>
  <rdfs:Type rdf:resource =''http://www.w3.org/TR/rdf-Schema#Class''>
</rdf:Description>
```

Ou encore, on l'écrit :

```
<rdfs:Class rdf:about =''://www.université.dz/Personne''/>
```

A-2. Concept individuel : Dans ce cas, le référent est connu. Le concept correspond à l'interprétation d'un objet identifiable. Il s'agit en RDF de la déclaration d'une classe représentant le type de concept associé au référent (l'élément identifiable) et il restera à déclarer par la suite un élément de cette classe correspondant au référent lui-même.

Exemple : On peut déclarer John Smith comme une personne et cette personne est le créateur de la page web particulière. Le concept individuel dans cette déclaration est [Personne : John Smith]. En RDF, on peut l'écrire comme suit :

```
<rdf:RDF
  xmlns:rdf = ''http://www.w3.org/1999/02/22-rdf-syntax-ns#''>
  xmlns:dc = ''http://purl.org/metadata/dublincore-core#''>
```

```

<rdf :Description about :’http:// www.ourorg.net/metadata.html’>
<dc:Creator>
  <rdf :Description  rdf:about =’http://www.ourorg.net/member/JohnSmith>
    <rdf:Type  rdf :resource =’http://www.universit .dz/Personne’/>
  </rdf:Description>
</dc:Creator>
</rdf: Description>
</rdf:RDF>

```

A-3. Concepts ensemblistes

- *Ensemble g n rique.* Le r f rent dans ce type de concept est un ensemble non identifiable. Pour ce type de d claration, on peut utiliser les possibilit s de RDF dans le cas de d claration d’ensembles. Et puisque dans les ensembles g n riques, il n’y a aucun ordre   respecter, l’utilisation des containers (type Bag) est recommand e car le nombre d’ l ments de l’ensemble n’est pas limit  au pr alable.

- *Ensemble de cardinal sp cifi .* Le r f rent de ce type de concept peut  tre un ensemble dont on conna t d j  le nombre d’ l ments le constituant. On peut dans ce cas, utiliser comme dans le cas pr c dent les possibilit s de RDF dans la d finition de groupes ou d’ensembles en pr cisant le nombre de n uds pouvant exister. Cet ensemble sera comme une liste cha n e dont la fin est marqu e par NIL. Les Collecteurs ont exactement cette fonction de d limitation des ensembles. Et puisque les  l ments de cet ensemble ne sont pas identifiables, chaque n ud aura alors comme valeur une URI se terminant par « _n » o  n est un entier diff rent de z ro.

Exemple : Trois personnes est un ensemble dont le cardinal est  gal   3, [Personne : {*}@3]:

```

< rdf : RDF>
  xmlns: rdf=’http://www.w3.org/1999/02/22-rdf-syntax-ns#’>
  xmlns : gc=’http://www.universit /termes#’>
<rdf :Description  rdf :about :’http://www.universit /Personne’>
<gc :HasReferent>
  <rdf : ParseType=’Collection’>
  <rdf :Description  rdf:about=’http://www.universit /Personne/_ 1’/>
  <rdf :Description  rdf:about=’http://www.universit /Personne/_ 2’/>

```

```

    <rdf:Description rdf:about="http://www.universit /Personne/_3"/>
</rdf:Description>
</rdf:RDF>

```

- *Ensemble fini en extension.* Dans ce type de concepts, les  l ments constituant l'ensemble peuvent  tre  num r s.

Exemple : Soit un ensemble de trois (03) personnes R1, R2 et R3 : [Personne : {R1, R2, R3}].

On le d clare comme une collection RDF :

```

<rdf:RDF>
xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#>
xmlns:gc="http://www.universit /termes#">
<rdf:Description rdf:about:"http://www.universit /Personne">
<gc:HasReferent>
<rdf:parseType="Collection">
    <rdf:Description rdf:about="http://www.universit /Personne/R1"/>
    <rdf:Description rdf:about="http://www.universit /Personne/R2"/>
    <rdf:Description rdf:about="http://www.universit /Personne/R3"/>
</rdf:RDF>

```

- *Ensemble partiellement sp cifi .* Dans ce cas, on ne conna t pas tous les composants de l'ensemble. L'ensemble est compos  de l' l ment « R1 » et bien d'autres. Dans RDF(S), on d clare un « Container », de type « bag », ayant comme premier  l ment celui qu'on conna t « R1 ». On sait qu'il n'existe pas de moyen de fermer un container mais, on peut aussi ajouter ici, une URI qui se termine par « _n », o  n est un entier diff rent de z ro.

```

<rdf:RDF>
xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#>
xmlns:gc="http://www.universit /termes#">
<rdf:Description rdf:about:"http://www.Universit /Personne">
<gc:HasReferent>
<rdf:Bag> <rdf:Li rdf:resource="http://www.universit /Personne/R1"/>
    <rdf:Li rdf:resource="http://www.universit /Personne/_1"/>
</rdf:Bag>
</rdf:Description>

```

</rdf : RDF>

- *Ensemble collectif*. Ces ensembles sont utilisés dans le cas où la même partie du graphe est appliquée à tous les éléments de l'ensemble. RDF permet l'utilisation de containers comme « rdf :Bag ». L'utilisation de l'attribut « rdf :aboutEach » dans l'élément <rdf :Description> permet ensuite d'affecter des propriétés avec la même valeur à un ensemble de ressources. L'utilisation de l'attribut « BagId » dans l'élément <Description> permet lui de réifier les propriétés contenues dans la déclaration RDF.

Exemple : On veut déclarer trois articles ayant les codes 11, 12 et 13. Ces articles appartiennent tous au même auteur YM. En RDF, on écrit :

< rdf : RDF>

```
xmlns: rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
```

```
xmlns : dc ="http://purl.org/metadata/dublincore-core#"
```

```
<rdf :Description rdf :about : "http://www.library/Articles/Info.htm">
```

```
<bagID= "Classement">
```

```
<dc :Subject rdf :resource="http://www.Articles/info/code# 11"/>
```

```
<dc :Subject rdf :resource="http://www.Articles/info/code# 12"/>
```

```
<dc :Subject rdf :resource="http://www.Articles/info/code# 13"/>
```

```
</rdf : Description>
```

```
<rdf :Description aboutEach ="Classement">
```

```
<dc : Creator> YM </dc : Creator>
```

```
</rdf :Description>
```

```
</rdf : RDF>
```

- *Ensembles respectifs*. Chaque élément de l'ensemble respectif entretient une relation avec un élément de l'autre ensemble. En RDF, on peut modéliser cela en spécifiant pour chaque élément, l'élément correspondant par le biais de la propriété représentant la relation entre les deux. En d'autres termes, si un concept [T : resp{R1, R2}] est relié par une relation « Rel » avec un autre concept [T' : resp{R3,R4}], on déclare alors, des relations de type Rel entre chacun des éléments. Et donc les triplets (Concept1, relation, concept2) sont : (R1, Rel, R3) et (R2, Rel, R4). On déclare alors deux propriétés «Rel» et «Rel1» telles que : «Rel» ayant comme «rdfs :Domain» les deux classes «R3» et «R4», «rdfs :Range» la classe «R1».

«Rel1» ayant comme «rdfs:Domain» les deux classes «R3» et «R4», «rdfs:Range» la classes «R2» bien sûr en précisant à chaque fois laquelle des classes est instanciée.

B. Relation conceptuelle

Les relations conceptuelles expriment l'existence d'un lien entre des concepts. La notation de relation dans les graphes conceptuels flous peut être interprétée comme suit : « il existe une *relation* de type *R*. Elle relie un *concept Source* et un *concept destination* et possède un *nombre de référents* participant à cette relation ». La relation est alors une propriété RDFS ayant comme source une ressource indiquée par «rdfs:Domain» et comme destination une ressource indiquée par «rdfs:range». Le type de la relation peut être déterminée grâce à « rdfs:subPropertyOf ».

Il existe des types de relations conceptuelles prédéfinies, comme : 'Agent', 'Expérienceur', 'instrument', etc. Les relations peuvent être représentées par des propriétés RDF.

Exemple :

- La relation 'Agent' doit avoir comme « rdfs:domain » une classe de type 'Action' et un « rdfs:range » une classe de type « Acteur ».
- La relation de type 'Expérienceur' s'applique aux classes de type 'Etat' (rdfs:domain) et prend ses valeurs dans les classes de types 'ANIME' .

Cependant, il existe des types de relations 'STAT' et 'STMT' qui sont des relations spéciales et qui représentent en réalité des graphes emboîtés dont leur prise en compte se fera par des descriptions RDF emboîtés dans des conteneurs de types « Bag ». Chaque élément de ce « Bag » est une déclaration RDF exprimée grâce au principe de réification.

Mais, ces déclarations réifiées dans le « bag » sont lourdes à traiter et difficiles à comprendre. Pour cela, une extension de RDFS en DRDFS (Defined Resource Description Framework Schema) a été définie (Delteil, & al., 2003). Des classes spéciales comme « DefinedClass », « DefinedProperty » et « Context » ainsi que des propriétés comme « isContextOf » sont spécifiées pour exprimer ces connaissances contextuelles.

A ce niveau, il nous reste qu'à déterminer la façon de représenter le nombre de référent. Il est important de noter qu'une relation RDFS (Propriété RDFS) ne peut être que binaire. Pour représenter le degré de compatibilité de la relation conceptuelle à son type, elle peut être vue comme une relation n-aire entre à la fois les deux classes conceptuelles et la classe de compatibilité. En d'autres termes, cela peut être lu comme suit : la Classe représentant le

concept « C1 » est liée à la classe représentant le concept « C » par une relation R avec un degré de compatibilité « Comp » à son type conceptuel. Pour représenter ce lien, on peut utiliser les nœuds anonymes (sans identifiant précis). Voici le Schéma correspondant :

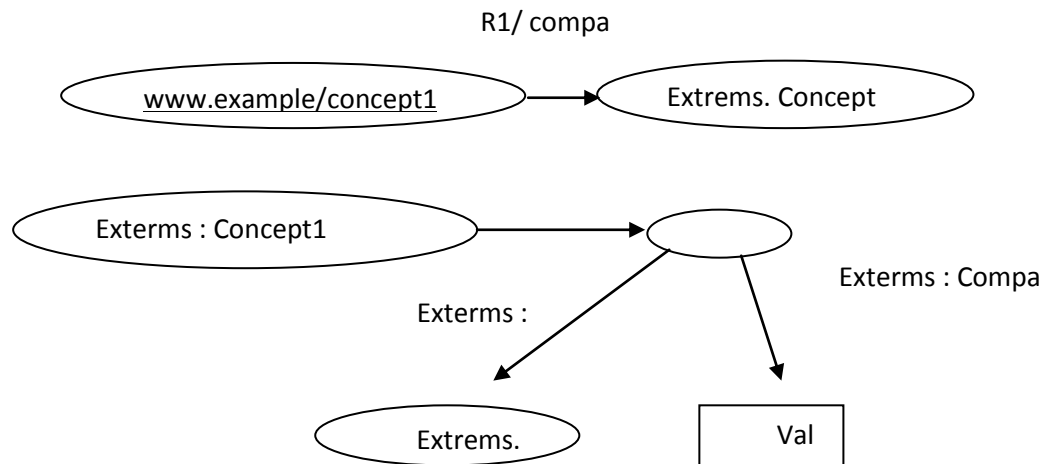


FIG. 3.1 – Déclaration RDF avec un nœud anonyme reflétant un concept flou

C. Annotations syntaxiques

Les annotations syntaxiques permettent de préserver les nuances syntaxiques une fois que les phrases en langage naturel sont traduites en graphes conceptuels. Pour le faire en RDF (S), l'utilisation de « rdfs : Comment » sera suffisante.

Exemple : Soit le graphe suivant : [Chat : Minou ; n] → (Agnt ; Subj) ← [Etat ; V]

En RDF/ XML on peut l'écrire

```
<rdfs :Class rdf :ID ='Chat'>
  <rdfs:Comment> nom </rdfs:Comment>
</rdfs :Class>
<rdfs :Class rdf :ID ='Etat'>
  <rdfs:Comment> Verbe </rdfs:Comment>
</rdfs :Class>
<rdf :property rdf :ID ='Agnt'>
  <rdfs:domain rdf: resource ='#Chat'>
  <rdfs: range rdf: resource ='#Etat'>
  <rdfs:Comment> Subj </rdfs:Comment>
</rdfs :Class>
```

A ce niveau, la représentation des concepts et des relations conceptuelles, avec prise en compte du flou, peut être faite par les graphes conceptuels flous. Grâce aux règles de passage proposées, leur représentation par des déclarations RDF(S) est devenue possible. Dans ce qui suit, nous allons appliquer ces règles dans le web sémantique géo spatial.

3.4 Application –Géo Spatial Sémantique Web-

En vue de développement de systèmes intelligents, plusieurs défis scientifiques et technologiques doivent être relevés, comme par exemple la technologie sémantique nécessaire pour résoudre les problèmes de pertinences des données et de la surcharge d'information qui deviennent de plus en plus sévères. En ce qui concerne les données et les services sur le web, cette approche fondée sur la sémantique est incorporée dans les activités du web sémantique géo spatial (Egenhofer, 2002). Ce dernier regrouperait les activités autour du développement d'ontologies spatiales et temporelles pour la description sémantique d'informations à références spatiales et temporelles accessibles sur le Web. La sémantique dans ce cas est donc très importante. Nous allons mettre en avant son lien avec l'information géo spatial.

3.4.1 L'information Géo Spatial et la sémantique

« Une information est dite géographique lorsqu'elle se rapporte à un ou plusieurs lieux de la surface terrestre. C'est une information localisée, repérée, ou encore « géocodée » (Béguin et Pumain, 2003). On peut la formaliser comme suit : un objet i est localisé en un lieu, situé dans le système de coordonnées terrestres x et y , dans lequel on repère la latitude x_i et la longitude y_i de l'objet. Plusieurs attributs peuvent être ajoutés complétant ainsi sa définition. Donc, décrire une donnée consiste entre autres à définir une composante descriptive sémantique relative à la qualification de l'objet (Bordin, 2002). Un certain vocabulaire technique est utilisé et donc n'est pas forcément compris par différentes communautés. Un même terme peut en effet représenter des concepts différents. La sémantique appliquée à l'information spatiale suppose de déterminer la signification de certains termes au sein d'une communauté et de comprendre quels termes sont utilisés par cette même communauté pour représenter différents concepts. Par exemple, le terme « forêt », pour les données forestières, a un sens qui peut paraître simple pour cette discipline. Mais, dans d'autres communautés, chez

les militaires par exemple, il peut désigner une superficie couverte à 70% d'arbres. Le même terme est donc utilisé pour désigner deux concepts différents.

Comprendre la sémantique de l'information spatiale est un problème plus complexe qu'il n'y paraît. Il s'agit en effet de formaliser ces données pour faciliter leur traduction entre les différentes communautés. C'est l'objet des ontologies qui définissent le vocabulaire commun afin de le rendre lisible par l'ordinateur. Cette traduction suppose une compréhension commune d'un certain nombre de termes de base ou leurs relations entre eux. Il s'agit de donner une sémantique, c'est à dire du « sens » aux informations échangées et de s'assurer que ce sens soit disponible dans tous les systèmes où des échanges sont mis en œuvre. La prise en compte de cette sémantique permet à ces systèmes de combiner les informations reçues avec d'autres informations et de les traiter de manière appropriée par rapport à cette sémantique.

Cette approche fondée sur la sémantique est incorporée dans les activités du web sémantique géo spatial (Egenhofer, 2002) que l'on désignera aussi par Web Géo Spatial.

3.4.2 Qu'est ce que le Web Géo Spatial ?

Selon Schell (Schell, 2007), le Web Géo spatial est un domaine complexe, situé à la croisée de plusieurs domaines de recherche parmi lequel on peut citer : la géographie physique, la géomatique, la représentation de connaissances, les réseaux, la logique, etc. Son objectif est semblable à celui du WS : associer aux ressources Web des descriptions (méta-données) ici, spatio-temporelles, interprétables par les humains, et surtout par les machines, afin que le traitement automatisé de ces données par des agents logiciels soit rendu plus performant. Les métadonnées sont donc centrales. Nous les considérons du point de vue de la représentation des connaissances.

Les données géo spatiales sont organisées dans des catalogues contenant les enregistrements de métadonnées, qui peuvent être interrogés afin de donner accès à des ressources connexes. Ils permettent aux utilisateurs de trouver les données dont ils ont besoin et déterminer la meilleure façon de les utiliser (Nebert, 2001). Il offre une description de la ressource sur le contenu, la qualité, la précision, l'auteur, et d'autres caractéristiques.

Il existe plusieurs travaux sur les métadonnées géographiques (Nebert, 2001), (Comber, & al., 2007), (Pathan, 2009), etc. Plusieurs normes ont été établies par les organismes de normalisation tels que ISO, FGDC (<http://www.fgdc.gov/>), l'OGC

(<http://www.Opengeospatial.Org/>), NSDI-Inde (<http://www.nsiindia.gov.in/NSDI/>) ou DCMi (<http://dublincore.org/>) qui sont en charge de coordonner l'élaboration de normes sur les métadonnées de contenu pour les données spatiales et temporelles. Toutefois, l'organisation actuelle de l'information géographique dans les catalogues de métadonnées est incapable de saisir toute la sémantique des données décrites en raison de l'imperfection des connaissances.

En effet, beaucoup de concepts dans les applications géo spatiales sont imprécis. La nécessité de raisonner à partir de tels concepts implique une logique de l'imprécis où les notions de vrai et de faux sont insuffisantes pour rendre compte de la valeur de certaines affirmations.

Nous nous attacherons ici aux métadonnées liées aux ressources, plus particulièrement à leur qualité et leur pertinence pour savoir si une ressource répond aux besoins d'un utilisateur.

3.4.3 Qualité de l'information géographique

La qualité des données est une information relative à la différence entre les données et la réalité qu'elles représentent. Ainsi, plus la qualité des données est faible, moins ces données nous apprennent sur le monde géographique et moins ce qu'elles nous apprennent a de la valeur (Goodchild, 2006). En effet, Les données géographiques proviennent de sources variées et il semble facile de les combiner par leur référence spatiale et de les utiliser pour analyser un problème précis et prendre une décision. Cependant, nous pouvons nous demander quelle est la valeur d'une décision basée sur des données dont la qualité est mal connue ou mal comprise par le décideur ?

La problématique de la qualité de l'information peut être alors associée à la gestion de l'imperfection des données, c'est à dire à la gestion de l'imprécision, de l'incertitude, de l'ambiguïté, de l'incomplétude, etc... (Dubois et Prade, 2001). Dans notre démarche, la qualité des données est prise en considération dans sa représentation.

Nous avons montré (Akli, & al., 2010) que pour représenter un objet géo spatial par une donnée géographique, la pertinence de l'information dépendait largement de la qualité de la données. En effet, si on observe attentivement les images de la figure 3.2, on constate rapidement qu'une image est plus claire que l'autre et que ses objets sont plus facilement détectables. Ceci est du D'où, on conclut qu'une image est de meilleure qualité que l'autre.

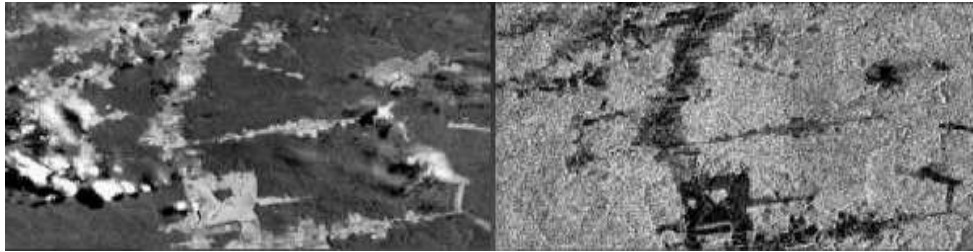


FIG. 3.2 -- Deux images de la même zone de la forêt amazonienne. Image de gauche LANDSAT-TM, et image de droite radar JERS-1 (source: Corina Freitas et Eymar Lopes, l'INPE)

Le problème maintenant, est qu'entendons nous par qualité de l'information géographique, comment peut on détecter automatiquement qu'une ressource est de meilleur qualité qu'une autre et en quoi l'imperfection des données est elle liée à leur qualité?

3.4.3.1 Imperfection des données géo spatiales

Si l'utilisation de données géo spatiales était, il ya quelques années, le champ des utilisateurs experts utilisant des systèmes complexes et coûteux, la réalité a changé de manière significative. Les logiciel SIG et les données géo spatiales sont maintenant disponibles à faible coût, voire gratuitement sur l'Internet et sont de plus en plus facile à utiliser. Les données géo spatiales ne sont plus gérés uniquement au niveau opérationnel dans les organisations, mais aussi au niveau stratégique et tactique (Longley, & al., 1999). Ils sont maintenant de plus en plus utilisés pour soutenir le processus décisionnel de la sélection d'itinéraires pour planifier nos vacances, dans la gestion d'un réseau routier par des organismes gouvernementaux, etc. De plus, avec le développement de services géo localisés (LBS) et les technologies sans fils, il sera probablement commun pour tout le monde dans un avenir proche pour prendre des décisions fondées sur des données géo spatiales affichés sur les téléphones mobiles, les systèmes mobiles, systèmes de navigation dans les voitures, etc. Mais, comment se fier à une décision si l'information est incertaine, imprécises, incomplètes, etc. ? En effet, on peut trouver beaucoup d'exemples liés à cette imperfection :

- Pour *l'incertitude* par exemple, l'information peut être issue d'erreurs de :
 - Mesure d'enregistrement provoquant une erreur de localisation de l'objet.
 - Classement où l'objet est affecté à une mauvaise classe d'objets à cause d'une erreur dans l'expertise.

- Regroupement de classes où l'objet est regroupé avec des objets aux propriétés différentes.
- Enregistrement avec un mauvais codage en entrée du SIG.
- Temporelle où l'objet a changé de caractéristique entre la date d'enregistrement dans la base et celle de son utilisation.
- Traitement. une erreur est causée pendant le traitement par un arrondi ou une erreur Algorithmique.

➤ *L'imprécision*, est due au caractère vague ou approximatif de la sémantique utilisée. Le caractère vague de l'information est la conséquence d'une insuffisance des instruments d'observation et de l'utilisation de connaissances subjectives ou flexibles. L'approximatif est issu de catégories aux limites mal-définies.

Nous voyons bien que l'imperfection est liée à la qualité des données. On ne peut donc prétendre à un système fiable et de qualité sans une gestion de cette imperfection.

3.4.3.2 Définition de la qualité des données

Deux définitions de la qualité des données peuvent être identifiées dans la littérature (Aalders and Morrison, 1998; Aalders, 1999). La première, appelée qualité interne, limite la qualité des données aux caractéristiques internes des données, découlant des méthodes de production des données (acquisition des données, modèles de données, etc.). La seconde, appelée qualité externe, correspondant au niveau d'adéquation existant entre les caractéristiques des données et les besoins des utilisateurs pour différents aspects (par exemple couverture spatiale et temporelle, actualité, etc..). La qualité des données est une composante importante pour sa description, de plus chaque unité de données doit diffuser l'information décrivant sa qualité. Il est nécessaire de connaître la qualité de l'information afin de pouvoir l'utiliser efficacement.

3.4.3.3 Composantes de la qualité

Aujourd'hui la série de normes ISO/1911x représente la convergence de la plupart des travaux de normalisation de la qualité des données. Les composantes principales sont la cohérence logique, l'exhaustivité, la qualité géométrique, la qualité sémantique, l'actualité et la généalogie (Berengere, & al., 2004). A travers la présentation de ces composantes, on mettra l'accès sur la nécessité de prise en compte de l'incertitude et de l'imprécision :

Cohérence logique : la cohérence logique définit le degré de cohérence interne de données selon les règles de modélisation et de spécification du jeu de données. Elle ne dépend pas du terrain nominal, mais uniquement des spécifications du modèle conceptuel de données et de la connaissance de l'univers. Cette composante est toujours liée à une contrainte donnée. On distingue deux types : les contraintes reliées aux données physiques et celles d'intégrité définies dans le schéma conceptuel issu de la spécification du jeu de données. Supposons que nous avons un ensemble de contraintes d'intégrité sur les attributs. A chaque contrainte, on associe une cohérence logique dont la valeur mesure le degré auquel le jeu de données respecte ces contraintes. Par exemple, supposons que la spécification du produit précise que seules les constructions les plus proches d'un endroit considéré doivent figurer dans le jeu de données. La question est de savoir comment considérer le qualificatif subjectif *plus proche* pour considérer comme erreur de cohérence logique toute habitation présente dans le jeu de données qui ne le sera pas.

Exhaustivité : Cette composante décrit la conformité de la présence ou de l'absence des éléments du jeu de données par rapport au terrain nominal. Elle s'attache aux objets, aux attributs et aux relations. La définition de l'exhaustivité utilise dans son modèle le taux de déficit et le taux d'excédent.

- *le taux de déficit* (ou l'omission) est le taux de données manquant par rapport au terrain nominal, c'est à dire existant dans le terrain nominal mais pas dans le jeu de données.

- *le taux d'excédent* (ou commission) est le taux de données présentes dans le jeu de données mais qui ne se trouvent pas ou sont indéterminées dans le terrain nominal. Une évaluation de ce taux pouvant être réalisé par un qualificatif subjectif comme par exemple un *taux plus ou moins élevé, non significatif*, etc.

- *Précision sémantique* : la précision sémantique concerne l'information descriptive (information thématique). Cette composante de la qualité est le plus souvent considérée par les utilisateurs pour évoquer la question de la qualité des données géographiques. Nous considérons cette notion comme une compatibilité des éléments des valeurs des éléments du jeu de données avec ceux du terrain nominal.

Actualité : cette composante, également appelée qualité temporelle, représente le décalage entre le jeu de données à une date t1 et le terrain nominal à une date de référence t2. L'actualité peut être vue de deux manières différentes. Pour le producteur, l'actualité des données implique une politique de mise à jour décrivant la validité du jeu de données avant la

prochaine modification. Pour l'utilisateur, l'actualité des données doit lui permettre de vérifier la validité des données au moment de leur utilisation.

Généalogie : la généalogie fournit une expression qualitative décrivant l'histoire des données. Elle donne des informations concernant les sources d'acquisition de données, les méthodes utilisées et les opérations appliquées. Cette composante permet aussi d'indiquer les références complètes des sources (la description, la date de création, l'organisme producteur, etc.), des données (zone couverte, classes concernées, objectif de la création, etc.) et des opérations (description, date d'application, équations et paramètres, etc.). De cette généalogie, un utilisateur peut construire un indice de confiance global, ou plusieurs indices relatifs aux autres aspects de la qualité.

3.5 Conclusion

Dans ce chapitre, nous avons représenté les concepts par les graphes conceptuels flous permettant ainsi la représentation de certaines expressions du langage naturel. Puis, nous avons proposé un certain nombre de règles de passage de ces types de graphes vers l'un des standards du web sémantique, RDF(S).

Mais le web sémantique, ce n'est pas seulement les métadonnées. C'est aussi les ontologies qui ont gagné en popularité dans la communauté de l'IA, générant de nombreuses ontologies qui se chevauchent dans le contenu. Si on considère, le domaine de l'anatomie par exemple, il ya plusieurs ontologies qui les représentent. On trouve le référentiel ontologie à la National Center pour les ontologies Biomedicales (biointology.org). Même parmi le petit nombre d'ontologies bien acceptées et largement utilisées, plusieurs contiennent la représentation de l'anatomie humaine: le modèle fondamental de l'anatomie, le thésaurus de l'Institut national du cancer, l'ontologie GALEN (Noy, 2009). Cette situation n'est pas surprenante puisque différentes applications exigent des points de vue différents sur les différentes représentations du domaine. Cependant, si nous voulons avoir des applications utilisant des ontologies différentes et qui interagissent entre elles, ou si nous voulons intégrer des données annotées avec ou structurées selon différentes ontologies, nous devons d'abord trouver les correspondances entre leurs concepts. Le processus de recherche de ces correspondances, appelé *appariement d'ontologies*, est un axe de recherche sur les ontologies très actif. Il fait l'objet d'étude du chapitre suivant.

Chapitre 4

Appariement des ontologies

4.1 Introduction

Le problème de la gestion de l'hétérogénéité de diverses sources d'information est croissant. Par conséquent, des solutions diverses ont été proposées pour faciliter le traitement de cette situation, et plus précisément, pour automatiser l'intégration des sources d'information distribuées. Parmi elles, on trouve les technologies sémantiques pour lesquels notre intérêt s'est porté. Pour surmonter cette hétérogénéité sémantique, deux étapes sont généralement préconisées (Shvaiko et Euzenat, 2012) à savoir, i) appairer les entités afin de réaliser un alignement ; ii) interpréter l'alignement en fonction des besoins des applications comme la traduction des données ou la réponse aux requêtes.

Dans ce chapitre, nous nous concentrons uniquement sur la première étape qui est l'appariement des ontologies. Il s'agit d'un processus de comparaison de deux ontologies et de découverte de relations ou de correspondances entre leurs entités. Ces correspondances peuvent être utilisées pour diverses tâches, telles que la fusion d'ontologies, répondre à des requêtes, ou encore traduire des données. Ainsi, l'appariement des ontologies permet d'exprimer les connaissances et les données en respectant les ontologies appariées à inter opérer (Euzenat et Pavel, 2007). Un engouement certain a été porté à cette thématique ces dix dernières années. En effet, des solutions diverses d'appariement ont été proposées (Batini, & al., 1986 ; Spaccapietra & Parent, 1991), des états

de l'art ont été faits (Kalfoughlou & Schorlemmer, 2003) ; (Rahm et Bernstein, 2001 ; Gal et Shvaiko, 2009) et des livres (Euzenat et Pavel, 2007 ; Bellahsene, & al., 2011) ont été publiés.

Afin de mieux appréhender cette notion, différents travaux traitant du processus d'appariement d'ontologies sont étudiés. Mais avant, des problèmes liés à l'hétérogénéité des ontologies qu'un processus d'appariement peut résoudre sont examinées.

4.2 L'hétérogénéité des ontologies

La mise en commun de connaissances représentées par des ontologies développées séparément est confrontée au problème de leurs structures et de leurs sémantiques. Dans la littérature, plusieurs travaux relatifs à la classification des types de disparités entre les ontologies existent (Goh, 1997; Wache, & al., 2001; Klein, 2001,), etc. Dans cette section, nous présentons la classification la plus rappelée dans la littérature (Klein, 2001) dans laquelle deux niveaux de disparité ont été distingués. Il s'agit de disparité au niveau du langage et celle au niveau des ontologies (voir figure 4.1):

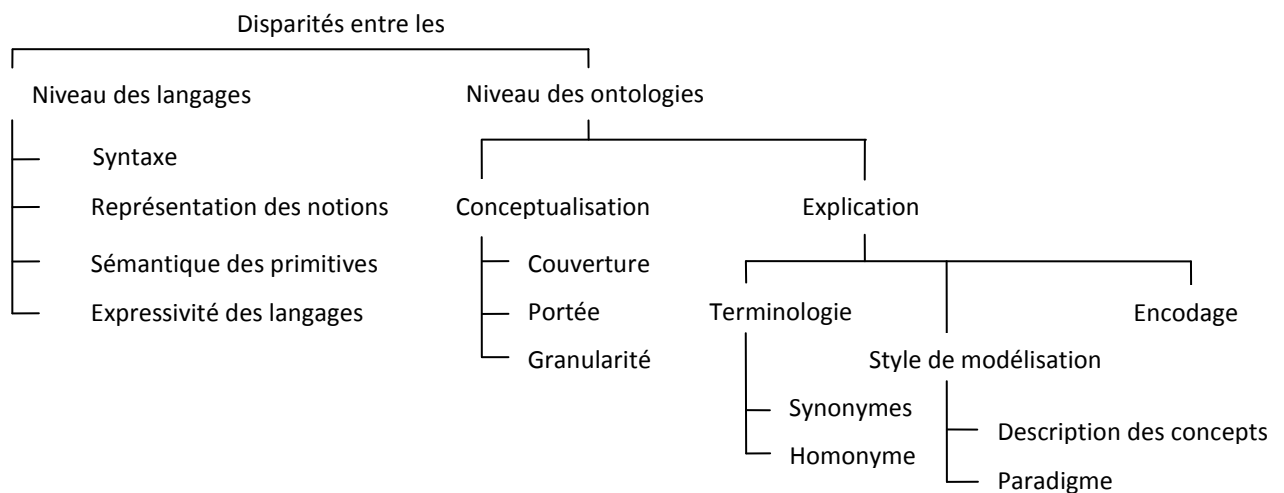


FIG. 4.1 – Disparités des ontologies (Klein, 2001)

4.2.1 Niveau des langages

Les disparités au niveau des langages se produisent lorsqu'on veut combiner des ontologies écrites dans des langages différents. Elles surgissent dans les mécanismes utilisés pour définir les concepts, les relations, etc.

Elles peuvent apparaître dans la *syntaxe*. Si on veut exprimer le concept « chaise », cela ne se fait pas de la même manière dans RDF(S) et OIL. On utilisera pour le premier langage, la primitive «<rdfs : Classe ID= "Chaise">», et le constructeur «class-def Chaise » dans le deuxième langage. La résolution de ce type de disparité est généralement mise en œuvre par un processus de translation ou de traduction.

On peut retrouver des disparités dans la représentation *des notions logiques*. Par exemple, pour représenter la disjonction de deux classes, certains langages permettent de l'énoncer explicitement par la primitive « disjoint A B »; tandis que d'autres exigent d'utiliser la négation dans des relations de sous-classe : « A subclass-of (NOT B), B subclass-of (NOT A) ».

Une disparité plus délicate au niveau des langages peut être détectée lorsque deux langages contiennent des *primitives* nommées identiquement mais avec des *sémantiques* différentes. On peut citer l'exemple de la déclaration successive de plusieurs primitives <rdfs : domain>. Dans OIL, cette déclaration signifie une intersection d'arguments. Tandis que dans RDF(S), elle est interprétée comme une union d'arguments.

La possibilité de traduire une ontologie écrite dans un langage donné vers un autre langage est strictement liée à la puissance d'expression des langages source et cible. La divergence dans la puissance d'expression signifie qu'un langage permet d'exprimer des choses qu'un autre langage ne permet pas. Par exemple, le langage KIF (*Knowledge Interchange Format*) permet d'exprimer toute la logique du premier ordre, tandis qu'OWL DL n'exprime qu'un sous-ensemble de cette logique. Par conséquent, la translation d'une ontologie vers un langage de moindre expressivité risque de provoquer une perte d'information. Cependant, il est important de noter que ces dernières années, avec l'acceptation de RDFS et OWL pour représenter les ontologies, le problème de l'hétérogénéité des ontologies au niveau du langage est devenu beaucoup moins important (Noy 2009).

4.2.2 Niveau des ontologies

Ces disparités se produisent lorsque des ontologies modélisées différemment ou décrivant des domaines partiellement recouvrant sont combinées. Elles sont de deux types: les disparités de conceptualisation et les disparités d'explication.

4.2.2.1 Conceptualisation

Ce type de disparités ne peut pas être résolu automatiquement. Il exige les décisions et la connaissance d'un expert du domaine. Quatre (4) types de disparités de conceptualisation sont distinguées: la couverture, la portée, la granularité (Klein, 2001) et la perspective (Bouquet, & al., 2004). Ces quatre notions (voir figure 4.2) sont détaillées dans ce qui suit :

- *Couverture.* Les ontologies diffèrent aussi souvent par leur couverture d'un domaine particulier. Par exemple, une ontologie O_1 modélise les voitures et les camions, pendant qu'une autre ontologie O_2 ne modélise que les camions.
- *Portée.* Les disparités de la portée surgissent lorsque deux classes semblent représenter le même concept, sans posséder les mêmes instances. Par exemple, toutes les administrations ont la même compréhension du terme « Employé ». Cependant, dans la pratique, un employé aura des droits et des devoirs différents d'une entreprise à une autre.
- *Granularité.* La granularité, avec laquelle les distinctions sont faites, peut être aussi une source de disparité entre les ontologies. Une ontologie de fine granularité est une ontologie très détaillée, possédant ainsi un vocabulaire riche capable d'assurer une description détaillée des concepts d'un domaine. Une ontologie de large granularité correspond à un vocabulaire moins détaillé. Par exemple, l'ontologie O_1 représente les camions sous des catégories basées sur leur structure physique, poids, but, etc., tandis que l'ontologie O_2 ne prend pas en considération tous ces détails.
- *Perspective.* Ce type de disparités surgit lorsqu'une ontologie représente un point de vue sur un domaine donné différent de celui représenté dans une autre ontologie. Par exemple, on peut définir deux ontologies décrivant les composants microélectroniques avec deux points de vue différents: l'une concerne l'aspect fabrication et l'autre concerne l'aspect marketing ou utilisation.

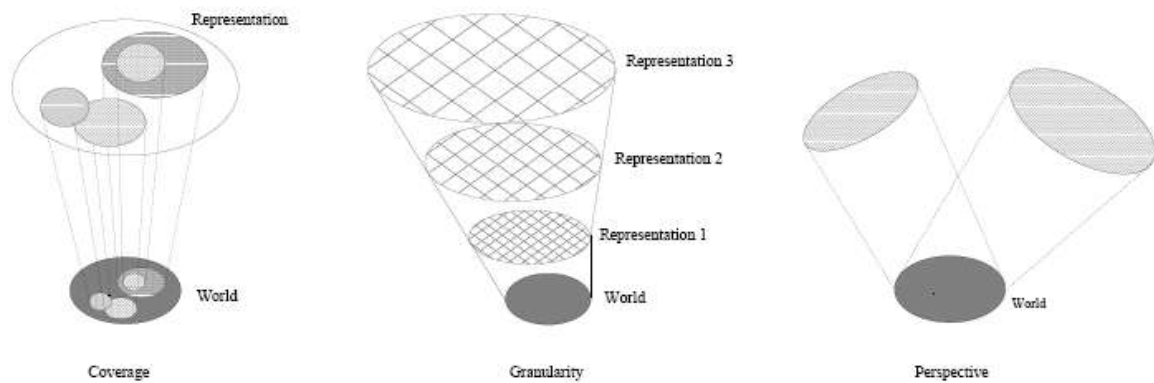


FIG. 4.2. Les disparités de conceptualisation : Couverture, Granularité et Perspective (Bouquet, & al., 2004).

4.2.2.2 Explication

Les disparités d'explication sont de trois types. Elles concernent les styles de modélisation, les terminologies et l'encodage des données.

- *Style de modélisation* : Les ontologies diffèrent dans leurs styles de modélisation sur les paradigmes et les descriptions des concepts qu'elles utilisent.

- Paradigme. Des paradigmes différents peuvent être utilisés pour représenter des concepts comme le temps, l'action, etc. Par exemple, on pourra trouver des ontologies qui décrivent un cercle par un point et un rayon, alors que d'autres le décrivent par un ensemble de trois points.
- Description des concepts. Plusieurs choix peuvent être faits pour la modélisation des concepts dans une ontologie. Par exemple, le concept « Personne » peut être modélisé soit avec la construction de deux sous-concepts « Mâle » et « Femelle », ou seulement avec la définition d'un attribut « sexe » pour déterminer le sexe de la personne.

- *Terminologie* : Les disparités terminologiques sont des disparités liées au processus de nommage qui associe un objet linguistique aux entités décrites dans une ontologie. Les exemples typiques de ces disparités sont l'utilisation de termes synonymes ou homonymes :

- Termes synonymes. Deux ontologies peuvent utiliser des termes synonymes pour se référer aux mêmes entités comme dans le cas de l'utilisation du terme « voiture » dans une ontologie et le terme « automobile » dans une autre.
- Termes homonymes. La signification d'un terme est différente d'un contexte à l'autre. Par exemple, le terme « Conducteur » possède une signification différente dans le domaine de la musique que dans le domaine du génie électrique. La connaissance humaine est exigée pour résoudre cette ambiguïté.

- *Encodage des données* : L'encodage des données au sein des ontologies diffère souvent, que ce soit pour les dates, les unités (monnaie, distances, ...), etc. Dans la plupart des cas, une étape de transformation est suffisante pour éliminer ces disparités.

4.3 Appariement, alignement et similarité

Afin de cerner et de comprendre l'intérêt de l'opération d'appariement d'ontologies et des concepts de base qui lui sont associés, une discussion sur un exemple motivant est donnée.

4.3.1 Exemple illustratif

Il s'agit de la reproduction de l'exemple donné par (Euzenat et Shvaiko, 2007) qui consiste à développer une application pour une université donnée. Cette application vise à fournir une interface unique permettant de gérer deux ontologies différentes. La première ontologie a été développée dans le but de donner une vue globale sur la gestion de l'université, tandis que la deuxième ontologie représente une vue d'un laboratoire de recherche particulier. Les développeurs de cette application doivent essentiellement appairier ces deux ontologies afin de fournir un alignement permettant une manipulation cohérente des deux ontologies.

La figure suivante montre les deux ontologies à appairier ainsi que les différentes correspondances entre leurs entités.

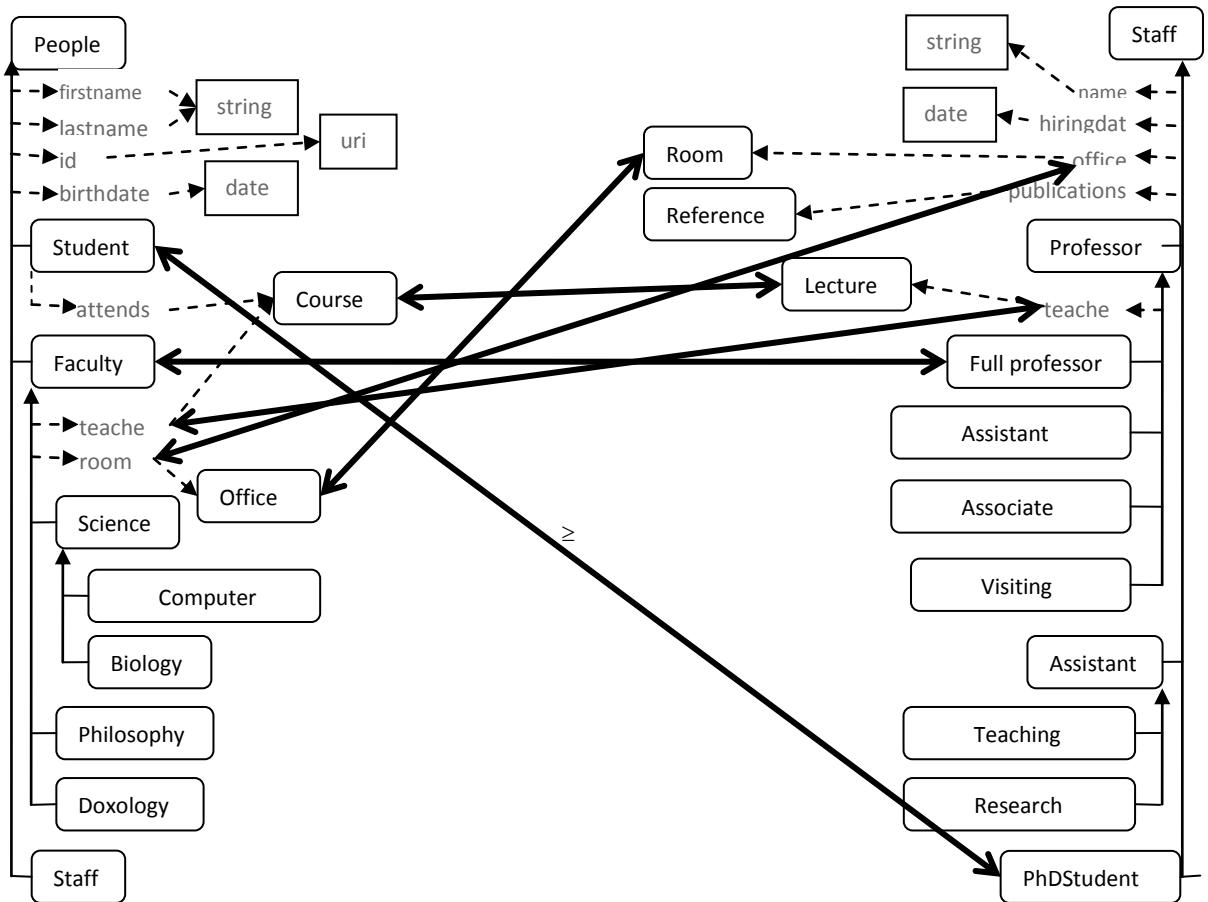


FIG. 4.3 – Exemple d'alignement entre deux ontologies (Euzenat & Shvaiko, 2007).

Les liens pleins représentent les différentes correspondances entre les entités. La relation d'équivalence (\equiv) et le degré de confiance (1) sont pris par défaut. Autrement, ils sont mentionnés, comme c'est le cas dans cet exemple :

Student \geq PhDStudent	Course $\equiv_{0.7}$ Lecture
Faculty $\equiv_{0.9}$ Full professor	teaches \equiv teaches
room \equiv office	Office \equiv Room

4.3.2 Définitions des concepts de base

Dans ce chapitre, nous adhérons à la terminologie la plus récemment proposée qui est celle de (Euzenat & Shvaiko, 2007). Ce choix est dû au fait qu'elle possède une très fine granularité par rapport à celles de (Klein, 2001; Noy et Musen, 2001; Ehrig, 2007).

Dans un premier temps, nous distinguons entre l'opération de découverte de correspondances des résultats obtenus. L'opération est dite « appariement » tandis que l'ensemble des correspondances qui en résulte est dit « alignement ».

4.3.2.1 Appariement, correspondance et alignement

L'appariement des ontologies est le processus de comparaison de deux ontologies et de découverte de relations ou de correspondances entre leurs entités. L'appariement est alors défini comme suit (Euzenat, & al., 2007 b) :

Définition (Appariement d'ontologies). L'appariement de deux ontologies O_1 et O_2 consiste à trouver les correspondances existantes entre leurs entités.

Le résultat du processus d'appariement est appelé « alignement ». Il exprime les correspondances existantes entre des entités appartenant à des ontologies différentes. Dans une correspondance, on trouve, les entités des ontologies avec la relation supposée exister entre elles. L'alignement peut concerner deux ontologies. On parlera alors d'*alignement simple* qui se présente comme des ensembles de paires d'entités des deux ontologies auxquels on associe des relations (voir figure 4.3). Lorsqu'on veut aligner plus de deux ontologies, on parle de *multi-alignement*, c'est-à-dire plusieurs alignements simples. Dans les deux cas, l'alignement peut être *complexe* ou *multiple*. Si on reprend l'exemple précédent, on peut rendre l'alignement complexe par l'ajout d'une correspondance entre les propriétés « *firstname* » et « *lastname* » de la première ontologie et la propriété « *name* » de la deuxième ontologie. Cependant, l'utilisation de relations binaires pour exprimer de telle correspondance est inappropriée. En général, les opérateurs tels que la concaténation, les opérateurs arithmétiques ou les connecteurs logiques sont utilisés. Ainsi, la correspondance précédente est exprimée par : « $firstname^lastname \equiv name$ ». On peut aussi parler de la multiplicité d'un alignement. Il concerne le nombre de correspondances reliant les entités. Dans l'exemple d'exécution de la figure 4.3, l'alignement n'est pas multiple puisque sa multiplicité est $?:?$, c'est-à-dire, les entités des deux ontologies sont impliquées dans au plus une correspondance. Cependant, si on ajoute la correspondance $Professor \geq Faculty$, l'alignement devient $?:*$. Si on considère que l'alignement relie toute entité de la deuxième ontologie avec une entité de la première ontologie, cet alignement devient $?:+$. Les figures suivantes montrent quelques configurations possibles pour l'alignement de deux ontologies composée chacune de trois entités.

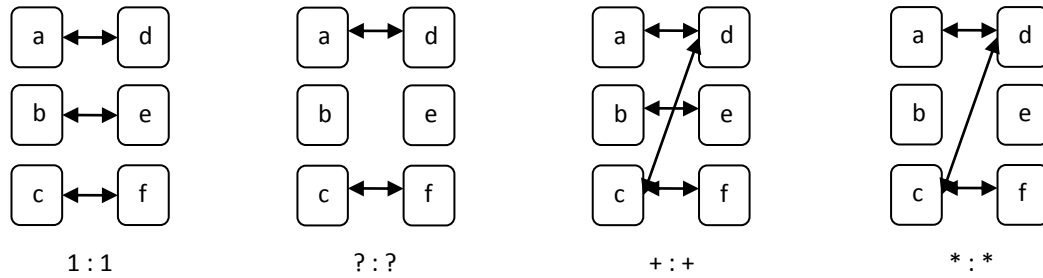


FIG. 4.4. – Quelques configurations possibles pour l’alignement de deux ontologies (Euzenat & Shvaiko, 2007)

Les entités d’une ontologie peuvent être les concepts, les relations, les fonctions, les instances, (\supseteq), l’opérateur de recouvrement (\cap) ou l’incompatibilité (\perp). En outre, on peut aussi avoir d’autres types de relations telles que les relations floues et les distributions probabilistes. L’ensemble de ces relations est dénoté par θ .

Pour des raisons pragmatiques les axiomes, les règles, etc. Les types de relations possibles entre deux entités de deux ontologies peuvent être principalement les relations d’équivalence (\equiv) et/ou de subsumption

, un degré de confiance est assigné à chaque relation liant deux entités. Ce degré de confiance signifie la probabilité d’existence de la relation et il est calculé par des mesures qui estiment la similarité des deux entités en question. Il est défini comme suit (Euzenat & Shvaiko, 2007) :

Définition (Structure de confiance). Une structure de confiance est un ensemble ordonné de degrés ($\Xi \leq$) pour lesquels il existe un plus grand élément T et un plus petit élément \perp .

La structure de confiance la plus utilisée dans les systèmes d’appariement d’ontologies est l’intervalle unitaire des nombres réels $[0,1]$. Cependant, d’autres structures sont également possibles telles que le treillis booléen et les degrés flous.

Ainsi, les correspondances sont définies à partir des éléments précédemment décrits, i.e., les entités des deux ontologies, la relation d’alignement et le degré de confiance, comme suit (Euzenat, & al., 2007 a).

Définition (Correspondance). Étant données deux ontologies O_1 et O_2 , un ensemble de relations d’alignements θ et une structure de confiance sur Ξ , une correspondance est un quintuple $\langle id, e_1, e_2, r, n \rangle$ tel que :

- id est un identifiant unique pour la correspondance en question ;
- $e_1 \in O_1$ et $e_2 \in O_2$;
- $r \in \theta$;
- $n \in \Xi$.

La correspondance $\langle id, e_1, e_2, r, n \rangle$ affirme que la relation r existe entre les entités e_1 et e_2 avec une confiance égale à n . Un alignement est donc défini (Euzenat & Shvaiko, 2007) :

Définition (Alignement). Étant données deux ontologies O_1 et O_2 , un alignement est construit à partir d'un ensemble de correspondances entre les paires d'entités appartenant à O_1 et à O_2 respectivement.

Une fois l'appariement de deux ontologies établi, on dit que l'entité e_1 est alignée avec l'entité e_2 s'il existe une correspondance ayant un degré de confiance satisfaisant.

4.3.2.2 Similarité, dissimilarité, distance et autres mesures

Afin de trouver les relations entre les entités exprimées dans des ontologies différentes, il est nécessaire de mesurer la quantité par laquelle ces entités se rapprochent, i.e., leur degré de similarité. De ce fait, la similarité joue un rôle important dans le processus d'appariement des ontologies. La similarité et les notions qui lui sont rattachées, comme la dissimilarité et la distance, font l'objet d'étude de cette section. Leurs définitions sont extraites de (Euzenat, & al., 2004) :

A. La similarité

La notion de similarité varie selon le contexte dans lequel elle est utilisée. En psychologie, par exemple, elle se rapporte à comment les attitudes, les valeurs, les intérêts et la personnalité présentent des correspondances entre les personnes. Tandis qu'en topologie mathématique, elle est définie par une fonction sur un ensemble de points. La valeur donnée par cette fonction est plus grande quand deux points sont plus proches.

Dans le contexte de l'appariement des ontologies, la notion de similarité est en rapport avec la similarité sémantique. Elle représente une évaluation du lien sémantique entre deux entités dont le but est d'estimer le degré par lequel les entités sont proches dans leur sens (Resnik, 1995).

La similarité sémantique de deux entités telle qu'elle est définie dans (Lin, 1998) est liée aux caractéristiques que ces entités possèdent en commun. Elle repose sur les suppositions suivantes :

- Plus les entités ont des caractéristiques communes, plus elles sont similaires ;

- Moins les entités possèdent de caractéristiques communes, moins elles sont similaires ;
- La similarité maximale est obtenue lorsque deux entités sont identiques.

Définition (Similarité). Étant donné un ensemble d'entités O , une similarité $\sigma: O \times O \rightarrow \mathbb{R}$ est une fonction à partir d'une paire d'entités vers un nombre réel exprimant la similarité entre deux entités telle que :

$$\begin{aligned} \forall x, y \in O, \sigma(x, y) &\geq 0 && \text{(positivité)} \\ \forall x, y, z \in O, \sigma(x, x) &\geq \sigma(y, z) && \text{(maximalité)} \\ \forall x, y \in O, \sigma(x, y) &= \sigma(y, x) && \text{(symétrie)} \end{aligned}$$

B. La dissimilarité

Parfois, la fonction inverse de celle de la similarité est utilisée. Il s'agit de la fonction de dissimilarité dont la définition est la suivante.

Définition : Étant donné un ensemble d'entités O , une dissimilarité $\delta: O \times O \rightarrow \mathbb{R}$ est une fonction à partir d'une paire d'entités vers un nombre réel telle que :

$$\begin{aligned} \forall x, y \in O, \delta(x, y) &\geq 0 && \text{(positivité)} \\ \forall x \in O, \delta(x, x) &= 0 && \text{(minimalité)} \\ \forall x, y \in O, \delta(x, y) &= \delta(y, x) && \text{(symétrie)} \end{aligned}$$

C. La distance

La distance permet de mesurer la dissimilarité entre deux entités. Si la valeur de la fonction de similarité de deux entités est élevée, la distance entre elle est petite et vice-versa (Bach, 2006).

Définition : Une distance $\delta: O \times O \rightarrow \mathbb{R}$ est une fonction de dissimilarité satisfaisant la définitivité et l'inégalité triangulaire :

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

D. Normalisation de la similarité

Les mesures précédemment définies sont, très souvent, normalisées pour pouvoir les interpréter d'une manière probabiliste et les combiner dans des formules plus complexes (Bach, 2006). La réduction de chaque valeur à la même échelle en proportion de la taille de l'espace considéré est la manière commune de normaliser (Euzenat, & al., 2004).

Définition : Une (dis) similarité est dite normalisée si elle s'étend sur l'intervalle unitaire des nombres réels $[0, 1]$. Une version normalisée d'une (dis) similarité σ (respectivement, δ) est dénotée par $\bar{\sigma}$ (respectivement, $\bar{\delta}$).

E. La représentation de la similarité

Selon les définitions précédentes, la similarité et la dissimilarité sont des fonctions qui font correspondre à des paires d'entités des nombres réels. Une représentation alternative plus commune d'une telle fonction de similarité sur un ensemble fini d'entités est une matrice. La matrice a l'avantage d'être une structure de données finie qui peut être échangée entre les programmes (Euzenat, et al., 2004; Ehrig, 2007).

Exemple. L'application d'une mesure de similarité sur quelques noms de concepts de deux ontologies d'université est donnée par la matrice suivante.

Ontologie 1\ontologie2	People	Student	Faculty	Science	Philosophy	Boxology	Staff	Course	Office
Staff	.56	.65	.33	.64	.12	.11	.63	.22	.13
Professor	.62	.36	.60	.40	.44	.32	.55	.21	.36
Assistant	.40	.44	.58	.62	.46	.33	.43	.32	.22
PhDStudent	.64	.92	.45	.60	.65	.52	.55	.33	.34
Room	.12	.20	.20	.18	.10	.12	.09	.11	.62
Reference	.23	.06	.18	.25	.26	.28	.22	.17	.23
Lecture	.15	.16	.26	.23	.34	.12	.14	.70	.16

TAB. 4.1– Matrice de similarité

Cette matrice indique par exemple que la valeur de similarité entre « Staff » de l'ontologie 1 et « People » de l'ontologie 2 est égale à 0.56 ; entre « Staff » de l'ontologie1 et « Student » de l'ontologie2 est égale à 0.65, etc.

F. Les niveaux de similarité

Dans (Hakimpour, 2003), quatre niveaux de similarité entre deux entités sont définis : l'égalité (ou l'équivalence), la spécialisation, le recouvrement, et l'incompatibilité (ou la disjointure). Ces niveaux correspondent à l'ensemble de relations θ défini dans la section 3.2.:

- *Incompatibilité.* C'est le niveau ayant le moindre degré de similarité. Deux entités, e_1 et e_2 , sont disjointes si elles ne possèdent aucune caractéristique commune, i.e., la conjonction de leurs caractéristiques est vide. Par exemple, « camion » et « employé ».

- *Recouvrement.* Si la conjonction des caractéristiques de deux entités e_1 et e_2 ne peut pas être prouvée comme vide alors e_1 et e_2 seront recouvrantes. Cela veut dire qu'il est possible qu'une instance de e_1 soit aussi une instance de e_2 .

- *Spécialisation.* (respectivement Généralisation). L'entité e_1 est un hyponyme (respectivement hyperonyme) de l'entité e_2 si « e_1 est une sorte de e_2 » (respectivement « e_2 est une super entité de e_1 ») est vrai. Par exemple, « Homme » est un hyponyme de « Personne ». De ce fait, toute instance de e_1 (respectivement e_2) est aussi une instance de e_2 (respectivement

e_1). La similarité de spécialisation (respectivement généralisation) est une relation partiellement ordonnée.

- *Egalité*. Ce niveau possède le degré de similarité le plus haut. « L'entité e_1 est égale à l'entité e_2 » si e_1 et e_2 possèdent les mêmes caractéristiques. Donc, toute instance de e_1 est aussi une instance de e_2 et vice versa. Par conséquent, si deux entités sont égales, chacune d'entre elles spécialise et généralise l'autre à la fois. Par exemple, les concepts « Véhicule » et « Automobile » sont égaux car elles possèdent les mêmes caractéristiques.

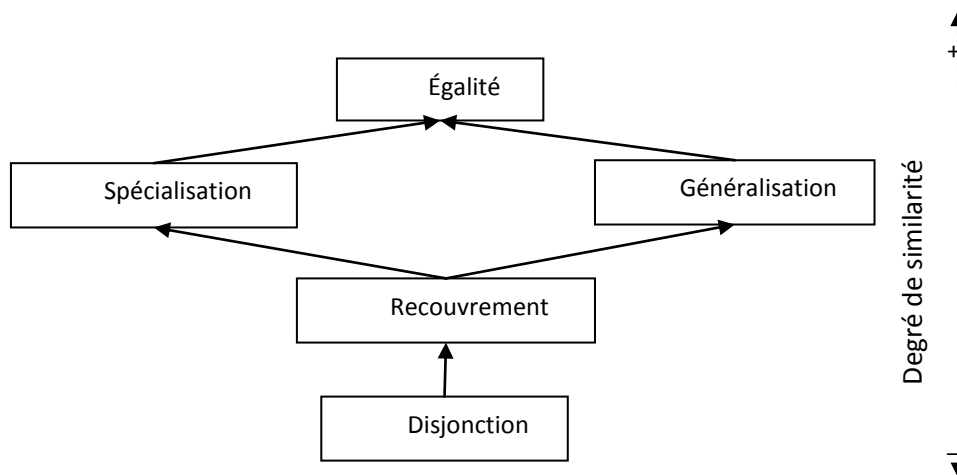


FIG.4.5 – Degrés de similarité entre les entités de différentes ontologies (Hakimpour, 2003)

4.4 Les fondements de l'appariement des ontologies

L'analyse des fondements de l'appariement des ontologies passe par trois notions : le niveau de complexité sémantique décrivant les caractéristiques des ontologies qui doivent être considérées dans le processus d'appariement, les techniques associées et le processus lui-même.

4.4.1 La complexité sémantique des ontologies

Lors du développement d'une ontologie, choisir le langage dans lequel sera exprimée et utilisée une ontologie est une décision importante à prendre. Des contraintes sont à considérer comme par exemple *la lisibilité* et *la portabilité* (Baneyx, 2007). En effet, le langage doit avoir une certaine continuité avec le langage naturel pour être lisible et compréhensible par un

utilisateur humain, et être le plus standard possible afin d'être réutilisé dans d'autres systèmes. De plus, des critères comme *l'expressivité* qui est liée au nombre d'éléments pouvant être utilisés pour décrire les composants d'une ontologie et *la performance* du langage qui est en rapport avec la capacité et la facilité de représentation des connaissances du domaine sont utilisées (Gandon, 2002). Cependant, d'après (Fensel, 2001), il semblerait que plus un langage est expressif moins il est performant, et vice versa. Il existe aussi un autre dilemme entre l'expressivité et le raisonnement dans le sens où l'expressivité d'un langage doit être parfois limitée afin d'assurer un bon service de raisonnement (Izza, 2006).

Par ailleurs, dans le cadre du WS, le W3C a mis en place un groupe de travail dédié au développement de langages standards pour représenter des ontologies utilisables et échangeables sur le Web assurant ainsi les contraintes de lisibilité et de portabilité. Ce groupe a publié trois langages qui semblent aujourd'hui faire un consensus. Il s'agit de RDF (S), d'OWL (Ontology Web Language) et de SWRL. Cependant, ces langages n'ont pas tous la même puissance d'expression.

Afin de faciliter le choix du langage ontologique dans ce cadre, des niveaux de complexité sémantique des ontologies ont été identifiés (Castano, & al., 2007) où une association entre ces niveaux et les langages ontologiques a été proposée (voir figure 4.6). Chaque langage fournit des constructions pour exprimer les caractéristiques correspondantes à son niveau.

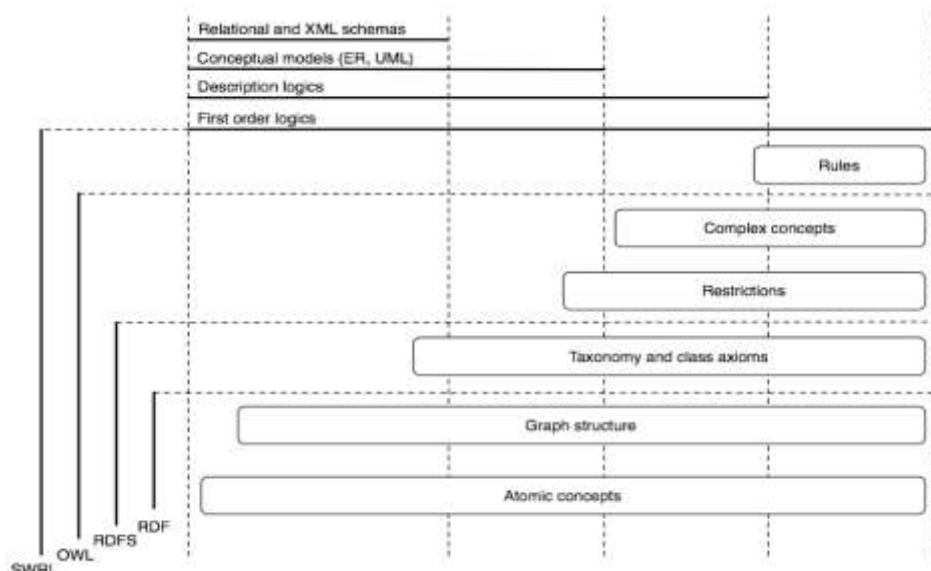


FIG. 4.6 – Représentation graphique de la complexité sémantique (Castano, & al., 2007)

- *Les concepts atomiques.* Au premier niveau, les ontologies décrivent les concepts et les entités comme ils sont, sans considérer d'autres caractéristiques telles que les attributs ou les relations. Un concept à ce niveau est dénoté seulement par un nom qui est une étiquette ou un identificateur. Le nom du concept est habituellement un terme du langage naturel qui peut être un seul terme ou un terme composé. De plus, un nom peut aussi être représenté par un acronyme ou, plus généralement, par une chaîne de caractères. À ce niveau, les concepts sont considérés similaires si leurs noms sont similaires. Beaucoup de techniques ont été proposées pour évaluer la similarité des chaînes au niveau syntaxique (ex., la distance d'édition) et au niveau sémantique (ex., en utilisant WordNet ou un thesaurus). Tous les types des langages d'ontologies, du relationnel et XML schémas aux logiques de premier ordre, fournissent un ensemble de symboles qui peuvent être adoptés pour représenter des noms de concepts.

- *Les structures de graphes.* Au deuxième niveau, les ontologies sont vues comme des structures de graphes où les concepts sont les nœuds du graphe et les relations sont les arcs du graphe. Dans cette vue, les concepts sont caractérisés par leur structure qui est déterminée par les arcs les reliant avec d'autres concepts ou avec les types de données simples. Les arcs sont des relations génériques entre les concepts et peuvent être représentés par des attributs ou des propriétés. Donc, la similarité entre deux concepts est donnée par la similarité entre leurs structures. Une façon générale de mesurer ce type de similarité est de déterminer combien d'arcs sont similaires dans les deux structures de concepts.

- *Les classes d'axiomes et taxonomie.* À ce niveau, les ontologies sont vues comme une taxonomie de concepts. Chaque concept est représenté par une classe d'objets (appelés instances de classe) et c'est possible de définir les axiomes de classes. Un axiome de classe exprime une relation sémantique entre des classes. Les relations typiques qui sont exprimées à ce niveau sont i) l'équivalence qui affirme que deux classes sont composées par les mêmes instances, et ii) la relation IS-A, appelée aussi relation de sous-classe, qui déclare qu'une classe donnée est une spécialisation d'une autre classe, c'est à dire, elle contient un sous-ensemble des instances de l'autre classe. À ce niveau, la similarité entre les concepts est déterminée en analysant les taxonomies des classes et des axiomes. Si les concepts sont les mêmes, ils auront probablement les mêmes super-concepts, et si les super-concepts sont les mêmes, les concepts sont semblables. Les sous-concepts de deux classes semblables seront aussi semblables. Si les sous-concepts sont les mêmes, les concepts comparés sont semblables. Aussi, si les concepts ont des frères et sœurs semblables, ils sont aussi

semblables. Ces relations sémantiques ne sont supportées ni par le modèle relationnel ni par XML. Les modèles conceptuels, tel que le modèle Entité/Relation et UML, supportent la définition des relations IS-A. Concernant les langages du web sémantiques, seul RDF ne supporte pas la définition de classes.

- *Les restrictions.* À ce niveau, les descriptions des ontologies peuvent être enrichies par des restrictions sur les propriétés. Une restriction dénote une contrainte qui est appliquée à une propriété quand cette propriété est appliquée à une classe. Fondamentalement il y a deux types de restrictions : les restrictions de cardinalité et les restrictions du co-domaine. Les restrictions de cardinalité dénotent le nombre minimal et maximal des valeurs pour une propriété, à la fois pour les valeurs des types de données et les valeurs des instances. Les restrictions du co-domaine dénotent les classes des instances qui sont considérés comme des valeurs valides pour une propriété. Ce type de restrictions peut déclarer que la contrainte est valide pour tous les événements de la propriété ou qu'une propriété existe avec cette contrainte. La similarité entre les classes à ce niveau est déterminée par l'analyse des contraintes exprimées par les restrictions. L'idée est que les classes avec des propriétés associées aux mêmes contraintes ou des contraintes compatibles sont considérées semblables. Une caractéristique importante est qu'une restriction décrit implicitement la classe des instances qui posent des contraintes. Pour cette raison, il est aussi important de mesurer la similarité entre deux restrictions quand elles sont considérées comme des classes. Les restrictions sont typiques aux logiques de description. Les modèles conceptuels supportent partiellement ces caractéristiques, en particulier pour les contraintes de cardinalité. Dans les langages du web sémantiques, RDFS supporte seulement la définition d'un domaine et d'un co-domaine pour les propriétés, mais ces restrictions ne sont pas applicables à une classe spécifique. Un support complet pour les restrictions est seulement disponible avec OWL.

- *Les concepts complexes.* Les concepts complexes sont dénotés par des expressions qui impliquent des concepts atomiques, tel que l'intersection, l'union ou le complément. Dans le processus de matching, la similarité entre ces concepts dépend de la similarité entre les concepts atomiques impliqués dans l'expression. L'idée est de commencer de la similarité du concept atomique et d'appliquer des règles de similarité spécifique pour les différents types d'expressions pour dériver leur similarité. Les concepts complexes sont supportés par les logiques de description et par les logiques du premier ordre.

- *Les règles.* Une règle est une expression du langage qui, étant donné un ensemble de déclarations, autorise d'inférer de nouvelles connaissances dans l'ontologie. Les règles peuvent devenir aussi intéressantes pour les considérations de la similarité. Si des règles semblables entre les concepts existent, ces concepts seront considérés comme semblables. Malheureusement, il n'y a pas eu de recherche suffisante et de support pratique concernant la couche des règles pour les considérations de la similarité. Les règles sont seulement disponibles au niveau de la logique du premier ordre.

4.4.2 Techniques d'appariement d'ontologies

Dans la littérature, plusieurs techniques de base permettant de mesurer la similarité entre les entités des ontologies ont été développées (Rahm et Bernstein, 2001; Kalfoglou et Schorlemmer, 2003; Wache, & al., 2001; Euzenat, & al., 2004; Castano, & al., 2007; Ehrig, 2007). Ces techniques sont appliquées sur un aspect particulier des entités, tel que le nom, les attributs et les relations avec d'autres entités (Bach, 2006; Euzenat et Shvaiko, 2007). Ces travaux abordent le problème de l'appariement à partir de plusieurs domaines tels que les systèmes d'information, les bases de données et le Web sémantique. L'objet de cette section est de présenter les classifications relatives au Web sémantique donc conformément aux langages de représentation des ontologies qui n'ont pas tous la même puissance d'expression. Par exemple, contrairement au langage OWL, le langage RDF(S) ne permet pas d'exprimer la cardinalité d'une relation. La question est donc de savoir comment guider le choix d'une technique pour un aspect donné.

Afin de répondre à cette question, cette section propose une classification des techniques de base qui sert à guider l'utilisateur dans le choix des techniques appropriées pour comparer l'ensemble des caractéristiques supporté par le langage de représentation des ontologies utilisées. Nous sommes parti de trois (3) travaux de la littérature sur les classifications des techniques de base d'appariement (Euzenat & Shvaiko, 2007 ; Castano, & al., 2007 ; Ehrig, 2007).

4.4.2.1 Etude de classifications existantes

A/ La classification de (Euzenat & Shvaiko, 2007)

Elles se basent principalement sur celle de (Rahm & Bernstein, 2001) qui est complétée par l'introduction de nouvelles classes. Quatre directives sont proposées :

- *L'exhaustivité.* L'extension des classes appartenant à une catégorie particulière doit couvrir ses extensions, c.à.d. leur agrégation doit donner l'extension complète de la catégorie en question.
- *La disjointure.* Les sous-catégories composant une catégorie doivent être disjointes.
- *L'homogénéité.* Les critères utilisés pour diviser les catégories sont de même nature.
- *La saturation.* La prise en compte de nouvelles techniques ne doit pas exiger la définition de nouvelles classes.

Afin d'atteindre la couche des techniques de base, cette classification peut être lue aussi bien d'une façon descendante qu'ascendante. La lecture descendante considère la manière avec laquelle les techniques interprètent l'information en entrée. Tandis que la lecture ascendante est concernée par les types d'objets manipulés par les techniques de base.

Lecture descendante : la couche supérieure est divisée en deux niveaux. Le premier niveau concerne la granularité avec laquelle les techniques d'appariement considèrent les entités. On trouve les techniques basées sur les éléments, qui considèrent chaque entité isolément, en ignorant ses relations avec d'autres entités et ceux basés sur les structures, qui calculent les correspondances en analysant comment les entités apparaissent ensemble dans une structure. Le deuxième niveau est lié à l'interprétation des entrées. Trois (3) types d'interprétation des techniques de base sont relevés, l'interprétation syntaxique, sémantique, ou celle utilisant des ressources externes. Dans le cas syntaxique, les entrées sont interprétées uniquement en fonction de leur structure syntaxique. Les ressources externes, comme les thésaurus de connaissances communes ou de domaines spécifiques sont exploitées par les techniques de base pour obtenir la signification des termes utilisés dans les ontologies. . La caractéristique des techniques sémantiques est qu'elles emploient de la sémantique formelle, telle que l'utilisation de la logique propositionnelle, pour interpréter les entrées.

Lecture ascendante : la couche inférieure est divisée en deux niveaux. Le premier niveau est classé selon le type de données sur lequel les techniques opèrent. C'est soit des chaînes de caractères (terminologiques), des structures (structurelles), des instances (extensionnelles) ou des modèles (sémantiques). Dans le deuxième niveau, on retrouve les techniques terminologiques qui sont décomposées selon la manière avec laquelle elles interprètent les mots, à savoir comme des chaînes de caractères ou comme des objets linguistiques. On y retrouve également les techniques structurelles qui sont divisées en deux types : Celles

considérant la structure interne des entités (par exemple, les attributs et leurs types) et celles prenant en compte la relation de l'entité avec d'autres entités.

Discussion : cette classification couvre toutes les techniques de base existantes dans la littérature. Elle satisfait parfaitement la directive de saturation. Cependant, ceci risque d'être très prochainement rompu puisque le développement de techniques pour appairer des règles ne peut trouver place dans cette classification et exigera donc l'introduction d'une nouvelle classe de techniques qui s'adaptent à cette catégorie. De plus, dans la couche des techniques de base, les auteurs distinguent les techniques utilisant des ressources externes de celles utilisant des ontologies de support. Toutefois, ces deux techniques possèdent exactement le même principe de fonctionnement, liées par la relation de subsumption. Elles peuvent donc être regroupées dans une seule classe qui est celle des techniques basées sur la structure d'une ressource externe. De ce fait, la directive de disjointure de classes est rompue dans cette classification.

B/ Classification de (Castano, & al., 2007)

Cette classification comprend deux catégories principales, à savoir: les techniques linguistiques et les techniques contextuelles :

- *Les techniques linguistiques* sont adoptées pour évaluer la similarité entre les noms et les étiquettes des concepts ainsi qu'entre les contextes de concepts en termes d'attributs et de relations sémantiques. Elles sont divisées en trois classes. Celles basées sur l'idée que les éléments ontologiques similaires sont dénotés par des noms ou des étiquettes similaires. On parlera alors de techniques syntaxiques. D'un point de vue syntaxique, les noms et les étiquettes des éléments ontologiques sont considérés comme des chaînes de caractères. Lorsque les chaînes de caractères sont utilisées pour nommer et étiqueter les éléments des ontologies, comme des mots du langage naturel, les techniques lexicales sont considérées. Elles adoptent les techniques de traitement du langage naturel pour analyser ces chaînes. En dernier, lorsque la signification des termes est considérée, ce sont les techniques sémantiques qui sont utilisées. L'idée est d'appairer les termes aux entrées d'une ressource externe puis de dériver leurs relations terminologiques.

- *Les techniques contextuelles* évaluent la similarité entre les concepts en les analysant en termes d'attributs et en termes de relations sémantiques avec d'autres concepts. L'approche générale consiste à calculer la similarité entre les éléments des deux contextes pour évaluer ensuite le nombre

d'éléments qui sont semblables. Trois catégories de techniques pour l'évaluation de la similarité contextuelle existent. Celles qui évaluent la similarité au moyen d'analyse de la représentation graphique ou taxonomique du contexte, qu'on désignera par techniques basées sur une métrique heuristique. Lorsque dans l'évaluation, on calcule la probabilité que deux concepts aient les mêmes instances, on parlera alors de techniques basées sur le raisonnement probabiliste. En dernier, lorsque la sémantique formelle, associée aux éléments du contexte d'un concept qui est vu comme un ensemble d'attributs logiques, est exploitée, ce sont les techniques basées sur le raisonnement automatique qui sont considérées.

- *Discussion* : une première critique peut être faite sur la dis jointure non vérifiée entre les définitions des deux classes principales de cette classification. En effet, les techniques linguistiques prennent en compte les contextes de concepts en termes d'attributs et de relations sémantiques, ce qui est également couvert par les techniques contextuelles. De plus, cette classification ne couvre pas toutes les techniques de base existantes. Par exemple, pour déduire la similarité entre les instances, elle ne considère que les techniques basées sur un raisonnement probabiliste malgré l'existence d'autres approches telles que celles basées sur les techniques statistiques (Cox et Cox, 1994).

C/ Classification de (Ehrig, 2007)

Une classification en trois couches est adoptée dans ce cas: Données, Ontologie, et Contexte. À travers toutes les couches, l'auteur désigne un champ orthogonal supplémentaire qui représente la connaissance du domaine qui peut être située à tous les niveaux de la complexité ontologique.

- *La couche des données* : C'est la première couche dont laquelle la comparaison des entités se fait en considérant seulement les valeurs des données de types simples ou complexes comme les nombres entiers et les chaînes de caractères. Pour comparer des valeurs de donnée, une fonction générique de similarité telle que la distance d'édition pour les chaînes de caractères est utilisées. Pour les nombres entiers, il détermine simplement une distance qui leur est relative. Les types de données complexes construits à partir des types de données simples exigeraient aussi des mesures plus complexes qui sont basées complètement sur les mesures simples.

- *La couche Ontologie* : D'abord, il est à noter que les mesures de similarité de la couche Ontologie peuvent inclure les mesures de similarité de la couche Données. De plus, quatre (4)

niveaux sont considérés selon la complexité sémantique citée précédemment (voir le paragraphe 4.4.1), les réseaux sémantiques, les logiques de descriptions, les restrictions et les règles. Dans le niveau le plus bas, les réseaux sémantiques, les ontologies sont vues comme des graphes avec des concepts et relations. Cependant, aucune technique permettant d'exploiter ce niveau n'est spécifiée. Le deuxième niveau, les logique de description, la structure taxonomique, basée sur la relation « IS-A », est considérée. Les techniques utilisées, dans ce cas, sont celles basées sur le nombre d'arcs séparant deux concepts. Pour les restrictions, l'auteur ne spécifie aucune technique qui peut les exploiter. Dans le cas des règles, qui selon l'auteur peuvent devenir intéressantes, il n'y a pas eu de recherche suffisante dans cette direction.

- *La couche Contexte* : Cette couche est concernée par l'usage concret des entités dans le contexte d'une application donnée. De ce fait, l'appariement est effectué en comparant les usages des entités dans les applications basées sur les ontologies. Le principe est que les entités similaires sont souvent utilisées dans un contexte similaire. Par exemple, dans le portail amazon.com, étant donnée une information sur quelles personnes achètent quels livres, on peut décider si deux livres sont semblables ou pas dans un contexte donné. L'auteur utilise les deux directions de l'implication dans la découverte de la similarité : si deux entités sont utilisées dans le même contexte alors elles sont similaires et vice versa : si les mêmes entités sont utilisées dans deux contextes alors ces derniers sont similaires.

- *Discussion* : Bien que la classification décrite dans cette section suit les niveaux de complexité sémantique, elle est très générale et ne permet pas d'avoir une classification détaillée de chaque niveau.

4.4.2.2 Principe de la classification proposée

L'objectif de la classification proposée est à la fois de guider le choix, dans le cadre du Web sémantique, de la technique appropriée selon le langage ontologique utilisé et de pallier aux insuffisances constatées dans les différentes classifications existantes (Oulefki & Akli-Astouati, 2008 b).

Pour cela, les techniques de base sont regroupées dans un premier lieu en trois couches, où chaque couche présente les techniques pouvant être utilisées pour appairer les niveaux de complexité sémantique supportés par un langage ontologique donné. De ce fait, puisqu'il

existe trois langages ontologiques recommandés dans le Web sémantique, la classification proposée est présentée en trois couches (voir figure 4.7).

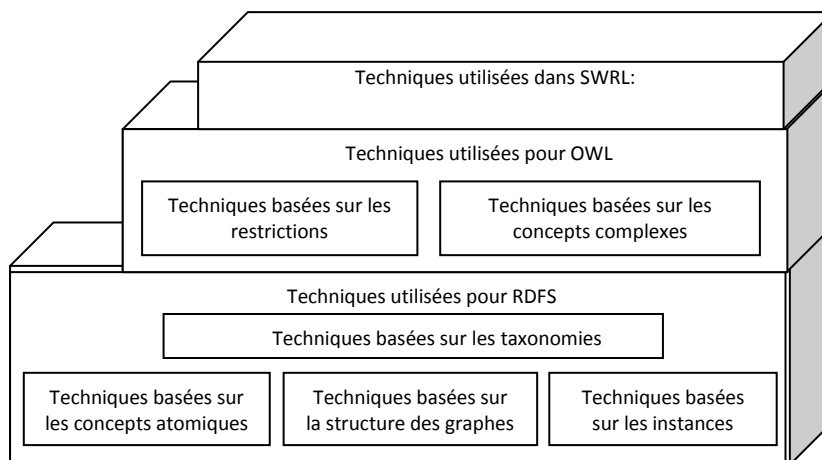


FIG. 4.7 – Modèle en couche proposé pour classifier les techniques d'appariement des ontologies

A l'intérieur de chaque couche, les techniques de base sont classées selon le niveau de complexité sémantique qu'elles considèrent. Pour chaque niveau de complexité sémantique, une classification plus détaillée est également proposée.

La classification ainsi proposée assure les quatre directives de la classification de (Euzenat, & al., 2004), l'homogénéité, la saturation, l'exhaustivité et la disjointure.

Dans ce qui suit, nous examinons pour chaque couche du modèle proposé, les techniques de base qui lui correspondent en présentant également, pour chaque niveau de complexité sémantique, une classification des techniques de base qui l'exploitent. Une présentation plus détaillée peut être trouvée dans (Oulefki, 2008).

A/ Techniques utilisées pour appairer des ontologies écrites en RDF

Les techniques utilisées pour appairer les caractéristiques pouvant être exprimées par des primitives RDF(S) sont celles exploitant les concepts atomiques, les structures de graphes, les instances et les structures taxonomiques.

A.1/ Concept Atomique

Le nom et l'étiquette sont habituellement représentés par des chaînes de caractères courtes, tandis que les commentaires sont représentés par des chaînes de caractères longues. Les mesures de similarité proposées sont différemment conçues pour chaque catégorie :

- Les chaînes de caractères courtes sont généralement comparées en utilisant des techniques syntaxiques et/ou basées sur la structure d'une ressource externes.

- Les techniques syntaxiques considèrent deux entités similaires si leurs noms ou leurs étiquettes sont similaires syntaxiquement.

- Les techniques basées sur la structure d'une ressource externes, qui peut être une ontologie de support, opèrent en deux étapes (Safar, & al., 2007). Dans une première étape dite « d'ancrage », les termes sont appariés aux entrées d'une ressource externe. Dans la deuxième étape dite « de dérivation », les relations entre les entités sont dérivées en utilisant les relations découvertes durant l'étape d'ancrage.

- Les chaînes de caractères longues sont comparées en utilisant des techniques dites « techniques basées sur les tokens » (Euzenat, & al., 2004 ; Bach, 2006 ; Euzenat et Shvaiko, 2007). Ces dernières découpent les chaînes de caractères en plusieurs morceaux appelés « tokens ». L'intégration des techniques basées sur la structure d'une ressource externes dans le calcul de la similarité entre les chaînes de caractères longues n'est pas utile et est coûteuse en temps de calcul et donne comme résultat plus de bruit que d'informations utiles (Bach, 2006). Les commentaires et les descriptions contiennent assez d'information descriptive pour pouvoir être comparés aux autres en employant les techniques basées sur les tokens.

A.2/ Structure de graphe

Ces techniques exploitent des informations structurelles lorsque les entités en question sont reliées entre elles par des liens sémantiques ou syntaxiques, formant ainsi un graphe des entités. Il existe deux types de relations les relations méréologiques et toutes les relations impliquées (Euzenat, & al., 2004 ; Euzenat et Shvaiko, 2007):

- *Structures méréologique* : L'exploitation de la structure méréologique présente la difficulté de trouver les propriétés qui portent cette structure. Cependant, s'il y a lieu de détecter les relations qui supportent la structure part-of, il sera possible de calculer la similarité entre les concepts en considérant qu'ils seront plus similaires s'ils partagent des parties similaires.

- *Les relations*: Les concepts sont aussi liés par la définition de leurs propriétés. Ces propriétés correspondent aux arcs du graphe représentant une ontologie. Par conséquent, considérer ces propriétés pour déduire la similarité entre les entités peut être intéressant. L'idée est que si l'on a deux entités similaires A et A', et si elles sont connectées par un même type de relation R avec deux autres entités B et B', alors on peut déduire que B et B' sont

d'une façon ou d'une autre similaires. De même, si on sait que A et A' sont similaires et que B et B' sont aussi similaires, alors les relations de A-B et de A'-B' peuvent être similaires.

A.3/ Techniques utilisées pour les instances

La similarité entre deux entités, qui sont notamment des concepts, est décidée par les méthodes extensionnelles en analysant leurs extensions, c.-à-d., l'ensemble de leurs instances. Généralement, les méthodes extensionnelles sont divisées en deux catégories (Euzenat et Shvaiko, 2007) : celles qui s'appliquent aux ontologies avec des ensembles d'instances communes, et celles qui travaillent sur des ensembles d'instances hétérogènes.

B/ Techniques utilisées pour appairer des ontologies écrites en RDFS

En plus des techniques décrites précédemment, le processus d'appariement des ontologies écrites en RDFS peut utiliser les techniques basées sur les taxonomies. Ces dernières exploitent la relation IS-A entre les concepts (Castano, & al., 2007). Si les concepts sont les mêmes, ils auront probablement les mêmes super-concepts, et si les super-concepts sont les mêmes, les concepts sont alors semblables. Les sous-concepts de deux classes semblables seront aussi semblables. Si les sous-concepts sont les mêmes, les concepts comparés sont semblables. Aussi, si les concepts ont des frères et sœurs semblables, ils sont aussi semblables.

C/ Techniques utilisées dans OWL

On retrouve deux niveaux associés à OWL. Le premier est consacré aux restrictions. Tandis que le deuxième niveau soulève l'aspect de concepts complexes.

C.1/ Techniques utilisées au niveau des restrictions

Ces techniques sont aussi référencées dans (Bach, 2006 ; Euzenat, & al. 2004) sous l'appellation « *méthodes basées sur la structure interne* » et dans (Rahm et Bernstein, 2001) comme « *méthodes basées sur les contraintes* ». Afin de mesurer la similarité entre les entités, ces méthodes utilisent les informations concernant les propriétés des entités telles que le co-domaine, la cardinalité ou la multiplicité. L'idée est que les classes avec des propriétés associées aux mêmes contraintes ou des contraintes compatibles sont considérées semblables (Euzenat, & al. 2004).

C.2/ Techniques utilisées au niveau des concepts complexes

Afin de déduire la similarité entre deux entités, les méthodes utilisées dans ce niveau se basent sur des modèles de logique tels que la satisfiabilité propositionnelle (SAT), la SAT modale ou les logiques de description (Bach, 2006 ; Euzenat, & al. 2004).

D/ Techniques utilisées pour appairer des ontologies écrites en SWRL

Les règles peuvent devenir aussi intéressantes pour les considérations de la similarité. Si des règles semblables entre les concepts existent, ces concepts seront considérés comme semblables (Castano, & al., 2007). Malheureusement, il n'y a pas eu de recherche suffisante et de support pratique concernant la couche des règles pour les considérations de la similarité.

4.4.3 Processus d'appariement

Dans la littérature plusieurs explicitations du processus d'appariement d'ontologies ont été proposées (Castano, & al., 2007; Ehrig, 2007; Euzenat & al., 2007a). Ils diffèrent souvent dans leurs degrés de granularité et de détails.

Dans (Euzenat, & al., 2007a) par exemple, les auteurs définissent le processus d'appariement comme «une fonction f qui prend en entrée une paire d'ontologies o et o' , un alignement en entrée A , un ensemble de paramètres p et un ensemble de ressources r et retourne en sortie un alignement A' entre ces ontologies : $A' = f(o, o', A, p, r)$ ». Cela suppose le processus comme une boîte noire où aucune spécification des étapes devant être suivies n'est indiquée.

Une explicitation plus détaillée est présentée dans (Castano, & al., 2007), où trois (3) étapes sont énumérées. On commence par l'acquisition des ontologies devant être appariées pour le représenter dans un modèle interne. Ensuite, l'étape d'exécution des techniques d'appariement de base est déclenchée avec répétition pour raffiner les résultats obtenus dans les exécutions antérieures. Enfin, selon les résultats des techniques d'appariement de base exécutées précédemment, pour générer les correspondances entre les éléments des ontologies, deux tâches sont citées :

- *Agrégation des valeurs de similarité* : Dans le cas des techniques d'appariement de base produisant une valeur de similarité entre les éléments des ontologies, les différentes valeurs de similarité individuelles sont combinées dans une valeur de similarité représentative. Les résultats qui ne sont pas considérés pertinents sont supprimés.

- *Raisonnement et vérification de la consistance* : Dans le cas des techniques d'appariement de base qui produisent une relation sémantique entre les éléments des ontologies, une tâche de raisonnement est exécutée afin d'inférer de nouvelles relations et de vérifier la consistance des correspondances.

Ce qu'on peut dire de ce processus, c'est qu'il les étapes le définissant sont bien étoffées par moment et le sont moins par d'autres. Par exemple, l'étape de sélection des entités sur lesquelles vont être appliquées les techniques de base n'est pas indiquée. De plus, ce processus anticipe l'étape de vérification de la consistance en la considérant propre aux approches fournissant des relations sémantiques. Cependant, il existe des outils, tels que Prompt (Noy et Musen, 2000), qui utilisent la vérification de consistance après avoir produit des valeurs de similarité entre les entités comparées.

Dans le troisième processus (Ehrig, 2007), cinq (5) étapes sont énumérées :

- *Ingénierie des caractéristiques* : afin de comparer deux entités de deux ontologies O_1 et O_2 , un sous ensemble spécifique de leurs caractéristiques est extrait. Ces extraits ne sont pas des constructions arbitraires, mais possèdent plutôt une signification spécifique dans l'ontologie. Par exemple, un algorithme d'appariement peut utiliser seulement un sous-ensemble de primitives OWL tel que la taxonomie, ou les descriptions linguistiques (par exemple, l'étiquette « Voiture » pour décrire le concept « Voiture »).

- *Sélection de l'étape de recherche* : avant de procéder à la comparaison des entités, il est nécessaire de choisir les paires d'entités à considérer. Cette étape a pour objectif de spécifier les paires d'entités candidate.

- *Calcul de la similarité* : le calcul de la similarité entre deux entités e et f de deux ontologies différentes fournit des valeurs de similarité individuelles.

- *Agrégation des valeurs de similarité* : dans cette étape, les différentes valeurs de similarité des différentes caractéristiques d'une paire d'entités candidate sont agrégées afin d'avoir une seule valeur de similarité représentative de la paire d'entités en question.

- *Interprétation* : l'interprétation utilise les valeurs de similarité représentatives ou individuelles pour dériver un alignement entre les entités. Le mécanisme le plus commun est celui des seuils (Do et Rahm, 2002; Noy et Musen, 2001).

- *Itération* : puisque la similarité d'une paire d'entité influence la similarité des paires d'entités voisines, des itérations sur l'ensemble du processus sont exécutées. Dans chaque itération, les similarités d'un alignement candidat sont recalculées en se basant sur les

similarités des paires d'entités voisines. L'itération termine lorsqu'aucune nouvelle correspondance n'est proposée ou après un nombre fixe d'itération.

Ce processus est au même niveau de détails que le processus 2. D'autre part, il ne fait pas la distinction entre les approches utilisant un raisonnement logique et celles qui opèrent sur des valeurs de similarité. Aussi, on ne voit à aucun moment au niveau de quelle étape l'intervention humaine est située. Cette remarque est très importante puisque, à l'heure actuelle, la plupart des approches d'appariement sont semi-automatique. L'implication humaine dans les différentes étapes du processus est donc de rigueur.

Suite à cette étude et afin d'apporter une solution aux problèmes cités ci-dessus, nous proposons dans le chapitre suivant un processus d'appariement d'ontologies permettant de fournir le maximum d'ingrédients pouvant composer un système d'appariement.

4.5 Applications de l'Alignement des ontologies

L'appariement des ontologies est supposé non seulement être une théorie de haut niveau, mais aussi destiné à des scénarios pratiques (Ehrig, 2007 ; Davidovsky, & al., 2012). Pour cette raison, cette section examine, à titre non exhaustif, quelques cas d'utilisation concrets d'appariement des ontologies avec une tendance à les inclure dans des travaux sur les Systèmes Multi-Agents (SMA).

4.5.1 Recherche d'information sémantique

Les travaux actuels en recherche d'information (RI) s'orientent vers les technologies issues du web sémantique comme l'usage des ressources sémantiques (ontologies, thésaurus, bases de données lexicales) et celles issues du traitement automatique des langues. Une thématique qui prend de l'ampleur, dans ce sens, est l'alignement des ontologies. En effet, si on se penche sur la conférence EGC'2012 (Extraction et Gestion des Connaissances) par exemple, on remarquera qu'un atelier a été spécialement dédié à la Recherche d'Information Sémantique. Sur les quatre contributions, deux concernent l'appariement. Un autre exemple peut être donné, c'est le projet SEWASIE (Dongilli, & al., 2005) qui est fondé sur la technologie agent pour l'interrogation de sources de données hétérogènes en utilisant des ontologies. Les agents intelligents sont utilisés pour extraire des informations ou des connaissances satisfaisant la sémantique et le contexte d'une requête utilisateur. Les alignements sont nécessaires pour la

structure et la sémantique de la requête en corrélation avec les schémas de ressources d'information et de métadonnées.

4.5.2 L'intégration de données/Schémas

L'alignement des ontologies est utilisé dans ces applications pour l'intégration de données stockées dans des sources hétérogènes différentes. L'intégration de schémas de bases de données peut également être considérée dans cette catégorie. Une architecture multi-agent pour l'intégration de données basée sur des ontologies est présentée dans (Medcraft, & al., 2003). Une étude détaillée du problème et des solutions qui s'y apportent est élaborée dans (Nagy et Vargas-Vera, 2010). Pour ce qui est du développement de nouveaux systèmes d'informations, la réalisation de l'interopérabilité entre des bases de données existantes est devenue une préoccupation des plus importantes. Pour ce faire, une solution radicale serait de créer une nouvelle base de données, reprenant toutes les informations provenant des différentes bases de données disponibles. C'est ce que l'on appelle l'intégration des bases de données (Samyn, 2002).

Une autre solution, qui évite de réécrire toutes les applications utilisant ces bases de données, est de créer une base de données virtuelle. Cette base de données n'est qu'une vue, permettant à de nouvelles applications d'accéder à l'ensemble des données disponibles.

L'accès à ces données se fait via un logiciel d'intégration qui se charge d'aller récupérer les informations dans les différentes bases de données disponibles. Cette solution est appelée fédération de bases de données.

D'un point de vue conceptuel, ces deux solutions sont identiques (Samyn, 2002). Elles nécessitent toutes les deux la création d'un schéma conceptuel réunissant les points communs des bases de données à unifier, et la traduction de ce schéma vers d'autres niveaux d'abstraction. Le processus permettant de créer le schéma conceptuel doit comporter trois étapes (Parent et Spaccapietra, 1996). Une *pré-intégration*, étape dans laquelle les schémas en entrée sont transformés de différentes manières pour les rendre plus homogènes (sur les plans sémantique et syntaxique) ; La *recherche des correspondances*, qui consiste à apparier les schémas initiaux afin d'identifier les éléments semblables et décrire les liens inter-schémas ; L'*intégration*, qui unifie les éléments en correspondance en un schéma intégré et produit les règles de traduction associées entre le schéma intégré et les schémas initiaux.

4.5.3 Dialogue Homme/Machine

L'alignement des ontologies peut-être utilisés dans les interactions et les dialogues Homme/Machine pour assurer une compréhension entre un utilisateur et un agent par exemple. En effet, (Atencia et Schorlemmer, 2008) soutiennent que l'alignement sémantique dans la plupart des cas est fortement lié à une interaction particulière entre des agents. Une approche alternative est mise en avant, c'est la théorie de l'argumentation (Dung, 1995) qui est largement adoptée en Intelligence artificielle (Rahwan et Simari, 2009) et dans les applications SMA (Maudet, & al., 2007).

L'argumentation est utilisée pour l'alignement d'ontologies par des agents dans un but de déterminer les correspondances acceptables dans les négociations. Une solution basée sur l'argumentation pour la négociation entre agents, utilisant différentes ontologies, est prise en compte dans (Euzenat, & al., 2006).

4.5.4 Evolution des ontologies/versionning

Quand une ontologie est développée par plusieurs ontologistes d'une manière collaborative et qu'elle devient de plus en plus grande et complexe au niveau de la structure et de la représentation, il est difficile alors de comprendre toutes les parties de l'ontologie (Luong, 2007). Il est aussi difficile de connaître les parties affectées lorsque des changements sont effectués sur elle. Par conséquent, le besoin d'un mécanisme qui surveille les changements ontologiques, i.e., un mécanisme qui permet de gérer et de maintenir les différentes versions de l'ontologie est nécessaire. Cette opération est assurée à travers l'appariement dont le rôle est de découvrir les différences entre deux versions d'une ontologie. On y retrouve les entités ontologiques qui ont été ajoutées, supprimées ou renommées.

Dans ce scénario, il est utile de: (i) trouver les correspondances entre l'ancienne version O_t et la nouvelle version O_{t+n} de l'ontologie, (ii) de générer une transformation par l'utilisation de ces correspondances et (iii) de transformer les instances de données sous-jacentes I_t vers I_{t+n} .

Un certains nombres de travaux traitant de l'évolution d'une ontologie ont été recensé dans la littérature (Packer, & al., 2009 ; Davidovsky, & al., 2010 ; Li et Yang, 2008).

4.6 Conclusion

Dans le but de résoudre les problèmes de l'hétérogénéité des ontologies, nous avons présenté des travaux dans la littérature qui abordent l'appariement des ontologies, solution à cette problématique. Toutes les notions de base qui s'y rattachent, ainsi que quelques cas d'utilisation sont introduites. Nous avons remarqué que la technologie agent est très présente dans cette thématique et qu'une tendance est aux travaux sur l'argumentation.

Nous avons étudié et critiqué certaines classifications et processus de la littérature. Pour la classification, une solution a été apportée (Oulefki et Akli, 2008) en partant des niveaux de complexité sémantique associés aux différents langages du web sémantique. C'est une classification naturelle, qui fournit un guide d'utilisation selon le langage ontologique utilisé. En ce qui concerne le processus, la critique est suivie de la conception et la réalisation d'un système d'appariement basé sur la technologie agent que l'on détaillera dans le chapitre suivant. L'alignement accepté sera conditionné par la construction d'un argument justifiant ce choix.

Chapitre 5

Agents coopératifs pour l'Alignement d'Ontologies

Notre idée quotidienne, conventionnelle, de la réalité est une illusion que nous passons une partie substantielle de notre vie à étayer, fut-ce au risque considérable de plier les faits à notre propre définition du réel, au lieu d'adopter la démarche inverse. De toutes les illusions, la plus périlleuse consiste à penser qu'il n'existe qu'une seule réalité. En fait, ce qui existe, ce ne sont que différentes versions de celle-ci dont certaines peuvent être contradictoires, et qui sont toutes des effets de la communication, non le reflet de vérités objectives et éternelles.

Paul Watzlawick (La réalité de la réalité. Confusion, désinformation, communication)

5.1 Introduction

Plusieurs approches d'alignement d'ontologies existent dans la littérature (Noy et Musen, 2000, 2002 ; Do, & al., 2002 ; Bach, 2006). Elles sont basées sur différentes mesures de similarité. La plupart d'entre elles sont subsumées par un processus général d'appariement décrit dans (Castano, & al., 2007) et est divisé en trois étapes :

– L'étape d'acquisition des ontologies devant être appariées. Le problème ici est de cohabiter avec les différentes représentations et modèles d'ontologies.

– L'étape de l'exécution des procédures d'appariement. Très souvent, ces procédures sont basées sur l'application des mesures de similarité sur les entités des différentes ontologies. Cette étape retourne alors un ensemble de valeurs de similarité entre les paires d'entités différentes des ontologies. En général, il existe plusieurs valeurs de similarité pour la même

paire d'entités, ex., une première valeur de similarité pour les noms, une deuxième pour les commentaires, et une troisième pour les relations avec les autres entités, etc.

– Au final, pour chaque paire d'entités, les différentes valeurs de similarité sont combinées afin d'obtenir une seule valeur. Les valeurs de similarité obtenues doivent être ensuite filtrées, par exemple en utilisant un seuil particulier, afin d'extraire un ensemble pertinent de correspondances appelé alignement.

Dans ces approches, plusieurs problèmes apparaissent. Le premier est celui de la vision centralisée qui va rendre les programmes longs et complexes. De plus, le calcul d'une seule valeur de similarité finale qui consiste en la moyenne des différentes valeurs calculées pour la même correspondance peut induire en erreur. Prenons l'exemple de deux concepts "Animal" et "Beast" appartenant respectivement aux ontologies O_1 et O_2 de la figure 5.1. Syntaxiquement, les deux chaînes de caractères sont différentes. Donc, la valeur de similarité approche de très près le 0. Du point de vue structurel, les deux concepts ne possèdent pas le même voisinage, ce qui fait que la valeur aussi est égale à 0. Par contre, sémantiquement, les deux concepts ont le même synonyme donc la valeur associée est égale à 1. Le calcul de la moyenne donne une valeur très inférieure à 0.5 ce qui ne permet pas de considérer les deux concepts comme équivalents malgré le fait qu'ils le sont.

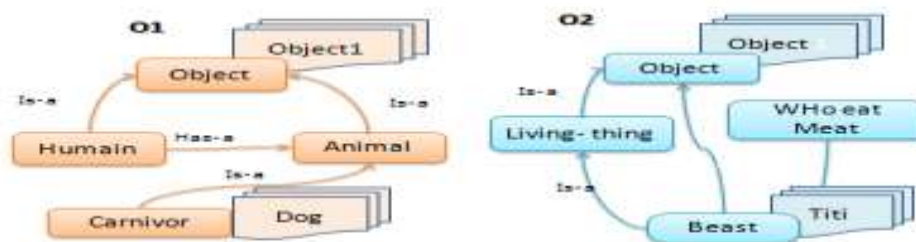


FIG. 5.1. – Exemple de deux ontologies de même domaine

Les travaux qui abordent le problème de la négociation entre agent autour de l'alignement des ontologies existent (Bailin et Truszkowski, 2002 ; Silva, & al., 2005 ; Van Digglen, & al., 2006). Dans (Bailin et Truszkowski, 2002) par exemple, un protocole de négociation pour l'alignement des ontologies qui permet aux agents d'échanger des parties de leurs ontologies

par un processus itératif d'interprétation, classification et explication est proposé. Chaque agent ne doit conserver à la fin qu'une seule ontologie partagée qui consiste en l'union de tous les termes et les relations. Donc, les ontologies initiales disparaissent ce qui n'est pas toujours convenable.

La solution de (Silva, & al., 2005) dont le but est d'établir un consensus entre les différents agents, se base sur les fonctions d'utilité et de méta-utilité utilisées par les agents afin de décider si une correspondance est acceptable ou pas. Le problème dans ce cas, c'est que son développement s'est fait sous la plate-forme de mapping d'ontologies distribuées MARFA (MAPPING FRamework) dont l'approche repose sur la notion fondamentale de Passerelle pour formaliser l'alignement. Dans ce cas, le système ne peut fonctionner sous un autre environnement. Une autre proposition a été faite dans (Van Diggelen, & al., 2006) où un système décentralisé pour l'établissement d'un vocabulaire de communication commun entre applications a été défini. C'est une approche qui vise un consensus dans la construction d'une ontologie commune entre des agents distribués. Les agents, dans ce cas, remplissent tous la même fonction et aucune séparation des tâches n'a été faite.

Pour pallier ces insuffisances, nous avons utilisé un système à plusieurs agents indépendamment de la plateforme de développement. Les agents alignent des ontologies en précisant les liens de correspondances entre elles sans que ces dernières ne disparaissent. Nous avons proposé, pour cela, un système d'alignement à base de 3 niveaux afin de décentraliser le travail sur plusieurs agents. Ces derniers vont coopérer pour rendre à la fin un fichier d'alignements qui comporte les correspondances trouvées par le processus. Comme actuellement, l'accent est mis sur les approches qui combinent des alignements générés par différentes techniques ou systèmes d'appariement (Trojahn, & al., 2010), en utilisant la théorie de l'argumentation (Dung, 1995) qui offre un cadre applicable à ce scénario, le choix d'acceptation d'une correspondance sera conditionné par la construction d'un argument le justifiant. Cependant, les deux cas (acceptation ou refus) sont possibles. Pour cela, un mécanisme de précision de préférences entre les correspondances a été introduit. Le fichier final contenant les correspondances jugées les plus acceptables peut être exploité par les diverses applications.

Dans ce qui suit, avant de présenter l'architecture de notre système, nommé S3A (Système multi Agents à 3 niveaux), un aperçu est donné sur les systèmes d'argumentation, où quelques

notions de base sont introduites. Pour clôturer ce chapitre, nous aborderons le processus proposé ainsi que la validation de notre outils.

5.2 La théorie de l'Argumentation

Lorsqu'un agent autonome dialogue avec les autres agents du système, il doit avoir des mécanismes lui permettant d'interagir et de raisonner pour décider de l'action qu'il doit entreprendre avec le monde qui l'entoure. Différentes approche pour construire des modèles d'interaction et de décision ont été développés (Genesereth et Nilso, 1987). Cependant, ils ne peuvent pas traiter efficacement les informations et les données incomplètes ou incohérentes. C'est pourquoi plus récemment, l'argumentation a été utilisée. Elle est définie comme suit: « *L'argumentation est une activité verbale et sociale de la raison visant à accroître (ou diminuer) l'acceptabilité d'un point de vue controversé pour l'auditeur ou le lecteur, en mettant en avant un ensemble de propositions destinées à justifier (ou réfuter) un point de vue devant un jugement rationnelle* » (Prakken 2000).

Ainsi, l'argumentation a pour but de résoudre un point de vue *controversé*, selon l'information disponible, dans le sens où, il est soumis à la *justification* ou à la *réfutation*.

L'idée derrière l'argumentation est de se doter de la possibilité de dire plus sur la certitude d'un fait particulier. En d'autres termes, au lieu de le qualifier uniquement par des valeurs vraies (1) et fausses (0), on évalue la raison pour laquelle le fait est vrai en construisant un argument en sa faveur.

Schiffirin souligne que l'argumentation est un mode de discours ni purement monologique, ni purement dialogique ... c'est un discours par lequel les locuteurs défendent des positions discutables (Schiffirin, 1987). Dans notre cas, on s'intéresse à l'argumentation dialogique dans laquelle un ensemble d'agents ou d'entités interagissent afin de construire des arguments pour ou contre une certaine conclusion. Si un agent propose un argument, celui-ci peut être confronté à des attaques de la part d'autres agents qui formulent à leur tour des contre-arguments. Des stratégies pour persuader les autres agents sont alors utilisées pour les amener à une certaine conclusion. L'argument est une raison offerte pour influencer la croyance de quelqu'un à propos de quelque chose. Son objectif est donc de montrer qu'une conclusion est valide ou vraie.

Le processus d'argumentation permet de construire des arguments en faveur des données appelées «Conclusion d'argument» dites supportées par celui-ci. Des conclusions contrariants cette dernière, peuvent être supportées par d'autres arguments. Un point très important de ce processus, est sa capacité à déterminer parmi tous les arguments ceux qui sont acceptables et qui supportent des conclusions justifiées.

5.2.1 Eléments de base d'un système d'argumentation

5.2.1.1 Le langage

Il est nécessaire d'être en possession de langage permettant de décrire les arguments ainsi que les relations entre ces derniers. Plusieurs propositions de langages pour les systèmes d'argumentation ont été faites, allant du langage propositionnel jusqu'aux langages de programmation, en passant par l'approche libérale.

5.2.1.2 L'argument

Un argument doit être défini de façon précise et sans ambiguïté. C'est une raison qui permet de justifier ou de croire en une proposition. On parle alors d'argument en faveur de la proposition. Un argument peut être écrit sous forme de (Prémisse, Conclusion) où la prémisse peut être considérée comme une preuve pour la conclusion.

5.2.1.3 La relation de contrariété entre les arguments

La relation de contrariété représente les différents conflits pouvant exister entre les arguments. Les relations de contrariétés sont de deux types essentiellement :

- *Relation Attaque* : L'attaque est une manière de nier un argument par la négation d'un des éléments ou une prémisse de celui-ci.

- *Relation Réfute* : Cette relation est utilisée pour nier la conclusion d'un argument par la présence d'un autre argument dont la conclusion est contraire de la première. La Réfutation est symétrique.

5.2.1.4 Les conditions d'acceptabilité

La valeur de la conclusion dépend du statut des arguments. Ce dernier dépend de l'interaction des différents arguments. Donc, une conclusion est supportée jusqu'à l'arrivée d'un contrariant.

- *Acceptabilité individuelle* : Attribuer un niveau d'acceptabilité à chaque argument en fonction de l'existence de contrariants directs. On regroupe les arguments ayant le même degré d'acceptabilité dans une seule classe appelée classe d'acceptabilité.

- *Acceptabilité Conjointe* : Dans le cas où un argument ne peut pas se défendre seul contre un contrariant direct, il est possible de savoir est ce que ce contrariant n'est pas à son tour contrarié par un autre argument.

Voici quelques définitions formelles d'un système d'argumentation :

Définition 1 : Un système d'argumentation est un doublé $SA = (EA, att)$ où EA est un ensemble d'arguments $\$$ et « att » est une relation binaire sur EA que nous nommons relation d'attaque.

Nous avons donc $att (EA \times EA)$. Cette relation définit les contrariétés entre arguments et formalise ainsi la notion de contre-argument.

Définition 2: Un argument A attaque un autre argument B si et seulement si on a la relation $att (A, B)$. On dit alors que A est un contre-argument de B .

Définition 3 : L'argument A est acceptable selon un ensemble d'arguments S si et seulement

Si $\forall X, \exists A \in EA, att(X, A) \Rightarrow (\exists Y \in S, att(Y, X))$.

Un ensemble d'arguments sans-conflit S est admissible si et seulement si $\forall X \in S, X$ est acceptable selon S . Un ensemble d'arguments est donc admissible si chacun de ses arguments est soutenu par le reste de l'ensemble. Nous avons ici une stabilité vis-à-vis du système d'argumentation. Enfin, nous voulons inclure le maximum d'arguments dans ce type d'ensemble pour représenter la position cohérente et défendable qui prend en compte le plus d'arguments possibles. Il n'est pas toujours facile de décider du statut d'un argument. Prenons l'exemple suivant :

Exemple :

Arg1 : ((Titi est un oiseau, les oiseaux volent), Titi vol)

Arg2 : ((Titi est un pingouin, les pingouins ne volent pas), Titi ne vole pas)

On remarque que les deux arguments se réfutent puisque leurs conclusions sont contradictoires. La question qu'on se pose est laquelle des deux conclusions considérer.

A ce niveau, il est clair qu'on doit effectuer un choix entre les deux conclusions plausibles. En d'autres termes, on peut préférer une conclusion par rapport à une autre. C'est la raison de l'introduction de préférence entre arguments.

5.2.2 Préférences entre arguments

La notion d'acceptabilité des arguments a été définie sur des considérations purement logiques, c'est-à-dire sur la base de l'existence d'autres arguments. Néanmoins, d'autres critères peuvent être pris en compte pour comparer les arguments.

Plus généralement, des relations de préférence peuvent être utilisées pour comparer les arguments. Ces relations de préférence peuvent être définies de plusieurs façons : on peut préférer un argument à un autre si le premier argument contient plus d'informations ou de prémisses que le second. Dans ce cas, on préfère l'argument qui a le plus de justifications. Une autre manière de comparer deux arguments consiste à choisir le plus récent des deux. Dans ce cas, la notion de temps intervient. Généralement, les préférences sont déterminées par des critères d'évaluation qui dépendent de l'application considérée, comme par exemple (Amgoud, 1999) :

- Degré de certitude : étant donné une mesure de probabilité, une mesure de nécessité, etc.
- Degré de fiabilité : dans le cas de la fusion de plusieurs bases de connaissances données par différentes sources d'information, les croyances de la source la plus fiable sont préférées à celles d'une source moins fiable.
- Degré d'utilité : en théorie de la décision, une fonction d'utilité permet de représenter les préférences d'un agent sur les conséquences possibles d'une décision.
- Les préférences de l'utilisateur : dans le cas du traitement de l'incohérence dans des bases de connaissances, la résolution des conflits peut être guidée par la prise en compte des préférences d'un utilisateur entre les croyances impliquées dans un conflit.
- Degré de spécificité: Il existe deux types de préférence :
 - *Préférences implicites*: On préfère plus les règles avec exception (principe de spécificité). Les cas les plus spécifiques sont les plus préférés.
 - *Préférences Explicites*: Préférences exprimées formellement. On trouve :
 - ✓ Préférence contextuelle : Ce sont celles qui dépendent du contexte, données sous forme d'un pré-ordre entre les connaissances.

- ✓ Préférence conditionnel : qui est de la forme « Si X alors A est préféré à B » la préférence dépend de l'information « X ».

Les préférences peuvent être contestables ou incontestables, suivant qu'elles soient mises en cause ou pas. Donc, un système d'argumentation basé sur les préférences est un triplet $SA_{Pref} = \langle EA, att, Pref \rangle$ où :

- EA est un ensemble d'arguments ;
- att est une relation binaire sur EA que nous nommons relation de contrariété. Elle formalise ainsi la notion de contre-arguments.
- Pref est une relation de préférence entre les arguments : pré-ordre, degré de certitude, etc. Elle formalise ainsi la notion d'argument préféré à un autre.

Exemple : Reprenons l'exemple des pingouins. Nous avons les deux arguments Arg₁ et Arg₂ avec deux conclusions contradictoires : « Titi vole » et « Titi ne vole pas »

Arg₁ << Titi est un oiseau, les oiseaux vols, Titi vole >

Arg₂ << Titi est un pingouin, les pingouins ne volent pas, Titi ne vole pas >

Si la relation de préférence utilisée et celle de degré de spécificité : on prendra le cas le plus spécifique qui est préféré au cas le plus général. Donc, «Titi est pingouin, les pingouins ne volent pas» est préféré à « Titi est un oiseau, les oiseaux volent ».

Donc, Arg₂ est préféré à Arg₁ : la conclusion « Titi ne vole pas » est la conclusion qui est acceptée.

5.3 Système multi-agents à trois niveaux (S3A)

Pour rester conforme au processus général cité en introduction (Castano, & al., 2007), notre système effectue l'alignement en trois étapes. Néanmoins, des modifications sont obligatoires au niveau des étapes 2 et 3 de ce processus (Oulefki 2008). Dans ce qui suit le détail des tâches effectuées à chaque étape est présenté. Mais avant, l'architecture du système est donnée.

5.3.1 Schéma général du système

Le schéma suivant présente une vue générale du système multi agent qui permet d'aligner deux ontologies à l'aide d'un ensemble de ressources externes et avec une intervention de l'utilisateur.

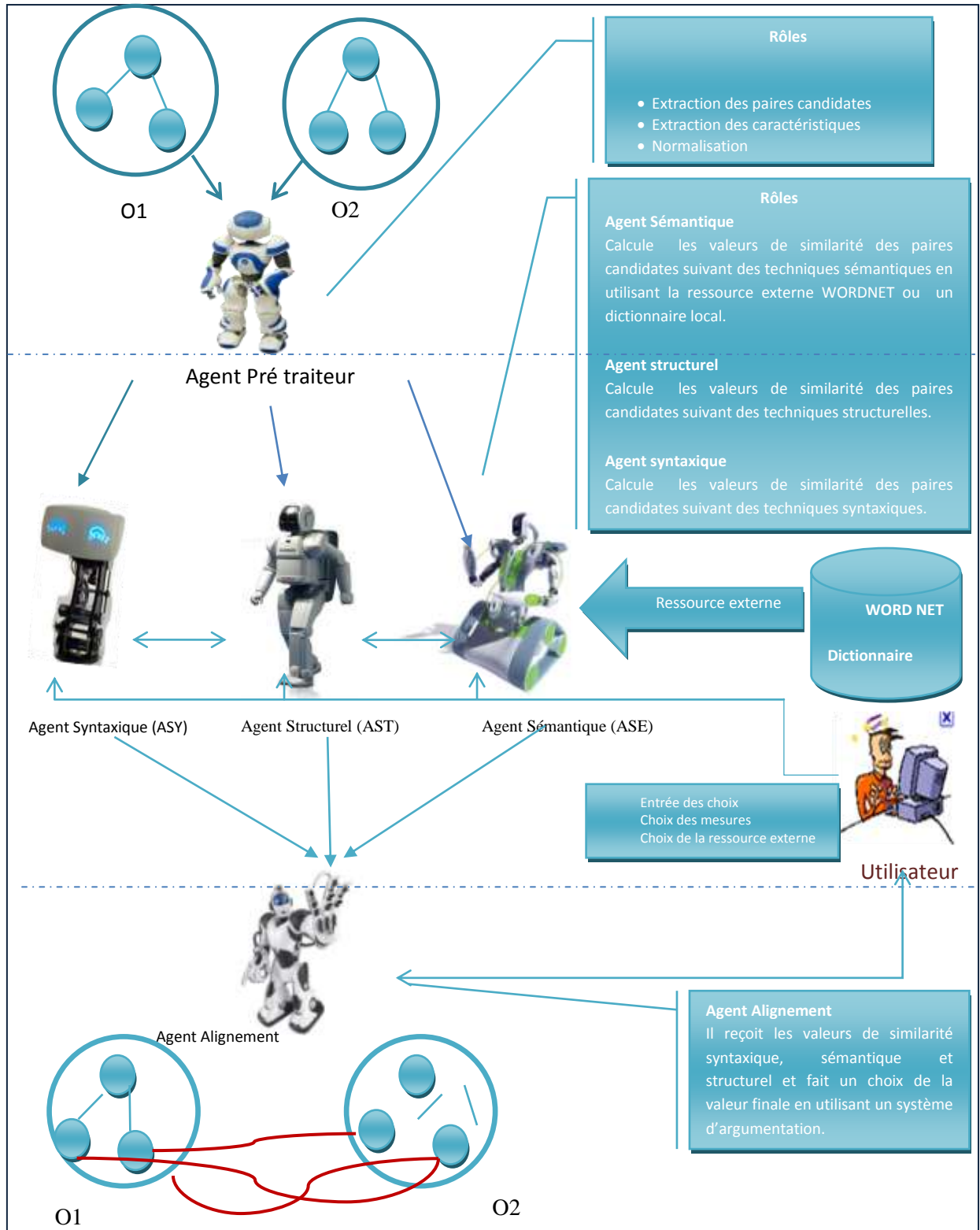


FIG. 5.2 – Schéma général du système S3A

5.3.2 Processus d'appariement

Notre système effectue l'alignement en trois étapes, selon le processus cité précédemment :

5.3.2.1 Pré-traitement

Le prétraitement est établi par l'agent "Pré-Traiteur" qui cherche les éventuelles correspondances. Ce dernier est chargé de l'extraction des paires candidates à partir de deux ontologies reçues en entrée. Ensuite, il extrait les caractéristiques nécessaires de chaque entité pour la normaliser. Enfin, il envoie les paires candidates aux agents du niveau 2, en ajoutant l'ensemble des liens de voisinages (sous concept, super concept, etc.). Le détail des traitements effectués est donné dans ce qui suit :

Agent Pré-traiteur (APR) : Cet agent est chargé de l'extraction des paires candidates à partir de deux ontologies reçues en entrée. Ensuite, il extrait les caractéristiques nécessaires de chaque entité (nom) pour la normaliser. Enfin, il envoie les paires candidates aux trois agents syntaxique, sémantique et structurel, en ajoutant l'ensemble des voisins nécessaires pour ce dernier. Le détail des traitements effectués est :

- *Extraction des paires candidates :* Dans cette étape, on doit comparer les entités des deux ontologies ayant le même type (concept, relation). Il s'agit de comparer une entité avec toutes les entités de même type appartenant à l'autre ontologie. On s'intéresse aux noms de classes et aux noms de relations.

- *Extraction des caractéristiques:* Les techniques que nous utiliserons sont les techniques linguistiques (sémantiques et syntaxiques) et structurelles d'où la nécessité d'extraire les caractéristiques "linguistique" et les caractéristiques "structurelles" .Pour les techniques linguistiques nous aurons besoin des noms des entités (concept et relation). Par ailleurs la structure taxonomique est utilisée pour le calcul de la similarité structurelle.

- *Normalisation des caractéristiques:* Avant d'entamer le calcul de similarité, les chaînes de caractères (longues et courtes) doivent être normalisées. Cette étape permet d'améliorer la comparaison entre les diverses chaînes de caractères. Pour avoir des noms normalisés on doit passer par les étapes suivantes:

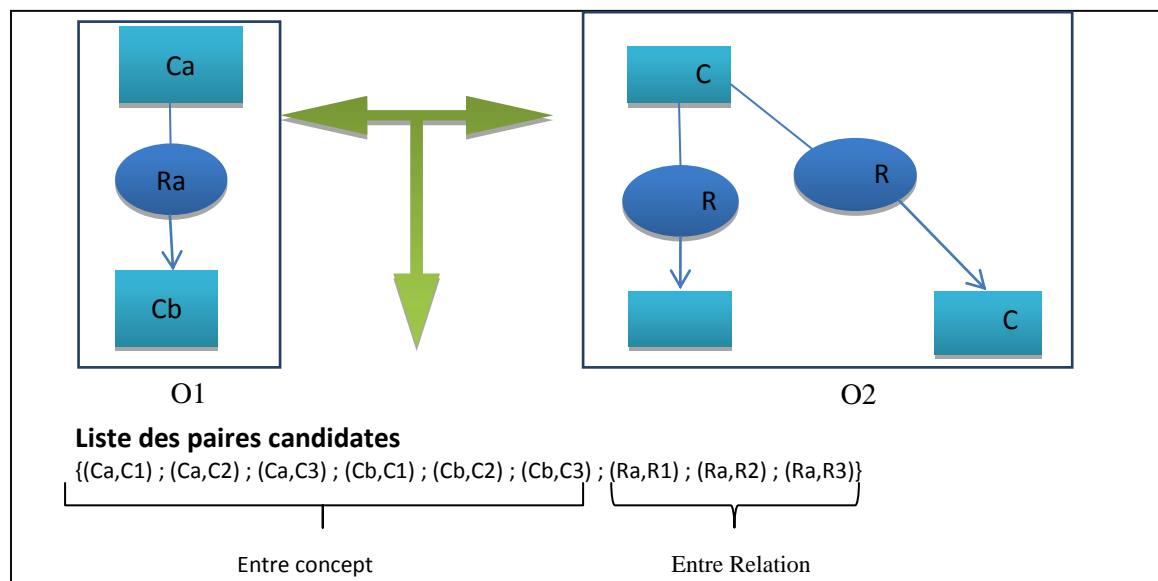


FIG. 5.3 – Exemple d'extraction des paires candidates

L'idée est de découper la chaîne de caractères en un ensemble de tokens, car nos chaînes peuvent être composées en plusieurs mots. Le découpage est basé sur les majuscules dans la chaîne et les caractères spéciaux comme les espaces, les "-", et les "_", etc.

Exemple: La tokenization « Num_BedsAttribute » donne l'ensemble [num, beds, attribute].

- *Élimination des mots vides:* Un mot vide est un mot non significatif parmi les articles, les conjonctions de coordination, les auxiliaires et leurs formes conjuguées, etc.

Pour la détection de ces mots, un fichier contenant tous les mots vides de la langue anglaise, car les ontologies utilisées sont en anglais, a été utilisé.

- *Convertir les caractères majuscules en leurs formes minuscules.*

5.3.2.2 Calcul des valeurs de similarité et Construction d'argument

Dans cette étape, les valeurs de similarité sont calculées par chaque agent qui dispose de ses propres techniques (syntaxique, structurel ou sémantique).

A/ Agent syntaxique (ASY) : L'agent syntaxique reçoit le résultat du prétraitement de l'agent pré-traiter (les paires candidates) et doit calculer la valeur de similarité en utilisant les techniques syntaxiques pour les chaînes courtes et longues de caractères. Cette valeur est obtenue par comparaison des deux ensembles de tokens. Si l'un des noms est une chaîne courte (respectivement longue) de caractères, la mesure de *Jaro-Winkler* (respectivement *d'hybridation de Jaro-Winkler*) est sélectionnée par défaut.

La mesure de Jaro-Winkler (Winkler, 1999) est une variante de celle de Jaro qui utilise un coefficient de préfixe P pour favoriser les chaînes ayant le plus long préfixe commun. Elle est définie comme suit (Bach, 2006) :

Soient s et r deux chaînes de caractères. La mesure de Jaro-Winkler $\alpha: S \times S \rightarrow [0, 1]$

$$\alpha_{\text{jaro_winkler}}(s,r) = \alpha_{\text{jaro}}(s,r) + (P \times Q \times (1 - \alpha_{\text{jaro}}(s,r)))$$

où la mesure de Jaro (Jaro, 1989) est une fonction de la similarité, qui est basée sur le nombre et l'ordre des caractères communs entre deux chaînes de caractères.

Pour la comparaison des chaînes longues de caractères, une méthode hybride a été proposée. Elle consiste à découper ces deux chaînes en plusieurs chaînes plus courtes. Ensuite, ces dernières sont comparées par une technique quelconque basée sur les chaînes courtes de caractères. Enfin, les résultats obtenus sont combinés.

Dans notre travail nous allons utiliser la mesure de JaroWinkler pour calculer la similarité entre les tokens. Ensuite, nous allons combiner les résultats en utilisant la formule donnée par la définition suivante :

Définition

Soient $s = a_1 \dots a_K$ et $t = b_1 \dots b_L$ deux chaînes de caractères, où a_i et b_j sont des sous-chaînes de s et t respectivement. Soit S une mesure de la similarité entre deux chaînes courtes de caractères. La similarité hybride est une fonction de la similarité

$\sigma_{Hybride} : S \times S \rightarrow [0, 1]$ telle que :

$$\overline{\sigma_{Hybride}}(s, t) = \frac{1}{K} \sum_{i=1}^K \max_{j=1..L} S(a_i, b_j)$$

D'autre part, l'utilisateur peut introduire son choix entre une variété de mesures mises à sa disposition pour chacune des techniques des chaînes courtes et chaînes longues qui sont les suivantes :

✓ **Mesure pour les chaînes courtes de caractères :**

Jaro-Winkler (sélectionnée par défaut), Jacard, etc.

✓ **Mesures pour les chaînes longues de caractères :**

Hybridation avec les techniques de chaînes courtes de caractères suivantes :

Jaro-Winkler (sélectionnée par défaut), Jaccard.

Le processus d'hybridation de la mesure de jaro-winkler est présenté dans l'algorithme 5.1.

Algorithme JaroWinklerHybride (entite1,entite2)

Debut

```
token1,token2 :liste de tokens correspondant à une entité
Max , val , sim, jaroWinkHyb: réel;
//pour chaque token de la liste des token1 de l'entite1
  Pour i=0 jusqu'à leng(tokens1)
    Debut
      //pour chaque token de la liste des token2 de l'entite2
      Max=0 ;
      Pour j=0 jusqu'à leng(tokens2)
        Debut
          // une fonction qui calcule la similarité de JaroWinkler
          val = metric2.getSimilarity(tokens1(i), tokens2(j));
          // Prendre le maximum des valeurs de similarité
          if (val>max) max=val;
        Fin
      sim=sim+max;
    Fin
  JaroWinkHyb = 1 /leng(tokens1)*sim;
Fin
```

ALG. 5.1 -- Algorithme qui calcule une similarité syntaxique pour les chaînes

B/ Agent Structurel (STR) : Cet agent est chargé d'exécuter les techniques structurelles (Oulefki 2008) en exploitant les relations de taxonomie (IS-A).

La similarité structurelle est calculée en utilisant la similarité linguistique, ainsi que la structure avoisinante. Pour chaque catégorie d'entités (dans notre cas, concept et relation) à aligner, on extrait ses voisins. Ensuite, les voisins de même type (pères, fils, sœurs) sont comparés. Trois types de voisins sont considérés (Exploitation de la mesure taxonomique) :

- les super-entités directes, qui sont les entités parentes directes ayant un lien de subsomption vers l'entité en question : $Sim_pere = similarite_voisin(pere_1, pere_2)$;
- les sous-entités directes, qui sont les descendantes ayant un lien de subsomption de l'entité en question : $Sim_fils = similarite_voisin(fils_1, fils_2)$;
- Les entités sœurs, qui ont la même entité mère que celles de l'entité en question :
 $Sim_sœur = similarite_voisin(soeurs_1, sœur_2)$.

Chaque valeur obtenue pour chaque type est multipliée par un poids introduit en entrée. La somme des trois valeurs donne la valeur de similarité structurelle. Le calcul de la similarité structurelle est illustré par les formules suivantes :

$$\text{Sim_structurel} = \text{poidsPere} * \text{sim_pere} + \text{poidsFils} * \text{sim_fils} + \text{poidsSœur} * \text{sim_soeur}$$

La similarité linguistique est calculée conformément à l'algorithme 5.2.

```

Algorithme sim_max(entite,list)
  Size : taille de la liste des voisins en entrée;
  Max : valeur de similarité linguistique maximale entre l'entité et les entités de la
liste des voisins ;
  Max=0 ;
Debut
  Pour i=0 jusqu'à size faire
    Envoi(entite,list.get(i), 'Agent Syntaxique') ;
    Envoi(entite,list.get(i), 'AgentSémantique') ;
    Recevoir(sim_syn,'agent syntaxique') ;
    Recevoir(sim_sem, 'agent semantique') ;
    Si (sim_sem=1) alors sim_ling=1 ;
      Sinon sim_ling=sim_syn ;
    si (sim_ling>max) Alors max =sim_ling;
  fait
  return(max) ;
fin

```

ALG. 5.2 – Calcul du maximum des similarités linguistiques entre deux entités

C/ Agent sémantique (ASE) : Pour calculer la valeur de similarité sémantique, cet agent utilise des ressources externes. Dans notre cas, nous avons utilisé deux ressources externes. La première est le dictionnaire wordnet et la deuxième est un dictionnaire que nous avons conçu « dict-local ». Le même principe est appliqué sur les deux ressources, pour identifier des correspondances entre des noms qui sont similaires sémantiquement (synonymes).

L'idée consiste à utiliser les listes de synonymes d'un mot ou d'un terme. Du fait que les mots de WordNet sont organisés en ensembles de "synsets" (synonym sets) et que le dictionnaire conçu est formé de regroupements de mots synonymes, l'extraction des synonymes d'une entité ne dépend que de la qualité (taille, diversité) de la ressource.

D/ Construction des Arguments : Chaque agent procède à la construction de ses propres arguments pour justifier sa décision d'accepter ou de refuser une correspondance suite au calcul fait conformément à la technique qui lui est allouée. Pour cela, les éléments de base du système d'argumentation suivi, sont définis dans ce qui suit :

– Langage : Vu que les prémisses des arguments sont les différentes correspondances reçues de l'agent pré-traiteur, les couples sont donc de la forme $\langle \text{Concept}_1, \text{Concept}_2 \rangle$. Le langage libéral est celui qui convient.

– Argument : Un argument, dans notre système, est défini par un uplet : $\text{Arg} : \langle \langle C_x, C_y \rangle, \text{ValSim}, \text{Decision} \rangle$, où :

- C_x et C_y sont respectivement deux concepts de l'ontologie O_1 et O_2 à aligner ;
- ValSim : est la valeur de similarité calculée par l'agent ;
- Decision : est la conclusion supportée par les prémisses. Elle peut prendre deux valeurs possibles "+" dans le cas où l'argument est pour la correspondance, et "-" dans le cas contraire.

– Relation de contrariété : Un argument ne peut être que pour ou contre une correspondance. De ce fait, la conclusion peut contrarier ou non une autre conclusion. La relation de contrariété employée est la relation de « réfutation ».

– Condition d'acceptabilité : Les deux types d'acceptabilité sont utilisés. Elle est individuelle dans le cas où l'argument est suffisamment fort pour se défendre, sinon, c'est l'acceptabilité conjointe qui est utilisée.

Donner la valeur positive ou négative à une conclusion supportée par la prémisses dépend de la valeur d'un seuil de confiance. Si la similarité entre deux entités est supérieure ou égale à ce seuil, la valeur de l'argument est positive, sinon elle est négative. Les arguments ainsi construits sont envoyés à l'Agent Alignement.

5.3.2.3 L'alignement

La troisième étape est assurée par un seul agent appelé "Agent d'Alignement". Dans le processus général (Castano & al., 2007), la moyenne des différentes valeurs de similarité est calculée. Dans notre système, cet agent qui reçoit les différentes valeurs de similarité ne va pas procéder de la même manière. Il doit prendre une décision d'accepter une correspondance ou la refuser sur la base des différents arguments et contre-arguments reçus. Pour ce faire, un système d'argumentation basé sur les préférences a été défini:

Définition du système d'argumentation :

Il est représenté par le triplet $\langle \text{EA}, \text{ref}, \text{Pref} \rangle$, où l'ensemble des arguments de EA est de la forme $\langle \langle C_x, C_y \rangle, \text{ValSim}, \text{Decision} \rangle$ comme précédemment ; La relation de contrariété

ref est une relation de réfutation sur EA x EA dans laquelle un argument réfute un autre argument si l'un accepte la correspondance (sa décision est "+") et l'autre la rejette (sa décision est "-"); la relation de préférence « Pref » entre les arguments est de type explicite conditionnelle incontestables, qui est définie à travers des règles suivantes:

1/ Préférence entre agent : l'agent sémantique est préféré aux deux autres agents, et l'agent structurel est préféré à l'agent syntaxique.

2/ Si l'argument A, concernant une correspondance, est donné par l'agent X qui est préféré par rapport aux agents formulant un argument le réfutant, alors A est préféré aux autres arguments.

Dans ce cas, une correspondance est acceptée si elle vérifie trois (3) conditions :

- Aucun agent ne formule un argument contre;
- L'argument la contrariant n'est pas préféré par rapport à l'argument la supportant (se défend seul : acceptabilité individuel) ;
- L'argument la contrariant est contrarié par un autre argument plus préféré (ne se défend pas seul : acceptabilité conjointe).

En fait, l'agent alignement commence d'abord par vérifier la décision de similarité entre deux entités (e1, e2) par exemple retournée par l'agent sémantique. Si elle est positive, l'argument retourné est positif. Dans le cas contraire, on vérifie la décision de similarité retournée par l'agent structurel. Le même raisonnement est suivi que celui de l'agent sémantique. Le structurel est préféré au syntaxique. Si toutes les décisions sont négatives, alors l'argument est négatif.

Le résultat de l'argumentation sera nécessairement un ensemble de correspondances entre les concepts. Ces correspondances peuvent être non pertinentes. Pour cela, une étape de filtrage est obligatoire.

5.3.2.4 Le filtrage

Il va nous permettre de ne garder que les correspondances jugées pertinentes. Habituellement, cette étape est faite par l'utilisateur qui accepte ou refuse une correspondance. Il arrive aussi qu'un simple algorithme de sélection de la correspondance disposant d'une valeur de similarité la plus élevée soit exécuté. Mais, cette vision est en contradiction avec les règles d'acceptation que nous avons définie précédemment. Dans ce

cas, l'algorithme ne donne pas de bon résultat. L'exemple suivant décrit l'insuffisance de cet algorithme.

Exemple :

Après l'application des règles d'argumentation sur le fichier d'alignement on aura les correspondances suivantes :

Correspondance 1 : $\langle\langle C1, C3 \rangle, 0.80, + \rangle$ valeur de similarité syntaxique.

Correspondance 2 : $\langle\langle C1, C3 \rangle, 0.70, + \rangle$ valeur de similarité structurelle.

La correspondance 1 a une valeur de similarité supérieure à la 2^{ème}. Donc on choisit la correspondance 1. Or, une similarité structurelle est plus forte que la syntaxique. Pour cela, une étape de filtrage basée sur les préférences est considérée (voir algorithme 5.3).

D'une façon plus formelle, un système d'argumentation pour le filtrage basé sur les préférences est un triplet $SAFPref = \langle EA, att, Pref \rangle$ où :

- EA est un ensemble d'arguments de la forme $\langle\langle Cx, Cy \rangle, ValSim, Tech, Conclusion \rangle$ où :
 - Cx et Cy sont deux concepts de l'ontologie O1 et O2 à aligner respectivement,
 - ValSim est la valeur de similarité calculée par l'agent,
 - Tech est la technique avec laquelle on a obtenu la valeur de similarité. Les trois valeurs possibles sont «sémantique», «structurelle» et «syntaxique».
 - Conclusion représente la décision de l'agent.
- att : est une relation de contrariété de type attaque dans le cas où un argument attaque les prémisses.
- Pref : est une relation de préférences conditionnelles sur les techniques utilisées et les valeurs de similarité.

```

Algorithme test_Correspondance (C1,C2,C3)
Debut
List_correspondance={ } ;
si(tech(C1,C2)= 'sémantique') alors
    si (tech(C1,C3)= 'sémantique' ) alors
        list_correspondance={(C1,C2),(C1,C3)} //garder les deux correspondances
    sinon Si (tech(C1,C3)='structurelle' ou tech(C1,C3)='syntaxique' ou
        tech(C1,C3)='structurelle_syntaxique') Alors
        List_correspondance ={(C1,C2)} ;
        Finsi ;
    Finsi ;

Sinon Si (tech(C1,C2)= 'structurelle') alors
    Si (tech(C1,C3)= 'structurelle') alors
        Si (ValSim(C1,C2)> ValSim(C1,C3)) alors List_correspondance={(C1,C2)}
        Sinon Si (ValSim(C1,C2)=ValSim(C1,C3)) alors
            List_correspondance={(C1,C2),(C1,C3)}
            Sinon List_correspondance={(C1,C3)}
            Finsi ;
        Finsi ;

    Sinon Si (tech(C1,C3)= 'sémantique' ou tech(C1,C3)='structurelle_syntaxique')
        alors List_correspondance={(C1,C3)}
        Sinon si (tech(C1,C3)= 'syntaxique') alors List_correspondance={(C1,C2)}
        FinSi ;
    Finsi ;

    Sinon Si((tech(C1,C2)= 'syntaxique') alors
        Si(tech(C1,C3)= 'sémantique' ou (tech(C1,C3)= 'structurelle' ou
            tech(C1,C3)= 'structurelle_semantique') alors
            List_correspondance={(C1,C3)}
        Sinon si tech(C1,C3)= 'syntyaxique' Alors
            Si (ValSim(C1,C2)> ValSim(C1,C3)) alors
                List_correspondance={(C1,C2)}
            Sinon Si (ValSim(C1,C2)=ValSim(C1,C3)) alors
                List_correspondance={(C1,C2),(C1,C3)}
            Sinon List_correspondance={(C1,C3)}
            Finsi ;
        Finsi ;

        Si (pas correpondnace(C1,C2)) alors Si (exit_correspondance(C1,C3)) alors
            List_correspondance={(C1,C3)}
            Finsi ;
        Finsi ;
    Finsi ;
Finsi ;
Fin .

```

ALG. 5.3 – filtrage des correspondances

5.4 Tests et Evaluation

Dans le domaine de l'évaluation des méthodes d'appariement d'ontologies, nous nous sommes intéressés aux travaux dans la littérature de la campagne OAEI (Ontology Alignment Evaluation Initiative). Le détail du processus d'évaluation que l'on présentera par la suite est extrait de (Zghal, & al. 2007) :

- La première phase consiste à appairer manuellement les paires d'ontologies de tests. Le résultat obtenu est considéré comme l'alignement de référence, dénoté par A_1 . L'ensemble des correspondances contenues dans ce dernier est dénoté par « Ntrue ».

- Dans la deuxième phase, le système d'appariement à évaluer est exécuté sur les paires d'ontologies de tests. Cela produit un alignement, dénoté par A_2 , qu'il faut évaluer dans les phases ultérieures. L'ensemble des correspondances contenues dans ce dernier est dénoté par « Nfound ».

- La troisième phase est la comparaison de l'alignement de référence avec celui obtenu par le système d'appariement à évaluer. Cette comparaison a pour objectif de trouver l'intersection des deux ensembles Nfound et Ntrue, c'est-à-dire, l'ensemble des paires appartenant à la fois à A_1 et A_2 . Cette intersection est dénotée par « Ncorrect ».

- La dernière phase consiste à évaluer la qualité de A_2 en calculant la valeur des mesures « précision », « rappel » et « f-mesure » (Bach, 2006), (Zghal et al. 2007). Ce sont des métriques largement utilisées et exploitées pour estimer la qualité des alignements obtenus. Elles permettent une analyse fine des performances du système.

La précision est la proportion des correspondances correctes, i.e., Ncorrect, parmi l'ensemble de celles contenues dans A_2 , i.e., Nfound. Cette mesure reflète la précision d'un outil. Plus la valeur de précision est élevée, plus le bruit dans le résultat de l'outil est réduit, et donc plus la qualité du résultat est imposante. Ainsi, la fonction précision est définie par :

$$\text{Précision} = |\text{Ncorrect}| / |\text{Nfound}|.$$

Le rappel est la proportion de correspondances correctes contenue dans A_2 , i.e., Ncorrect, parmi toutes celles qui sont correctes, i.e., Ntrue. Le rappel mesure l'efficacité d'un outil. Plus la valeur de rappel est élevée, plus A_2 couvre toutes les correspondances correctes. La fonction *rappel* est donc définie par :

$$\text{rappel} = |\text{Ncorrect}| / |\text{Ntrue}|.$$

Plus la valeur de rappel est élevée, plus le résultat de l’algorithme couvre toutes les correspondances correctes. La Fmesure permet d’estimer le pourcentage d’erreurs obtenu au cours du processus d’alignement. C’est un compromis entre le rappel et la précision. Plus F-mesure augmente, plus la mesure est performante. La f-mesure est définie par :

$$f\text{-mesure} = (2 * \text{Rappel} * \text{Précision}) / (\text{Rappel} + \text{Précision})$$

Ce sont, ces mesures que nous avons utilisées pour évaluer notre système S3A. Ce dernier a fait l’objet de plusieurs tests sur des Benchmarks afin de valider l’approche utilisée. A chaque test, le calcul des différentes valeurs a été fait. D’après les résultats obtenus, l’approche est très satisfaisante au vue des valeurs obtenues par f-mesure. En effet, la valeur s’approche souvent de 1 et elle est supérieure la plupart du temps à 0.8. La figure nous montre les résultats sur une petite ontologie appelée "Hôtel 1".

Technique Syntaxique	Seuil syntaxique	Technique structurale	Seuil structurel	Dictionnaire	Précision	Rappel	F-mesure
JaroWinkler	0.7	Super Classes	0.7	WordNet	0.7	1.0	0.82
JaroWinkler	0.7	Super Classes	0.8	WordNet	0.875	1.0	0.93
JaroWinkler	0.7	sous Classes	0.7	WordNet	0.875	1.0	0.93
JaroWinkler	0.7	Soeur	0.7	WordNet	0.777	1.0	0.87
JaroWinkler	0.9	Soeur	0.7	WordNet	0.875	1.0	0.93

TAB. 5.1 – Test de l’ontologie Hôtel

Après avoir testé avec des différents seuils, nous allons fixer le seuil à 0.70, vu que ce seuil donne de meilleurs résultats pour les ontologies testées. Puis, nous modifions à chaque fois la valeur des poids des voisins. Les tests seront fait sur la paire d’ontologie «Network». Les résultats sont présentés dans le tableau suivant : Le résultat de l’évaluation représenté par la mesure f-mesure, qui définit les performances d’un outil d’alignement, varie selon le seuil de similarité donné par l’utilisateur. Par exemple pour la paire d’ontologie «Network», on a la valeur f-mesure égale à 1.0 pour le seuil 0.70, et égale à 0.89 pour un seuil de 0.80. Cette variation est due au fait que la liste des correspondances dépend du seuil choisi. Autrement dit, seulement les correspondances ayant une valeur supérieure au seuil sont gardées. Pour ce premier test, on a fait varié le seuil et on a gardé les valeurs des poids des voisins, introduit dans la mesure structurelle, à 1/3 pour chaque type de voisin, c'est-à-dire tous les voisins ont

le même niveau d'importance. Dans la deuxième catégorie de tests, nous avons fixé le seuil à 0.70, et nous avons varié les poids des voisins, en favorisant à chaque fois un type de voisin. Les résultats obtenus, varient selon les valeurs affectées au poids. Par exemple, pour une combinaison de (0,1/3,2/3), pour les poids des pères, frères, fils respectivement, on a la valeur f-mesure égale à 1.0, tandis qu'avec la combinaison (0,0,1), elle est de 0.64. Donc, on ne peut pas dire qu'on favorisant les fils on a de meilleurs résultats. De même pour la combinaison (2/3,0,1/3), on a f-mesure égale à 1.0, et pour la combinaison (1,0,0), f-mesure est de 0.64.

On ne peut donc conclure qu'on donnant la plus grande valeur aux frères, cela donne de meilleurs résultats.

Après avoir testé sur d'autres valeurs de seuil, ainsi que d'autres paires d'ontologies, nous avons constaté que généralement c'est la combinaison (1/3,1/3,1/3) qui donne de meilleurs résultats. C'est-à-dire que tous les voisins ont le même niveau d'importance (voir tableau 5.2).

Poid super classe	Poid sous Classe	Poid soeur	Seuil syn-taxique	Seuil Structu-rel	Precision	Rappel	F-mesure
1/3	1/3	1/3	0.9	0.8	0.875	1.0	0.93
1/3	2/3	0	0.9	0.8	0.875	1.0	0.93

TAB. 5.2 – Test de l'ontologie Hôtel avec les poids

5.5 Conclusion

Dans ce chapitre, le système d'alignement d'ontologies que nous avons développé est présenté. Basé sur la technologie agent, il prend en compte trois techniques d'appariement réalisées chacune à travers un agent. Les résultats obtenus sont exécutés par un système d'argumentation à base de préférences. Le système est satisfaisant en termes de précision et de rappel, au vu des tests effectués sur des Benchmarks.

Cependant, il n'est pas systématiquement possible d'aligner les ontologies des agents mis en jeu. En effet, en considérant les hypothèses principales des travaux de Morge et Routier (Morge et Routier, 2007), rien n'assure que l'alignement obtenu soit correct et complet. Or, si l'alignement est imparfait, la communication devient difficile voire inopérante (Sabouret,

2009). C'est pourquoi une alternative au problème de l'hétérogénéité sémantique est le dialogue argumentatif entre les agents détenteurs des ontologies, c'est-à-dire la communication est gérée directement au moyen d'un protocole de négociation sémantique (Morge et Routier 2007). C'est ce que nous aborderons dans le chapitre suivant.

Chapitre 6

Dialogue argumentatif appliqué au E-recrutement

“Entre ce que je pense, ce que je veux dire, ce que je crois dire, ce que je dis, ce que vous avez envie d’entendre, ce que vous croyez entendre, ce que vous entendez, ce que vous avez envie de comprendre, ce que vous croyez comprendre, ce que vous comprenez, il y a 10 possibilités qu’on ait des difficultés à communiquer. Mais essayons quand même...”

(Bernard Werber, Encyclopédie du savoir relatif et Absolu).

6.1 Introduction

Pour aborder le dialogue argumentatif, nous nous penchons sur la notion d’ontologie dans un système interactif, en particulier au niveau d’un agent dans un SMA hétérogène, qui est qualifié d’ouvert. Les agents, dans ce cas, sont amenés à communiquer pour résoudre les tâches et buts qui leurs sont assignés. Or, dans un contexte ouvert, chaque agent est construit de manière indépendante ainsi que l’ontologie qu’il utilise pour décrire formellement ses connaissances. L’exploitation des techniques traditionnelles de communication est donc inappropriée. En effet, les agents, dans ce cas, ne partagent pas forcément la même définition d’un objet, et donc chacun possède sa propre ontologie. Au terme du dialogue qui les confronte, les agents doivent atteindre un accord qui concerne la définition de cet objet et

l'inférer par la suite. Autrement dit, lorsqu'un agent *A* souhaite communiquer avec un agent *B*, il va utiliser sa propre ontologie pour construire ses messages. L'agent *B* recevra alors un message formulé selon les termes de l'ontologie de l'agent *A*, rendant le message difficilement interprétable. La question est de savoir comment un agent *A* peut transmettre le sens d'un concept à un agent *B* et comment l'agent *B* peut utiliser ces informations pour établir un alignement. Le problème de l'hétérogénéité dans les agents se situe donc dans la confrontation des ontologies des deux agents et la définition de méthodes pour les réconcilier (Sabouret, 2009).

Dans ce chapitre et contrairement au chapitre précédent, nous nous intéressons à la réconciliation des ontologies entre agents interagissant dans un contexte ouvert dans le cadre du recrutement électronique. Notre contribution, dans ce cas, consiste à réaliser un système dialogique permettant d'assurer un rapprochement sémantique entre les offres et les demandes d'emploi. Une architecture, basée sur la négociation pour permettre aux agents d'aboutir à un consensus sur leurs représentations, est adoptée.

6.2 Communication/ Argumentation Inter-Agents

Depuis quelques années, on a cherché à permettre aux programmes informatiques de s'échanger des informations et des connaissances. Des techniques comme, par exemple, RMI¹ (Remote Method Invocation), CORBA² (Common Object Resource Broker Architecture) ont déjà été utilisées, indépendamment de la notion d'agent. Cependant, ces techniques sont limitées car elles restent cantonnées au cadre de la programmation impérative ou procédurale. Elles sont mal adaptées aux systèmes multi-agents. C'est pourquoi, sont apparus les langages de communication entre agents (ACLs : Agent Communication Languages). Deux standards ont émergés à partir des études linguistiques et de modélisation de connaissances: KQML (Knowledge Query and Manipulation Language) (Finin, & al., 1994) et FIPA ACL (Foundation for Intelligent Physical Agents) (Fipa, 2002). Le format utilisé pour l'échange des connaissances est donné par un langage de contenu, indépendant du langage ACL soit : KIF, FIPA-SL ou FIPA-CCL. Le vocabulaire commun concerne les définitions précisées dans une ontologie. Ces langages de communication sont basés sur la théorie des actes de langage

¹ <http://www.oracle.com/technetwork/java/javase/tech/index-jsp-136424.html>

² <http://www.omg.org/gettingstarted/corbafaq.htm>

(Searle, 1969) qui permet d'identifier les constituants élémentaires de la communication. C'est la raison pour laquelle, elle a été d'une grande influence dans le domaine des SMA.

6.2.1 Communication inter-agents

Pour que les agents du système puissent communiquer, ils doivent partager un vocabulaire commun, c'est-à-dire une ontologie commune. Or, dans un SMA ouvert, les agents ne partagent pas forcément la même ontologie. De ce fait, la description de leurs connaissances peut être différente. La communication ne peut pas se faire uniquement par l'échange de simples messages. De nouveaux moyens de communication doivent être trouvés.

Dans ce sens, des travaux, pour la réconciliation d'ontologies (Bailin et Truszkowski, 2001), (Cuenca-Grau, & al., 2005), (Jingshan, & al., 2005), (Van digglen, & al., 2006), ont été présentés. Ceci consiste en l'utilisation de ponts pour réconcilier et relier deux ontologies présentant quelques similarités. Ces ponts ont la propriété de servir de « médiateur » entre les notions utilisées par les différentes ontologies. Mais, la contrainte du contexte ouvert limite l'exploitation de ces méthodes de réconciliation d'ontologies.

Une autre solution a été présentée (Morge, & al., 2006 a, b). Elle utilise des techniques d'argumentation dans le but de fournir un mécanisme de dialogue permettant aux agents d'aboutir à un consensus sur leurs représentations et ainsi permettre de sélectionner les services répondant aux spécifications des utilisateurs.

Dans ce travail, nous adoptons cette méthode (le dialogue argumentatif) pour permettre la communication inter-agents dans un contexte ouvert. Elle découle d'une approche que nous avons proposée (Akli, 2006) et qui est basée sur un processus de négociation. Nous présentons ci-dessous ces fondements théoriques.

6.2.2 Le dialogue argumentatif

Il n'existe pas de consensus quand à la définition du dialogue argumentatif. Dans le cadre de notre travail, la définition suivante est adoptée (Amgoud, 1999): « Le dialogue argumentatif est un mécanisme formel de dialogue à travers duquel deux agents dialoguent et argumentent pour aboutir à un consensus ontologique ». Ce modèle va permettre aux agents de justifier leurs représentations et gérer ainsi leurs conflits ontologiques.

Nous présentons, dans ce qui suit, les concepts sur lesquels s'appuie le dialogue argumentatif.

6.2.2.1 Le système de dialogue

En règle générale, un dialogue est un enchaînement de phrases énoncées par deux interlocuteurs. Entre agents artificiels dans les SMA, un système de dialogue est une famille de dialogues régulés, c'est à dire un dispositif à travers lequel un ensemble de participants communiquent en respectant certaines règles (Hamblin, 1970).

Dans le cas le plus simple, un dialogue se déroule entre deux participants qui prennent la parole chacune à leur tour. Elles interrogent et/ou répondent à leur partenaire via des locutions en prenant en compte les locutions précédentes. Le système dialectique proposé par Hamblin utilise deux notions essentielles : la notion de convention et celle de tableau d'engagements :

La notion de convention : C'est un ensemble de règles dialectiques avec les propriétés suivantes :

- **Les règles sont contextuelles** : Elles dépendent de l'histoire passée du dialogue, le plus souvent, seul le dernier coup est considéré.
- **Les règles sont déontiques** : Elles spécifient les réponses qui sont autorisées et celles qui sont interdites.
- **Les règles dialectiques peuvent être individuelles** : La cible est un participant ou un ensemble de participants.

La notion de tableau d'engagement: Les tableaux d'engagements sont les structures de données qui répertorient les engagements (les promesses) pris par les participants au cours du dialogue à propos d'un certain contenu, comme une proposition ou une action, selon certains termes. Ces tableaux visent à sauvegarder la progression de la conversation. Un tableau est associé à chaque agent, et il est vide au début du dialogue. Ensuite, les engagements pris par chaque agent durant le dialogue sont sauvegardés dans son tableau. Le contenu des tableaux d'engagement est public, donc accessible à tous les agents participant au dialogue, contrairement au contenu des bases représentant le raisonnement interne des agents qui est privé.

Nous nous sommes intéressés à l'approche d'engagement de Maudet (Maudet, 2001) qui vise d'abord l'étude du dialogue humain afin de bien comprendre les mécanismes sous-jacents et les structures employés lors de l'activité du dialogue. Cette étude vise à comprendre la manière dont il est possible de construire un bon dialogue, structuré et cohérent, entre différents participants. Parmi les quatre types d'engagements qu'il a défini, en action, propositionnels, conditionnels et conjoints, ce sont les engagements propositionnels qui

s'adaptent bien à notre problématique. Dans ce cas, un agent s'engage envers un autre agent sur une certaine proposition.

C'est dans les tableaux de conversation que l'on retrouve les collections des engagements des participants à l'activité dialogale. Des opérations de manipulation de ces engagements sont donc nécessaires. Il s'agit des opérations de (Maudet 2001) :

- création : permet l'ajout d'un nouvel engagement dans le tableau d'engagement;
- annulation : permet le retrait d'un engagement présent dans le tableau d'engagement;
- satisfaction : permet de retirer du tableau, les engagements qui sont satisfaits suite à un événement;
- priorité : permet de donner des priorités entre les différents engagements se trouvant dans le tableau de conversation;
- atteinte : cette opération permet de recevoir une sorte de notification indiquant si le tableau a atteint un état particulier.

6.2.2.2 Les types de dialogue multi-agents

Les théoriciens de l'argumentation (Walton et Krabbe, 1995) ont suggéré une classification concernant les différents types de dialogues que les humains utilisent régulièrement. Elle est basée sur la situation initiale du dialogue, les buts individuels des agents, et les buts communs auxquels tous les agents adhèrent implicitement. Cette typologie, qui ne se veut pas exhaustive, a permis de mettre en évidence six types de dialogues:

Le dialogue de persuasion : La situation initiale est un désaccord, c'est-à-dire un point de vue conflictuel. Dans ce dialogue, chacun essaie de persuader son interlocuteur d'adopter son point de vue. Cela signifie que le processus dialectique est couronné de succès dans la mesure où l'un au moins des participants a changé d'opinion au cours du dialogue. Le but global étant la résolution du conflit.

Le dialogue de négociation: La situation initiale est un conflit d'intérêt mêlé à un besoin de coopération. Le but consiste à réaliser un échange ou conclure une affaire, c'est-à-dire trouver un compromis attractif pour les participants.

Le dialogue d'investigation : Une enquête tache d'établir (ou de prouver) la véracité de propositions dans le but de répondre à une question ouverte afin qu'un accord stable et général émerge.

Le dialogue de délibération: Une délibération, comme une enquête, débute sur un problème ouvert et non pas sur un conflit. Le problème consiste en une décision qui devra (ou non) être mise en œuvre. Ici, chacun des participants tente d'influencer le résultat. Le but global étant la décision d'un plan d'actions.

Le dialogue d'interrogation: Un dialogue de recherche d'information aura lieu lorsqu'il y a une asymétrie entre les agents dans le sens où un agent a une information incomplète sur un sujet particulier p , et il suppose que l'autre agent détienne plus d'informations, par exemple parce que cet agent est reconnu comme un expert sur le sujet ou que la connaissance de l'un est plus complète que celles de l'autre. Les buts individuels des agents sont soit de demander, soit de donner une information. Par exemple, le premier agent pose une question au second, du genre « est-ce que p ? », en ayant pour objectif que le second agent complète son manque d'information en avançant des arguments en faveur ou contre p . Le but du dialogue est de transférer l'information d'un agent à l'autre.

Dans ce travail, nous nous intéressons au dialogue de demande d'information ou interrogation. Nous avons considéré l'application d'e-recrutement pour réaliser ce dialogue.

6.3 Le e-recrutement

Nous avons vu les problèmes dues à l'hétérogénéité des agents dans le contexte ouvert, et l'utilisation du dialogue argumentatif pour la résolution de ce problème. Dans ce contexte, le recrutement électronique semble être une bonne application. En effet, le recrutement électronique manipule une grande source d'information fournit par différentes organisations, ayant donc chacune son propre vocabulaire.

6.3.1 Définition

Le recrutement électronique (appelé aussi le e-recrutement) est l'utilisation des ressources électronique, en particulier internet pour assister au processus de recrutement. Du côté chercheur d'emploi, cette approche lui offre une réactivité pour la mise à jour de ses informations personnelles ainsi qu'un gain important de temps. Tandis que du côté recruteur, en plus d'avoir un nombre important de candidats, il peut bénéficier d'une masse de diffusion importante de ses offres d'emploi, d'une rapidité pour contacter les candidats, d'une réactivité

pour la mise à jour de ses offres, etc. Plus loin même, pour certaines sociétés, la publication en ligne de leurs offres d'emploi, est un signe d'une bonne économie (Bource, & al., 2002).

6.3.2 Processus de recrutement électronique

D'un point de vue organisationnel, un processus de recrutement typique peut être divisé en quatre phases importantes (Mochol, & al., 2004):

Analyse des besoins : L'organisation prévoit ses besoins en recrutements grâce à la GPEC (Gestion Prévisionnelle des Emplois et Compétences). La description des requis exigés pour un poste ouvert est formulée en texte libre, ce qui limite le traitement automatique plus tard dans le processus de recrutement. L'idéal serait l'usage d'un « langage » commun sous la forme d'un ensemble de vocabulaires contrôlés pour décrire les détails d'une offre d'emploi facilitant ainsi la communication.

Publication de l'offre d'emploi : Il existe un certain nombre de portails de travail commerciaux, tels que Monster³, JobPilot⁴ et StepStone⁵, qui se concurrencent afin de publier les offres d'emploi des recruteurs pour des frais de publication. Ces portails sont différents dans leurs stratégies et généralement spécialisés pour certaines régions géographiques. Les recruteurs peuvent aussi publier leurs offres sur leurs propres sites, afin de réduire les frais de publication. Actuellement, même en Algérie, il existe des portails de recrutement, des sites d'emplois, tels que Twdif (www.Tawdif.com), Emploitic (www.Emloitic.com), Algerie-Emploi (www.Algerie-Emploi.net), permettant au recruteur de publier leurs offres d'emplois et aux candidats de déposer leur candidature.

Réception et présélection des demandes des candidats : Les demandes d'emploi, pour le poste de travail offert, sont analysées par le recruteur manuellement. Leur classement se fait selon un ordre de pertinence, dans un temps raisonnable. Pour réduire les coûts supportés, les candidatures en ligne sont préférées. Par exemple, des sociétés comme IBM⁶ ou Volkswagen⁷ ont imposé des formulaires Web prédéfinis, à remplir par les candidats. Ces données sont utilisées pour la présélection automatique des candidats répondants aux requis de l'offre d'emploi.

³ : www.monster.com

⁴ : www.jobpilot.com

⁵ : www.stepstone.com

⁶ : <https://forms.bpfj.intronet.com/ibm/Forms/emea/expro/GeneralApp.jsp>

⁷ : <https://www.vw-personal.de/content/www/de/bewerbung/onlinebewerbung.html>

Interview et décision de recrutement finale : Dans cette phase, un entretien est réalisé avec les candidats dans un but d'évaluer directement leurs compétences. Cette phase est manuelle.

6.3.3 Un système multi-agents pour le e-recrutement, pourquoi ?

L'exploitation de la technologie des agents dans le cadre du recrutement électronique semble être bénéfique pour les raisons suivantes :

L'inefficacité de la recherche : L'un des problèmes affronté par le recrutement électronique est l'inefficacité de la recherche; la recherche du candidat le plus qualifié, via des moteurs de recherche spécifiques (propre à un site), ou universels n'est pas pertinente. Avec de tels systèmes, la précision est faible, car toute la difficulté est de savoir quelle offre s'accorde le mieux au profil des candidats. Or, par définition un agent est autonome et proactif donc non seulement il peut décider sans l'intervention d'un être humain mais aussi il agit en vue d'atteindre ses objectifs. Donc, on peut se servir d'un agent pour analyser et sélectionner les Csv (selon l'intérêt) automatiquement et en un temps raisonnable.

La distributivité des sources de données : Les portails de travail disponibles sont nombreux mais indépendants. Ils ont divisé le marché du travail en ligne en îles d'informations isolées (Bizer, & al., 2005), ce qui a rendu impossible, pour un chercheur d'emploi, d'avoir une idée globale de tous les postes de travail offerts.

Dans ce contexte, le besoin d'agents logiciels capables de communiquer avec les autres à la place de leur utilisateur est ressenti. En effet, les agents sont capable d'un comportement social et ils peuvent s'engager dans des interactions complexes, par exemple coopération, compétition, négociation et ainsi ils peuvent trouver les offres d'emploi les plus convenables.

6.4 Présentation du système

Pour aborder la partie conception de nos systèmes, nous suivons certaines étapes définies dans (Akli, 2006). La première, consiste en une phase de recueil et d'expertise permettant de développer les ontologies. La seconde, phase d'exploration et de dialogue, se traduit par des questions, des justifications et des commentaires entre les agents pour aboutir à un consensus. Mais avant, l'architecture de notre système est présentée.

6.4.1 L'architecture globale du système

L'architecture globale du système dans le cadre du recrutement électronique proposé est illustrée dans la figure 6.1. Elle est constituée des composants suivants :

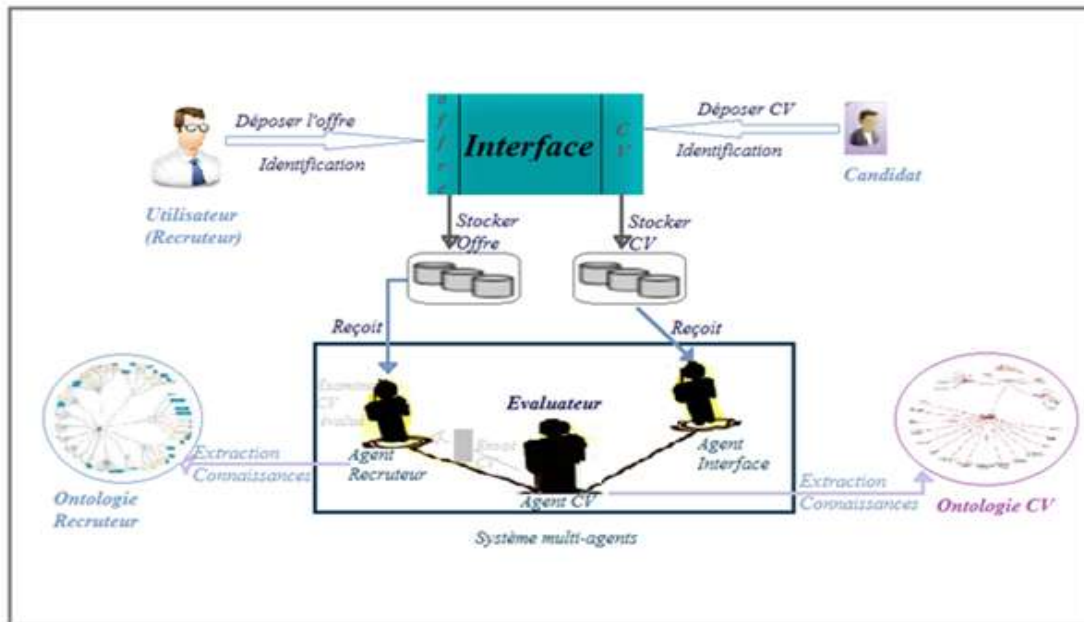


FIG. 6.1 -- Architecture globale du système proposé

L'interface : Elle donne la possibilité de déposer d'une part le CV pour le candidat, et d'autre part l'offre d'emploi pour le recruteur. Elle permet aussi de trouver les CV qui correspondent le plus au recruteur sans faire appel au SMA.

Le système multi-agent : L'objectif de ce système est de trouver un CV satisfaisant l'offre du recruteur. Il est constitué de trois agents (voir figure 6.2):

- **L'agent interface** : Qui sert à stocker les CV des enseignants déposés par ces derniers.
- **L'agent CV** : Qui évalue, trie les CV et sélectionne le CV le plus approprié à l'offre du recruteur selon un degré correspondant à une note attribuée à chaque CV et qui dépend des exigences du poste, et croît selon un ordre de priorités défini par l'organisme recruteur. Il extrait ces connaissances à partir de l'ontologie CV.
- **L'agent recruteur** : est un assistant du recruteur qui détient l'offre ; il entre dans le système et examine les CV sélectionnés par l'agent CV. Il extrait ces connaissances à partir de l'ontologie Recruteur.

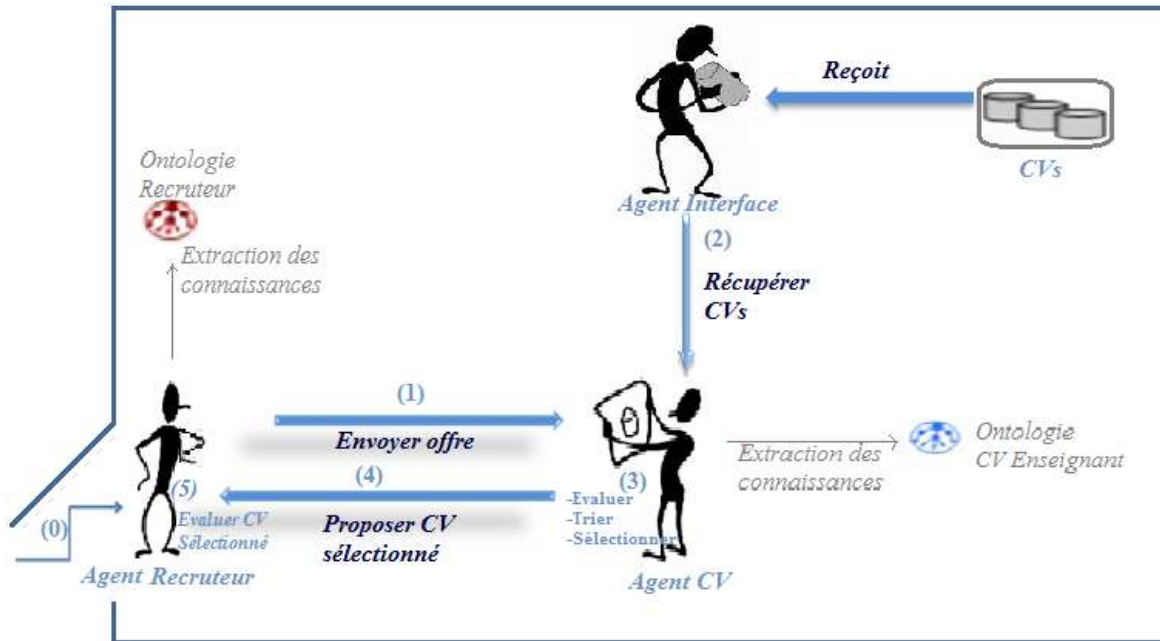


FIG. 6.2 -- Le système multi-agents

L'ontologie CV : C'est une ontologie construite pour la modélisation du contenu des CV des candidats. Elle sert aussi comme une base de connaissance pour l'agent CV.

L'ontologie Recruteur : C'est une ontologie construite pour la modélisation du contenu des offres d'emploi, dans le cadre de l'enseignement supérieur. Elle modélise aussi les connaissances sur les activités universitaires des enseignants.

6.4.2 Recueil et expertise

La première étape consiste à développer deux ontologies pour le recrutement électronique. Il est à noter que le développement d'une ontologie n'est pas évident car il faut réfléchir à comment représenter les connaissances, c'est-à-dire décider quels sont les concepts et leurs notions ainsi que les relations à représenter. Il est obligatoire de modéliser le monde réel avec des concepts et des relations non ambiguës. Comme aucune méthode de conception des ontologies n'est standardisée, nous avons opté pour une méthodologie combinant un guide de construction d'une ontologie (Noy et McGuinness, 2001) avec un modèle de données exprimé sous forme de diagramme de classe UML. Les deux ontologies ainsi construites, l'une pour l'agent CV et l'autre pour l'agent Recruteur, vont servir, comme Bases de Connaissances pour les agents. La même démarche de construction des ontologies est suivie par les deux agents.

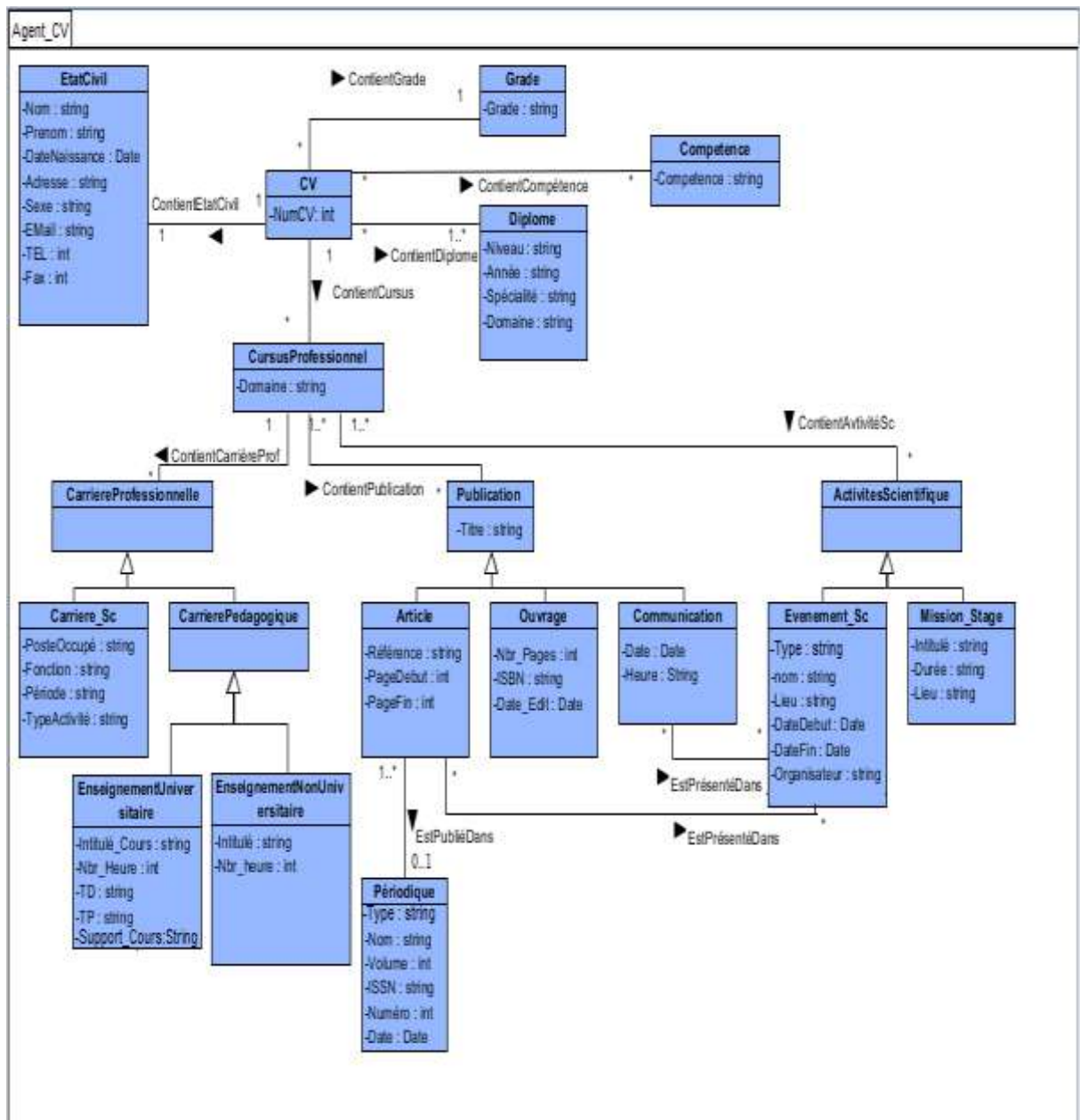
Définition du domaine et de la portée de l'ontologie : Le domaine que va couvrir l'ontologie CV correspond aux documents CV des candidats. Le but opérationnel de cette ontologie est la modélisation du contenu des documents CV des candidats en termes de leurs acquis ou requis.

Le domaine que va couvrir l'ontologie *Recruteur* correspond aux postes offerts par un recruteur (dans le cadre de l'enseignement supérieur) ainsi que les connaissances du recruteur concernant les différentes activités universitaires. Le but opérationnel de cette ontologie est la modélisation des offres d'emploi dans le cadre de l'enseignement supérieur en termes de besoins du recruteur, requis des enseignants et des connaissances sur les activités universitaires.

Acquisition des connaissances : Les sources d'informations sur lesquelles nous nous sommes basées pour arriver à la construction des deux ontologies sont les travaux antérieurs manipulant des ontologies pour le domaine du recrutement électronique ainsi que des documents techniques, l'étude de la structure des CV pour l'ontologie CV et l'étude de la structure des offres d'emploi pour l'ontologie *Recruteur*. Afin de structurer et représenter les informations trouvées, nous avons élaboré un tableau récapitulatif (voir tableau 6.1) qui contient le terme associé à tous les concepts pertinents, relatifs au domaine, ainsi que sa description.

Concepts Pertinents	Diplôme	Diplômes obtenus par un enseignant-chercheur ;
	Grade	Degré dans la hiérarchie des études supérieures ;
	Etat civil	Informations personnelles d'un candidat ;
	Activité scientifique	Événement scientifique ou missions ou stages ;
	Article	Publié dans périodiques, présenté dans des communications ;
	Publication	Manière de publier un écrit d'enseignement dans un ouvrage, ou article ou présenté dans des communications ;
	Événement scientifique	Événement qui permet la transmission du savoir scientifique entre les individus
	Communication	Forme de discussion sur un sujet donné entre des chercheurs ;
	CV	Curriculum vitae;
	Carrière Professionnelle	Etant soit une carrière pédagogique ou scientifique ;
Carrière scientifique	Représente les postes et fonctions occupées durant sa carrière professionnelle ;	

TAB. 6.1 -- Tableau récapitulatif



TAB. 6.2 – Diagramme de classes uml

Ainsi, notre tableau va nous permettre de retrouver toutes les informations pertinentes et essentielles à l’élaboration de notre ontologie. On fera remarquer que les descriptions sont approximatives. Elles ne sont pas définitives. Une fois les concepts et les relations identifiées par leurs termes, nous allons décrire leurs sémantiques dans un langage semi-formel tel que UML (voir tableau 6.2).

Formalisation des connaissances : Dans notre travail, la modélisation de l'ontologie est basée sur le diagramme de classes, vu qu'il existe une grande ressemblance entre son architecture et celle de l'ontologie. Dans ce qui suit, nous considérons :

- ☞ Les classes pour représenter les concepts.
- ☞ Les attributs pour représenter les caractéristiques du concept.
- ☞ Les relations d'associations pour représenter les relations entre concepts.
- ☞ Les relations d'héritages pour représenter les relations de subsomption.

Construction du dictionnaire de relations: Dans cette partie, nous allons nous concentrer sur la description des relations. Pour chaque relation déjà trouvée dans la partie précédente, nous allons lui préciser sa relation inverse, ses caractéristiques (transitivité, symétrie, fonctionnalité et fonctionnalité inverse) ainsi que ses sous relations (si elles existent).

Définition formelle des concepts: Ici, nous définissons les concepts relatifs à notre domaine en utilisant les constructeurs fournis par les logiques de descriptions pour faciliter leurs représentations ultérieures dans un langage complètement formel et opérationnel.

Voici un extrait des définitions des concepts de l'ontologie *Cv* :

- ✚ Publication \supseteq Article \cup Ouvrage \cup Communication
- ✚ Article \cap Ouvrage \cap Communication $\equiv \perp$
- ✚ Article $\subseteq \leq 1$ EstPubliéDans.Periodique
- ✚ ActivitésScientifique \supseteq Evenement_Sc \cup Mission_Stage
- ✚ Evenement_Sc \cap Mission_Stage $\equiv \perp$
- ✚ Communication $\subseteq \exists$ EstPresentéDans.Eventement_Sc
- ✚ CarrièreProfessionnelle \supseteq Carrière_Sc \cup CarrièrePedagogique
- ✚ Carrière_Sc \cap CarrièrePedagogique $\equiv \perp$
- ✚ CursusProfessionnel $\equiv \exists$ ContientCarriereProf.CarrièreProfessionnelle \cap
 \exists ContientPublication.Publication $\cap \exists$ ContientActivitéSc.ActivitésScientifique
- ✚ Cv \equiv ContienEtatCivil.EtatCivil \cap ContientCurcus.CursusProfessionnel \cap
 ContientDiplome.Diplôme \cap ContientGrade.Grade \cap ContientCompétance.Compétence
- ✚ ...

Voici quelques définitions des concepts de l'ontologie *Recruteur* en logique de description :

- ✚ Activités_Universitaire \supseteq Conférence \cup Publication \cup Stage
- ✚ Publication \supseteq Publication_Scientifique \cup Publication_Pédagogique

- ✚ Publication_Scientifique \supseteq Article_Scientifique \cup Communication
- ✚ Publication_Scientifique $\subseteq \leq 1$ EstPubliéDans.Periodique
- ✚ Article_Scientifique $\subseteq \exists$ EstPresentéDans.Communication
- ✚ Communication $\subseteq \exists$ EstPresentéDans.Conférence
- ✚ Offre \equiv Contien.PosteRecherché \cap Exige.InformationsPersonnelles \cap Exige.Diplôme
- ✚ ...

6.4.3 Le système de dialogue argumentatif

Dans ce qui suit, nous présentons la définition des composants de base de notre système:

L'ensemble des agents impliqués dans le dialogue: Nous supposons que l'Agent CV a déjà sélectionné un CV et l'a envoyé à l'agent *Recruteur* et nous nous intéressons à la communication entre l'agent CV et l'agent *Recruteur* en cas de conflit ontologique. En effet, lors de l'évaluation du CV sélectionné par l'agent CV, l'agent *Recruteur* peut trouver des concepts qui n'appartiennent pas à sa base de connaissances car les deux agents sont hétérogènes et ils ne partagent pas la même ontologie.

Les règles de commencement du dialogue: Le dialogue commence lorsque l'agent recruteur envoie son offre à l'agent CV.

Le but du dialogue: Le but de notre dialogue est que les deux agents arrivent à un consensus sur les conflits ontologiques pour satisfaire les besoins du recruteur.

Le rôle de chaque agent dans le dialogue: on doit associer un rôle à chaque agent.

☞ **Les rôles associés à l'agent Recruteur :**

- Poser des questions lors d'un conflit.
- Argumenter ou défier les arguments de l'agent CV.
- Inférer de nouvelles connaissances.
- Accepter ou refuser les arguments de l'agent CV.

☞ **Les rôles associés à l'agent CV :**

- Proposer les CV sélectionnés.
- Répondre aux questions de l'agent recruteur.
- Argumenter ou défier les arguments de l'agent recruteur.
- Accepter ou refuser les arguments de l'agent recruteur.

La base de connaissances : dans notre cas, l'agent *CV* a comme base de connaissances l'ontologie *CV* et l'agent *Recruteur* a comme base de connaissance l'ontologie recruteur.

Un langage de communication d'agents : dans un dialogue argumentatif entre des agents, la syntaxe du langage de communication utilisé revient à définir l'ensemble des performatives des actes de langages qui sont autorisés durant le dialogue. Nous définissons les principales performatives autorisés dans notre dialogue :

- **Informer (ϕ)** : Un agent *S* informe un agent *H* d'une proposition ϕ .
- **Propose (ϕ)** : Faire une proposition ϕ .
- **Question (ϕ)** : Poser une question ϕ .
- **Argumenter (ϕ_i, ϕ_j)** : Justifier ϕ_i par une proposition ϕ_j .
- **Défier (ϕ_i, ϕ_j)** : Remettre en cause la définition de ϕ_i par la proposition ϕ_j .
- **Accepter (ϕ)** : Accepter un argument ou une offre.
- **Refuser (ϕ)** : Refuser un argument ou une offre.

Comportement de chaque agent dans le dialogue : Afin de concevoir un agent cognitif, il faut penser à définir ses comportements. A cette fin, une base des règles (règles de réception, règles d'envoi,...) est conçue.

La base de règles est modélisée à l'aide du diagramme d'activité d'UML. La modélisation des comportements (ou behaviors) de l'agent *CV* (respectivement l'agent *Recruteur*) est représentée par le diagramme d'activité de la figure 6.3. (respectivement la figure 6.4).

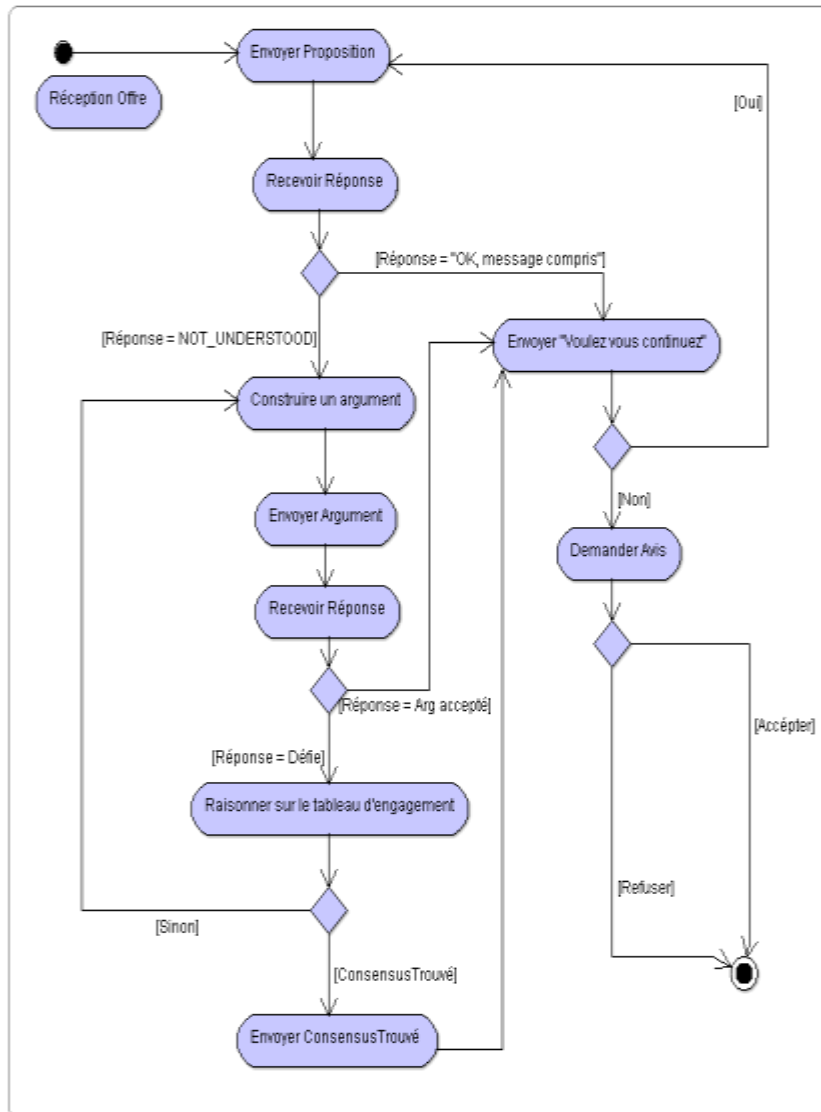


FIG. 6.3 -- Le diagramme d'activité de l'agent CV

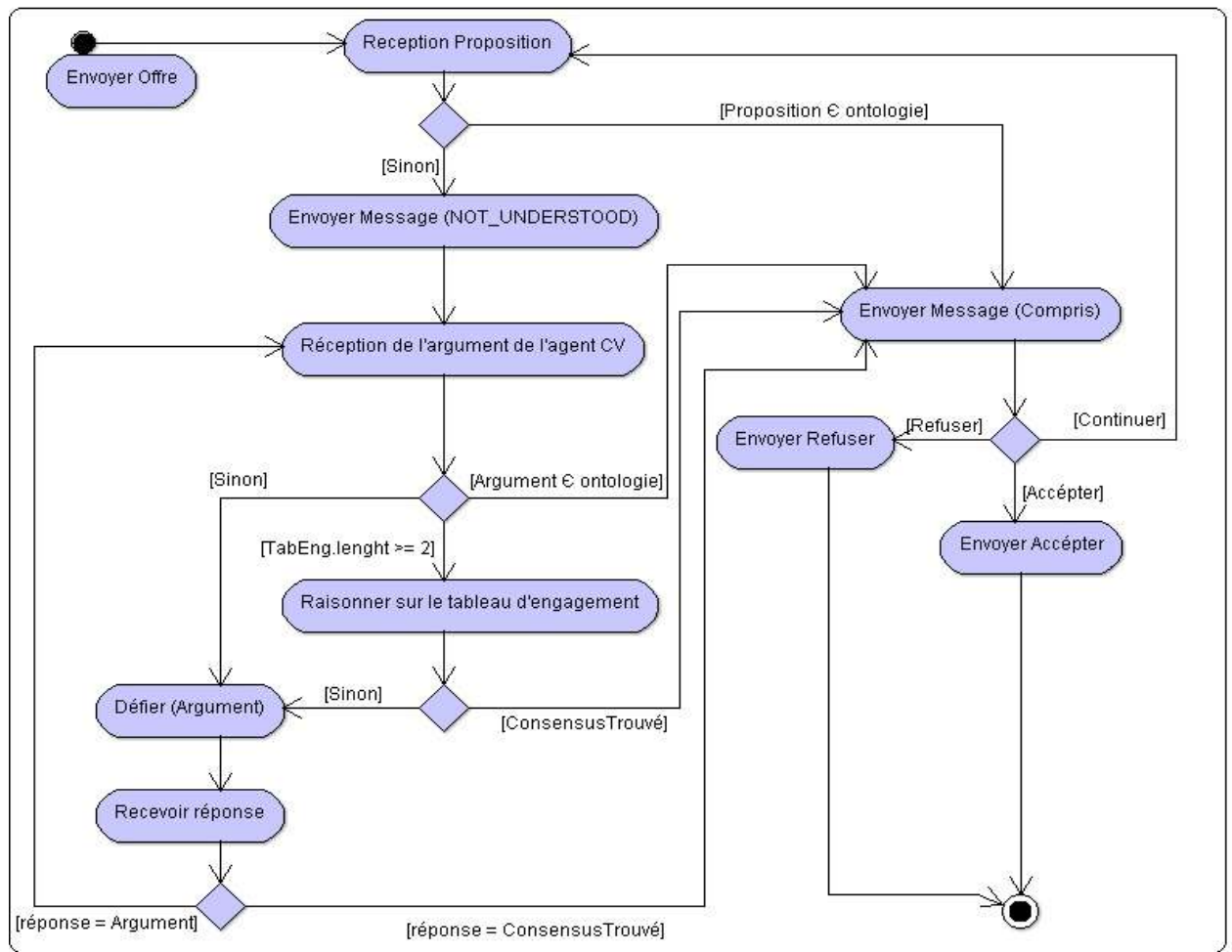


FIG. 6.4 -- Le diagramme d'activité de recruteur

Le protocole de dialogue : Il détermine les actes autorisés à chaque moment du processus de dialogue. Le protocole de dialogue proposé est représenté dans le diagramme de séquence UML suivant :

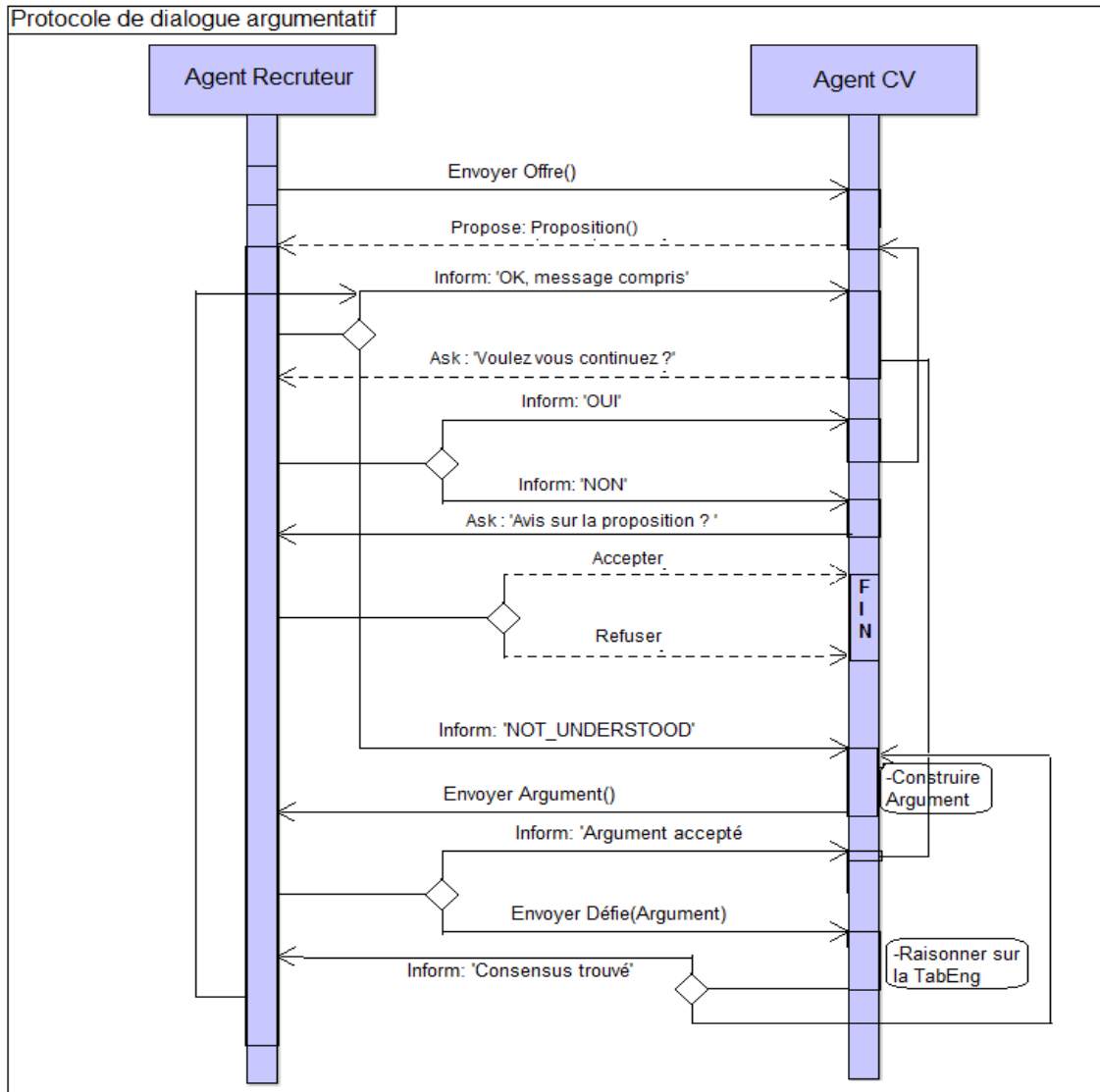


FIG. 6.5 -- Diagramme de séquence UML

Afin de concrétiser le protocole du dialogue argumentatif, nous présentons ci dessous, un exemple en langage naturel:

- ✚ **L'agent CV** : l'enseignant a présenté un article dans un événement scientifique.
 - ✚ **L'agent Recruteur** : Selon moi, un article est présenté dans une communication.
 - ✚ **L'agent CV** : Une communication est présenté dans un événement scientifique.
 - ✚ **L'agent Recruteur** : Selon moi, une communication est présenté dans une conférence.
 - ✚ **L'agent CV** : Donc, une conférence est un événement scientifique.
 - ✚ **L'agent Recruteur** : OK, alors un article est présenté dans une conférence.
- /* Consensus trouvé */*

6.5 Réalisation du système

Nous avons choisis la plate-forme Multi-agent Jade, dans laquelle l'ontologie, codifiée en OWL, est vue comme un ensemble de concepts, de prédicat et d'actions. Une réorganisation totale des ontologies était donc nécessaire. Pour l'implémentation des agents, des comportements doivent être associés. Nous avons utilisé un comportement par tâche unitaire: recevoir un message, envoyer un message, faire un traitement. Le langage SL pour exprimer le contenu des messages où l'expression du contenu est une chaîne de caractères lisible pour l'homme, facile à comprendre et à manipuler, a été utilisée.

Une interface, dans laquelle on peut voir l'offre envoyée par l'agent recruteur et le CV sélectionnée par l'agent CV, s'affiche (voir figure 6.6). Elle permet aussi de déclencher le dialogue argumentatif.

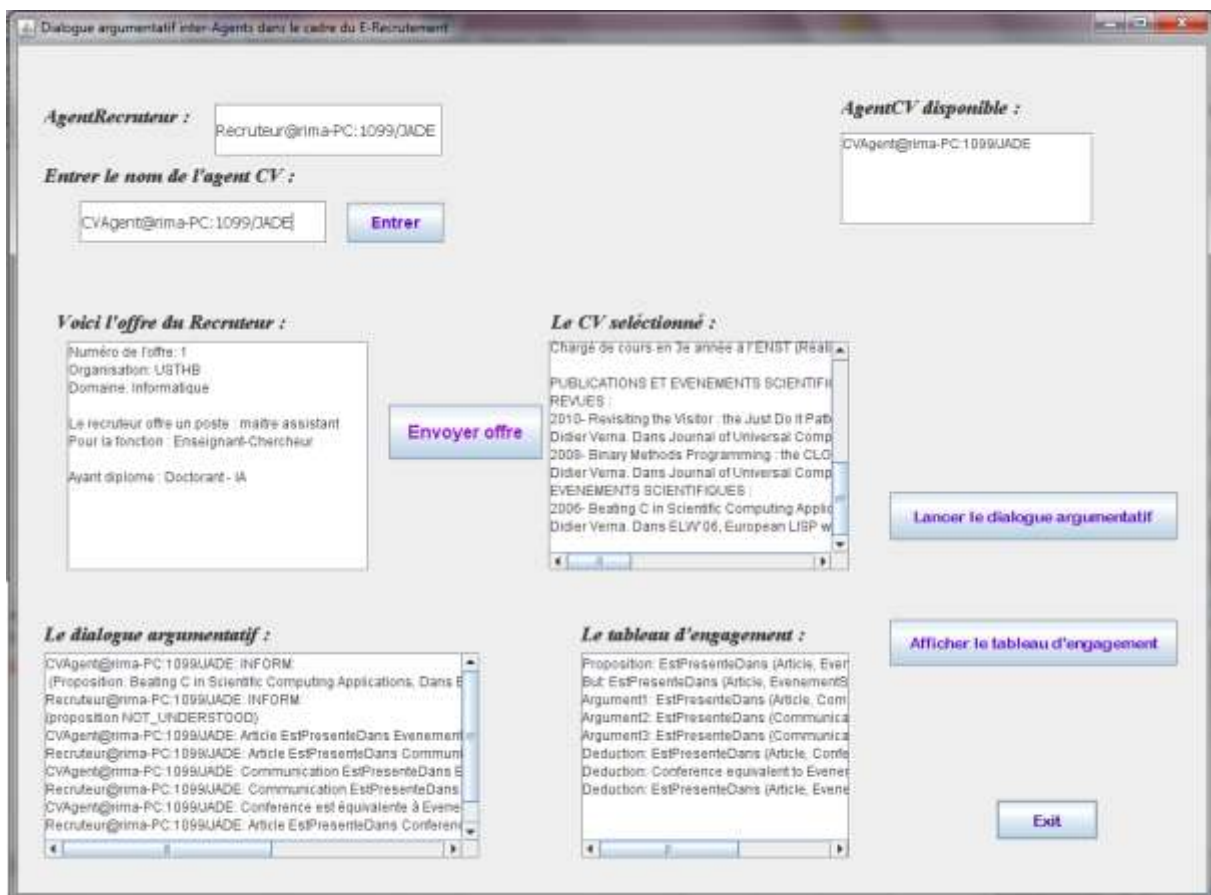
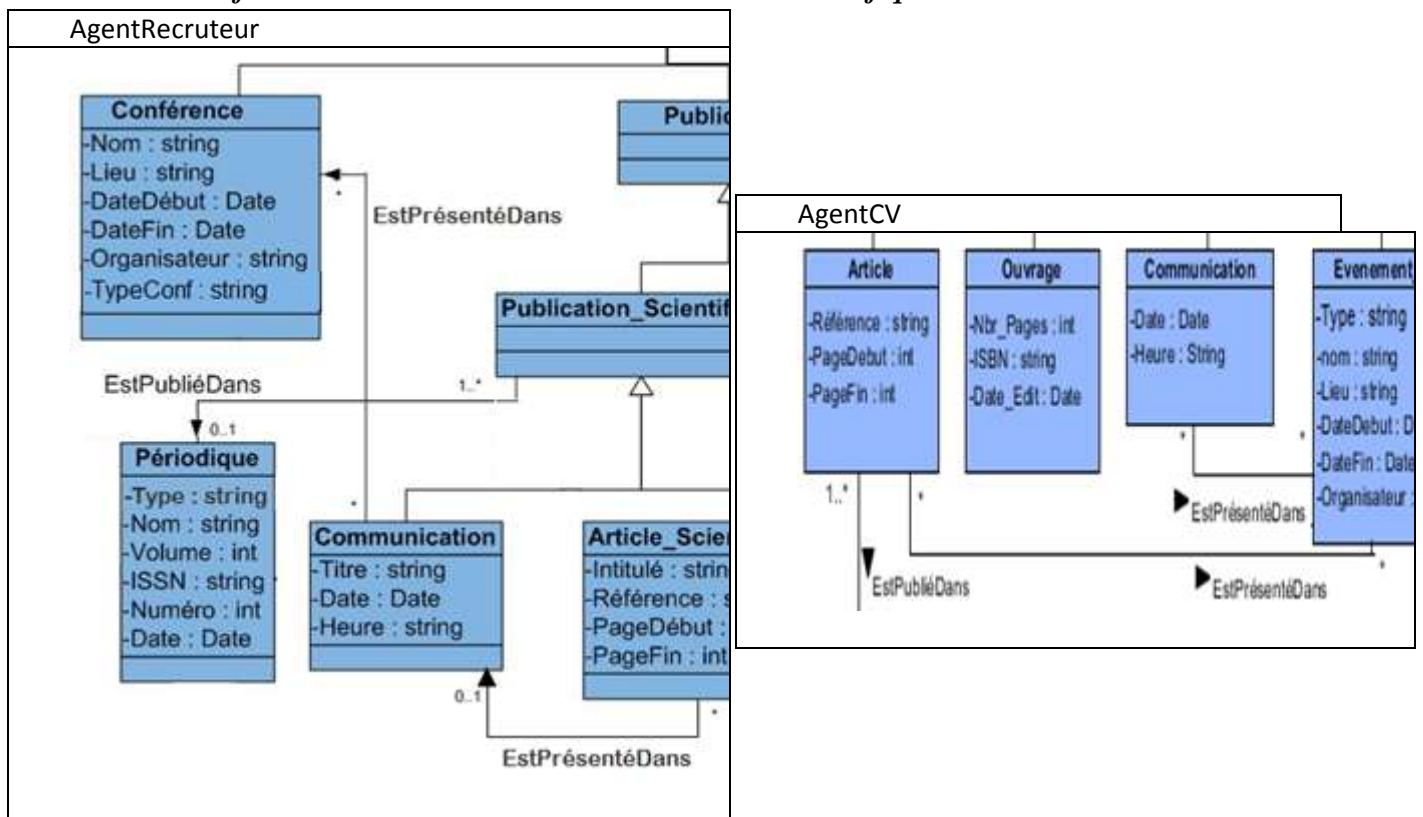


FIG. 6.6 – Interface de communication entre agent

Un certain nombre de conflit est recensé. Afin de les présenter pour montrer les types de conflit que notre système peut régler, l'accent est mis sur les parties de l'ontologie

(diagramme de classes) sur lesquelles porte le dialogue. Ensuite, le dialogue argumentatif déclenché par notre système, en spécifiant l'engagement ontologique de chaque agent, est affiché. Mais avant, pour une meilleure compréhension, un dialogue en langage naturel est présenté.

A/ Conflit : Article EstPresenteDans EvenementScientifique



Le dialogue en langage naturel :

- ✚ **L'agent CV** : l'enseignant a présenté un article dans un événement scientifique.
- ✚ **L'agent Recruteur** : selon moi, un article est présenté dans une communication.
- ✚ **L'agent CV** : une communication est présenté dans un événement scientifique.
- ✚ **L'agent Recruteur** : selon moi, une communication est présenté dans une conférence.
- ✚ **L'agent CV** : donc, une conférence est un événement scientifique.
- ✚ **L'agent Recruteur** : OK, alors un article est présenté dans une conférence.

/ Consensus trouvé */*

Le dialogue argumentatif :

CVAgent@rima-PC:1099/JADE: CVSelectionné Envoyé
Recruteur@rima-PC:1099/JADE: NOT_UNDERSTOOD
CVAgent@rima-PC:1099/JADE: What ?
Recruteur@rima-PC:1099/JADE[Request]: Article EstPresenteDans
CVAgent@rima-PC:1099/JADE[Conflit]: Article EstPresenteDans EvenementScientifique
Recruteur@rima-PC:1099/JADE : Article EstPresenteDans Communication
CVAgent@rima-PC:1099/JADE : Communication EstPresenteDans EvenementScientifique
Recruteur@rima-PC:1099/JADE : Communication EstPresenteDans Conférence
CVAgent@rima-PC:1099/JADE : Article EstPresenteDans EvenementScientifique
CVAgent@rima-PC:1099/JADE : EvenementScientifique est equivalent à Conférence
CVAgent@rima-PC:1099/JADE : Article EstPresenteDans EvenementScientifique
CONSENSUS TROUVE

Le tableau d'engagement :

But: EstPresenteDans (Article, EvenementScientifique)
Argument1 : EstPresenteDans (Article, EvenementScientifique)
Argument2 : EstPresenteDans (Article, Communication)
Argument3 : EstPresenteDans (Communication, EvenementScientifique)
Argument4 : EstPresenteDans (Communication, Conférence)
Argument5 : EstPresenteDans (Article, EvenementScientifique)
Argument6 : EstPresenteDans (Communication, EvenementScientifique)
Argument7 : EstPresenteDans (Communication, Conférence)
Argument8 : EstPresenteDans (Article, EvenementScientifique)

B/ Conflit : Communication EstPresenteDans EvenementScientifique

Voir le diagramme de classes présenté précédemment, pour ce conflit.

Le dialogue en langage naturel :

✚ ***L'agent CV*** : l'enseignant a présenté une communication dans un événement scientifique.

✚ ***L'agent Recruteur*** : selon moi, une communication est présenté dans une conférence.

✚ ***L'agent CV*** : donc, une conférence est un événement scientifique.

✚ ***L'agent Recruteur*** : OK, alors une communication est présenté dans une conférence.

/ Consensus trouvé */*

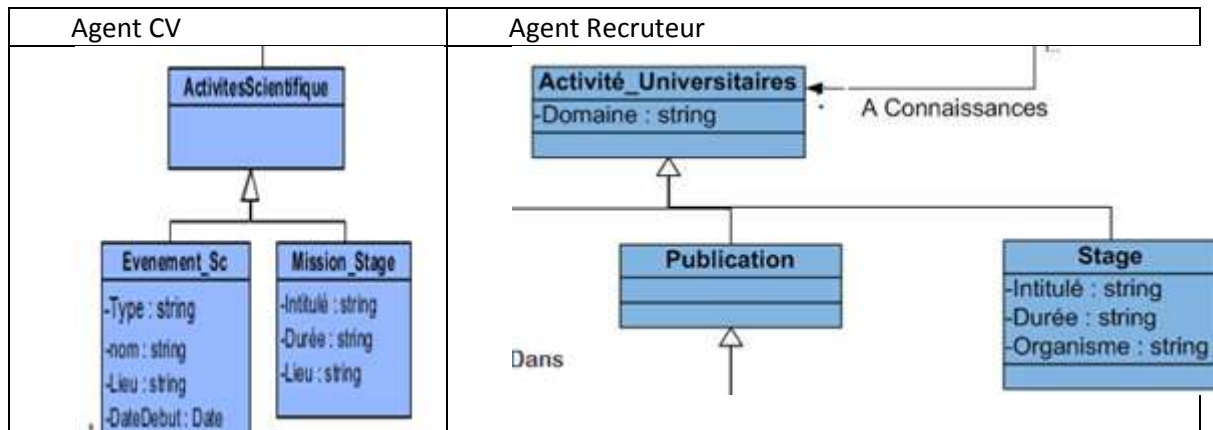
Le dialogue argumentatif :

CVAgent@rima-PC:1099/JADE[Conflit]: Communication EstPresente
Recruteur@rima-PC:1099/JADE : Communication EstPresenteDans Conférence
CVAgent@rima-PC:1099/JADE : EvenementScientifique est equivalent à Conférence
CONSENSUS TROUVE

Le tableau d'engagement :

But: EstPresenteDans (Communication, EvenementScientifique)
Argument1 : EstPresenteDans (Article, EvenementScientifique)
Argument2 : EstPresenteDans (Article, Communication)
Argument3 : EstPresenteDans (Communication, EvenementScientifique)
Argument4 : EstPresenteDans (Communication, Conférence)
Argument5 : EstPresenteDans (Article, EvenementScientifique)
Argument6 : EstPresenteDans (Communication, EvenementScientifique)
Argument7 : EstPresenteDans (Communication, Conférence)

C/ Conflit : Mission_Stage



Le dialogue en langage naturel :

- + Agent Recruteur : Qu'est ce qu'une Mission_Stage ?
- + AgentCV : Mission_Stage est une Activité Scientifique.
- + Agent Recruteur : Selon moi, Activité Scientifique est équivalente à Activité Universitaire.
- + AgentRecruteur : Alors, Mission_Stage est une Activité Universitaire.

/* Consensus trouvé */

Le dialogue argumentatif :

```

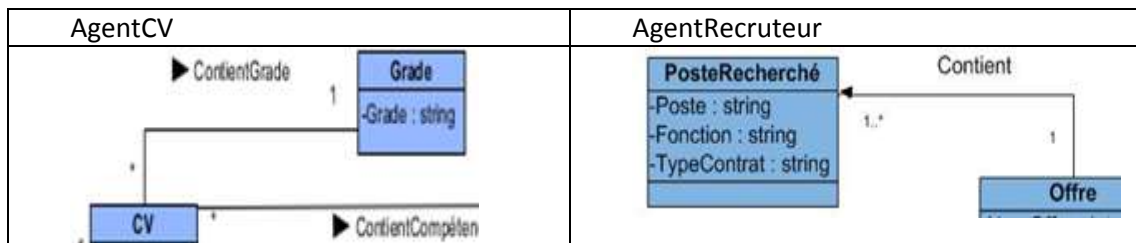
CVAgent@rima-PC:1099/JADE: What ?
Recruteur@rima-PC:1099/JADE: Mission_Stage
CVAgent@rima-PC:1099/JADE: Mission_Stage is a .Activite_Scientifique
Recruteur@rima-PC:1099/JADE: Activite_Scientifique est equivalente Activite_Universitaire
Consensus Trouvé
          
```

Le tableau d'engagement :

```

But: Mission_Stage
Argument1: (Mission_Stage, Activite_Scientifique)
Argument2: (Activite_Scientifique, Activite_Universitaire)
          
```

D/ Conflit : Le concept Grade n'existe pas dans l'ontologie Recruteur



Le dialogue en langage naturel :

- ✚ Agent Recruteur : Qu'est ce qu'un Grade ?
- ✚ AgentCV : Le grade de l'enseignant est Maître Assistant.
- ✚ Agent Recruteur : Selon moi, Maitre Assistant est le poste recherché.
- ✚ AgentCV : Alors, un Grade est équivalent au poste recherché

/ Consensus trouvé */*

The screenshot displays two side-by-side windows. The left window, titled "Le dialogue argumentatif :", contains a chat log with the following messages:
Recruteur@rima-PC:1099/JADE: NOT_UNDERSTOOD
CVAgent@rima-PC:1099/JADE: What ?
Recruteur@rima-PC:1099/JADE: Grade
CVAgent@rima-PC:1099/JADE: Grade = Maitre Assistant
Recruteur@rima-PC:1099/JADE: Maitre Assistant = Poste Recherché
Recruteur@rima-PC:1099/JADE: Grade est equivalent Poste Recherché

The right window, titled "Le tableau d'engagement :", contains a table of commitments:
But: Grade
Argument1: (Grade, Maitre Assistant)
Argument2: (Maitre Assistant, Poste Recherché)
Argument3: (Grade, Poste Recherché)

6.6 Conclusion

Nous nous sommes intéressées dans ce travail à la communication inter-agents dans un SMA ouvert appliqué au recrutement électronique. Nous avons vu, dans ce contexte, que deux agents ne partagent pas forcément la même définition d'un objet, soit parce que l'un d'eux est ignorant soit parce que leurs définitions ne proposent pas le même point de vue (du aux ontologies différentes). C'est pour résoudre ce type de problème que nous avons proposé et implémenté un protocole de dialogue basé sur l'argumentation.

Différents types de conflits ont été réglés. Cependant, nous ne pouvons affirmer à travers ces exemples que notre système est complet. En effet, notre proposition est toujours en cours de tests. Ce qui est aussi intéressant, c'est d'autoriser d'autres performatives dans le langage de communication, pour prendre en compte les engagements conditionnels.

Conclusion Générale et Perspectives

Définie pour un objectif donné, l'ontologie exprime un point de vue partagé par une communauté. Elle s'exprime dans un langage et repose sur une théorie (sémantique) assurant des propriétés en termes de *consensus*, *cohérence*, *réutilisation* et *partage* (Charlet 2002). Un intérêt particulier lui est donc consacré vu l'espace de recherche de plus en plus important qu'elle occupe. Elle devient incontournable tant ses domaines d'application sont variés. Elle est utilisée pour décrire et traiter des ressources multimédia ; asseoir l'interopérabilité d'applications en réseaux ; piloter des traitements automatiques de la langue naturelle ; construire des solutions multilingues et interculturelles ; permettre l'intégration de sources hétérogènes d'information ; décrire des protocoles d'interactions complexes ; permettre les raisonnements temporel et spatial ; faire des approximations logiques ; etc. (Gandon 2006). Ceci est réalisé grâce aux deux rôles principaux qui leurs sont alloués :

- **Les connaissances du domaine.** L'ontologie est surtout utilisée pour la représentation de connaissances et l'application de raisonnements sur ces dernières. Elle décrit les objets, leurs propriétés et la façon dont ils peuvent se combiner constituant ainsi les connaissances du domaine.

- **La communication.** L'ontologie, décrite comme un vocabulaire consensuel, est utilisée pour harmoniser la communication entre différentes applications ou entre différents agents (humains-machines). Cependant, le succès des ontologies, a fait qu'un engouement certain pour ce domaine, a suscité des travaux où des ontologies développées séparément par des organisations et/ou des personnes différentes, même si elles se rapportent au même domaine, ou a des domaines ayant une intersection non vide, sont souvent hétérogènes, à la fois au niveau de leur structure et au niveau de la sémantique des données qu'elles contiennent. Toute la question est de pouvoir les exploiter simultanément. Ceci soulève le

problème d'interopérabilité sémantique dans le cas de la communication et l'échange d'information.

Dans ce mémoire de thèse, après avoir mis l'accent sur le rôle de l'ontologie et l'importance qu'elle revêt dans la représentation de connaissances, nos contributions, aussi modeste soient-elles à la recherche dans ce domaine – ontologie et représentation de connaissances d'une part et problème d'hétérogénéité des ontologies d'autre part- sont présentées. Ceci est réalisé dans le contexte du « web sémantique », paradigme de réflexion en parfait accord avec les points soulevés dans cette thèse, et dans lequel se dégage un système symbolique où deux notions se distinguent :

- L'ontologie: qui est une spécification explicite d'une conceptualisation fournissant le - vocabulaire consensuel;
- La base d'annotations: qui est constituée de descriptions de faits ou de ressources que l'on formule avec le vocabulaire de l'ontologie.

Notre travail a été subdivisé en deux parties. La première partie, d'un point de vue architectural, intervient dans la couche la plus basse du web sémantique à savoir les métadonnées. Ces dernières permettent d'assurer l'interopérabilité entre les différents formalismes de représentation et ontologies.

Après avoir motivé l'utilisation d'un formalisme à base de graphes, en l'occurrence, les graphes conceptuels de Sowa, nous nous sommes intéressés à RDF(S) langage d'annotations et de métadonnées du web sémantique. Nous avons montré les points communs et les différences entre les deux modèles de représentation puis un ensemble de règles en a été extrait permettant le passage automatique entre les deux modèles. Leur validation a été faite dans le cadre d'un travail de magister (Guebaili 2006).

Afin d'appliquer ces règles, nous nous sommes intéressé au web sémantique géo spatial pour justifier la prise en compte de l'aspect d'incertitude et d'imprécision de la connaissance. Nous avons montré que des normes de métadonnées, tiennent compte de l'aspect qualité de la donnée géo spatiale. Cette dernière peut être formalisée par une ontologie où certains concepts peuvent être de nature imparfaite

Dans la deuxième partie, on s'est consacré au problème d'hétérogénéité des ontologies. Pour cela, nous avons orienté nos travaux vers l'alignement des ontologies, en proposant un processus d'appariement. Mais avant, une classification des techniques d'appariement de base

en partant des niveaux de complexité sémantique associés aux différents langages du web sémantique a été définie. Elle fournit un guide d'utilisation pour appairer les ontologies dans le Web sémantique.

Dans un premier temps, un système d'alignement d'ontologies basé sur les mesures de similarité, en utilisant la technologie agent a été réalisé. Trois (3) agents ont été implémentés, dédiés respectivement à la mesure syntaxique –JaroWinkler-, la mesure lexicale –WordNet-, et une mesure structurelle, dont le calcul est basé sur la relation de voisinage -père, frère, fils- dans la structure de l'ontologie. Pour prendre la meilleure décision, et donc le meilleur alignement, un système d'argumentation basé sur les préférences est déclenché. Le système a été validé sur des Benchmarks.

Cependant, il arrive que des agents ne partagent pas forcément la même définition d'un concept. Et donc, le système précédent ne permet pas de résoudre ce type de conflit. D'où, la suite à ce travail qui consiste à proposer et implémenter un protocole de dialogue basé sur l'argumentation. Au terme du dialogue qui les confronte, les agents doivent atteindre un accord qui concerne la définition du concept et l'inférer par la suite. Ceci est réalisé par une justification argumentative entre agents où la description de l'objet est extraite de leur base de connaissances (l'ontologie). Nous avons appliqué ce dialogue dans le contexte d'une application de e-recrutement. Plusieurs tests sont en cours de réalisation pour détecter les conflits éventuel que notre système peut régler.

Perspectives

A l'issue de ce travail de thèse, plusieurs points restent en suspens et d'autres problèmes restent encore à développer. Nous présentons ici ce qui nous semble être le cadre d'investigation et où des avancées importantes sont tout à fait envisageables :

- Des langages de requêtes tels que RDQL (Seaborne, 2004) et SPARQL (Prud'hommeaux, 2006) peuvent être employés pour répondre à des requêtes utilisateurs en tenant compte de cet aspect d'imperfection de la connaissance. Une étude plus approfondie de ces langages ou même une proposition d'un langage de requête plus adéquat fera l'objet de travaux futurs.

- Le web est en train de subir des changements particulièrement importants, comme l'augmentation du nombre d'applications en ligne, les réseaux mobiles et ubiquitaires, l'augmentation du nombre de services, etc., fait qu'on doit approfondir nos recherches et voir quel rôle on doit allouer à l'ontologie dans ce cas.

- En ce qui concerne le web sémantique géo spatiale, il serait intéressant de se concentrer sur une application géo spatiale spécifique, visant à introduire l'imprécision et l'incertitude dans les éléments de qualité des données. En outre, comme les utilisateurs peuvent donner des noms différents à des entités ayant la même signification, la solution pourrait consister à élaborer une ontologie de la qualité qui décrit formellement les termes et les relations. On peut aussi considérer les profils utilisateurs comme dans le projet REVIGIS (Frank et al, 2004; Grum et Vasseur, 2004) dans un système de décision, en tenant compte de la notion de distance approximative et requêtes floues (Buche et al., 2008). Cela pourrait être accompli en nous positionnant sur les travaux de projection approximative (Thomopoulos et al., 2003) dans les graphes conceptuels.

- Pour soutenir l'importance du système d'alignement réalisé, il serait bon de l'intégrer dans le cadre de travaux relatifs au web service sémantique.

- Pour la dernière partie, la généralisation du protocole dialogique argumentatif entre plusieurs agents, ainsi que l'enrichissement de la base d'inférences pour améliorer le moteur de raisonnement argumentatif développé, pour plus de performances seraient une perspective à ce travail.

- Les ontologies ont été construites à l'aide du langage OWL-DL mais pour la communication entre agents c'est le langage FIPA SL pour coder les messages qui a été utilisé. L'idéal serait de coder même les messages avec le langage OWL, le standard de définition des ontologies pour le Web sémantique.

Bibliographie

- (Aalders et Morrison, 1998) Aalders H., Morrison J., Spatial Data Quality for GIS. *Geographic Information Research: Trans-Atlantic Perspectives*, Craglia M., Onsrud H. (dir.), pp. 463-475, Taylor & Francis, London, Bristol, 1998.
- (Aalders, 1999) Aalders, H., The registration of quality in a GIS. *Proceedings of the 1st International Symposium on Spatial Data Quality*, 18–20 July, Hong-Kong, pp.23–32, 1999.
- (Akli, 2005) Akli_Astouati K., Assistance Tool for Conflicts Resolution by Technical Negotiation. Proceedings of the first Edition of the Information and Communication Technologies International Symposium (ICTIS 2005), June, Tetuan, Morocco, pp. 188-194, Essaidi, M., Raissouni, N. (eds.), ISBN : 9954-8577-0-2.
- (Akli, 2006) Akli_Astouati K., Assistance Tool for Conflicts Resolution through Technical Negotiation. In *International Journal of Computer Science and Applications (IJCSA 2006)*; vol. 3, n°2, June, ISSN : 0972-9038.
- (Akli, & al., 2010) Akli_Astouati K., Guebaili R., Mokhtari, A., Geographic metadata : data quality, uncertainty , imprecision. In *International Conference on Web and Information Technologies. -IEEE-* 16-19 June 2010, Marrakech. Morocco.
- (Akli, & al., 2011) Akli-Astouati K., Guebaili-Djider R., Mokhtari A., Geographic Metadata: Data quality, uncertainty and imprecision. In *International Journal of Reasoning-based Intelligent Systems- by Inderscience Publishers (IJRIS 2011)*; vol. 3, no.3/4 , 2011. ISSN : 0972-9038.
- (Amgoud, 1999) Amgoud, L., Contribution à l'intégration des préférences dans le raisonnement argumentatif. Thèse de Doctorat, Université de Paul Sabatier, Toulouse, 1999.

- (Amgoud, & al., 2000) Amgoud, L., Parsons S., Perrussel, L., An argumentation framework based on contextual preferences. In *Proceedings of the International Conference on Formal and Applied and Practical Reasoning*, pp. 59–67, 2000.
- (Atencia et Schorlemmer, 2008) Atencia, M., Schorlemmer, M., Formalising interaction-situated semantic alignment: The communication product. In *Tenth International Symposium on Artificial Intelligence and Mathematics (ISAIM'08)*, Fort Lauderdale, Florida, USA, 2008.
- (Aussenac-Gilles & al., 1992) Aussenac-Gilles N., Krivine J.-P., Sallantin J., Editorial. L'acquisition de connaissances pour les systèmes à base de connaissances. In *Revue d'Intelligence Artificielle*, vol., 6, n°1.2, pp. 7–18, 1992.
- (Bach, 2006) Bach, T.-L., (2006), *Construction d'un Web sémantique multi-points de vue*. Thèse de doctorat, École des Mines de Nice à Sophia Antipolis, 2006.
- (Bachimont, 1996) Bachimont, B., Herméneutique matérielle et Artéfacture : des machines qui pensent aux machines qui donnent à penser ; Critique du formalisme en intelligence artificielle. Thèse de doctorat d'épistémologie, École Polytechnique, 1996.
- (Bachimont, 2007) Bachimont, B., *Ingénierie des connaissances et des contenus- le numérique entre ontologies et documents*, collection Science informatique et SHS dirigée par C. Garbay, Hermès Lavoisier, 2007, ISBN, 978-746213.
- (Baget & al., 2010) Baget, J.-F., Croitoru, M., Gutierrez, A., Leclère, Mugnier, M.-L., *Translations between RDF(S) and Conceptual Graphs*. In *Proceedings of the 18th International Conference on Conceptual Structures (ICCS 2010)*, vol. 6208 of LNCS, pp. 28-41, 2010, Springer.
- (Bailin et Truszkowski, 2001) Bailin, S.-C., Truszkowski, W., Ontology Negotiation as a Basis for Opportunistic Cooperation between intelligent information agents. In *Proceedings of the Cooperative Information Agents, 5th International Workshop, Modena, Italy, September 6-8*, pp. 223-228, 2001.
- (Bailin et Truszkowski, 2002) Bailin, S.-C., Truszkowski, W., Ontology negotiation: How agents can really get to know each other. In *Proceedings of the Workshop on Radical Agent Concept*, pp. 320–334, 2002.
- (Baneyx, 2007) Baneyx, A., *Construire une ontologie de la pneumologie, aspects théoriques, modèles et expérimentations*, Thèse de doctorat, Université Paris 6, 2007.
- (Barr et Feigenbaum, 1982) Barr, A., Feigenbaum, E., *Handbook of Artificial Intelligence*. Palo Alto, Pitman, 1982.

- (Batini, & al., 1986) Batini, C., Lenzerini, C., Navathe, S., A comparative analysis of methodologies for database schema integration. *ACM Computing Surveys*, vol. 18, no. 4, pp. 323–364, 1986.
- (Bellahsene, & al., 2011) Bellahsene, Z., Bonifati, A., Rahm, E., Eds., *Schema Matching and Mapping*, 2011, Springer.
- (Bénel, 2011) Bénel, A., *D'où viennent les ontologies? À la recherche du chaînon manquant*. Manuscrit auteur, publié dans *Atelier Philosophie et Ingénierie. Le formel face à l'histoire, la technologie et la matérialité (IC'2011)*, Chambéry, France.
- (Béguin et Pumain, 2003) Béguin, M., Pumain, D., *La représentation des données géographiques - statistique et cartographie*, 2003, Cursus. Armand Colin.
- (Berengere & al., 2004) Berengere, V., Van de Vlag, D., Stein, A., Jeansoulin, R., Dilo, A., Spatio-temporal ontology for defining the quality of an application. *Proceedings of the 3rd International Symposium on Spatial Data Quality ISSDQ'04*, 15–17 April, Bruck an der Leitha, Austria, pp.67–81, 2004.
- (Berners-Lee, 2001) Berners-Lee, T., Conceptual graphs and the semantic web - reflections on web architecture, 2001, <http://www.w3.org/DesignIssues/CG.html>.
- (Berners-Lee & al., 2001) Berners-Lee, T., Hendler, J., Lassila, O., *The Semantic Web*. 2001, Scientific American.
- (Berners-Lee, 2002) Berners-Lee, T., *Prepare for Next-Gen Web Now*. 2002.
- (Berthier, 2004) Berthier, D., *Discordances ontologiques et questions d'interopérabilité*. colloque EHESS : sources et ressources pour les sciences sociales, 9-11 décembre 2004, Paris.
- (Bizer, & al., 2005) Bizer, C., Heese, R., Mochol, M., Oldakowski, R. Tolksdorf, R., Eckstein, R., The Impact of Semantic Web technologies on Job Recruitment Process. In *Proceedings of the 7th International Conference Wirtschaftsinformatik*, 2005.
- (Boin, 2008) Boin, A.-T., *Exposing Uncertainty: Communicating spatial data quality via the Internet*. Phd Thesis, Degree of Doctor of Philosophy, Department of Geomatics, the university of Melbourne, 2008.
- (Bordin, 2002) Bordin, P., *SIG - concepts, outils, et données*, 2002, Hermès.
- (Bos & al., 1997) Bos, C., Botella, B., Vanheeghe, P., Modeling and Simulating Human Behaviors with Conceptual Graphs. In *proceedings of the International Conference on Conceptual Structures*, vol. 1257 LNAI, pp. 275–289, 1997, Springer.

- (Bouchon-Meunier 1995) Bouchon–Meunier, B., La logique floue et ses applications, 1995, Edition Addison – Wesley.
- (Bouquet, & al., 2004) Bouquet, P., Ehrig, M., Euzenat, J., Franconi, E., Hitzler, P., Krotzsch, M., Serafini, L., Stamou, G., Sure, Y., Tessaris, S., *Specification of a common framework for characterizing alignment*. d2.2.1v2., Knowledge web, Universitat Karlsruhe, 2004.
- (Bource, & al., 2002) Bource, M., Harzallah, M., Leclère, M., Trichet, F., CommOnCV: Modeling the competencies underlying a Curriculum Vitae. Rapport de recherche, n° 02-2, 2002, IRIN, Université de Nantes.
- (Bourigault, & al., 2004) Bourigault, D., Aussenac-Gilles, N., Charlet, J., Construction de ressources terminologiques ou ontologiques à partir de textes : un cadre unificateur pour trois études de cas. In J. Pierrel & M. Slodzian, Coordinateurs, *Revue d’Intelligence Artificielle.*, vol. 18, pp. 87–110, 2004. Hermès. Numéro spécial sur les Techniques Informatiques et Structuration de Terminologies.
- (Boutouhami-Nouioua, 2010) Boutouhami-Nouioua, S., Un système de génération de descriptions argumentées, Thèse de Doctorat de l’Université Paris 13, 2010.
- (Brasoveanu, & al., 2010) Brasoveanu, A. et al., Generic Multimodal Ontologies for Human-Agent Interaction. *Int. J. of Computers, Communications & Control*, vol. V, no. 5, pp. 625-633, 2010.
- (Bratman, 1990) Bratman, M.-E., What is intention? In *Cohen, P.-R., Morgan, J.-L., Pollack, M.-E., editors*, intentions in Communication, pp. 15-32, 1990. The MIT Press, Cambridge, MA.
- (Buche, & al., 2008) Buche, P., Dibie-Barthelemy, J., Hignette, G., Flexible Querying of Fuzzy RDF Annotations Using Fuzzy Conceptual Graphs, *proceedings of the International Conference on Conceptual Structures*, LNAI n°5113, pp. 133–146, 2008. P. Eklund and O. Haemmerle (Eds.), Springer-Verlag Berlin Heidelberg.
- (Cao, 1999) Cao, T.-H., Foundations of Order-Sorted Fuzzy Set Logic Programming in Predicate Logic and Conceptual Graphs. PhD thesis, University of Queensland, Australia, 1999.
- (Cao, 2001) Cao T.-H., Fuzzy Conceptual Graphs for the Semantic Web. *International Workshop on Fuzzy Logic and the Internet*. 2001, Invited to the Berkeley Initiative in Soft Computing.
- (Cao et Wuwongse, 1996) Cao, T.-H., Wuwongse, W., Towards Fuzzy Conceptual Graph Programs. *Proceedings of the International Conference on Conceptual Structures*, LNAI, vol. 1115, pp. 263–276, 1996. Sydney, Australia, Springer–Verlag.

- (Carrillo-Ramos, 2007) Carrillo-Ramos, A.-C., *Agents ubiquitaires pour un accès adapté aux systèmes d'information: Le Framework PUMAS*. Thèse de doctorat, Université Joseph Fourier, Grenoble, 2007.
- (Chamoun, 2006) Chamoun, M., *Intégration de l'Internet 3G au sein d'une plateforme active*. Thèse de doctorat, Ecole Nationale Supérieure des Télécommunications de Paris, 2006.
- (Charlet, & al., 2000) Charlet, J., Reynaud, C., Bourgina, R., Ingénierie des connaissances pour les systèmes d'information, Chapitre 10, dans *Conception des Systèmes d'Information*, Ed. Cauvet, collection I2c – Paris : Hermès 2000.
- (Charlet, 2002) Charlet, J., *L'ingénierie des connaissances, Développement, Résultats et Perspectives pour la gestion des connaissances médicales*. Mémoire d'Habilitation à Diriger les Recherches – Université Pierre & Marie Curie, Paris, Informatique, 2002.
- (Chein et Mugnier, 1992) Chein, M., Mugnier, M.-L., Conceptual Graphs: Fundamental Notions. *Revue d'Intelligence Artificielle* vol.6, n°4, pp. 365-406, 1992.
- (Chuttur, 2011) Chuttur, M.-Y, Challenges Faced by Ontology Matching Techniques: Case Study of the OAEI Datasets, *Journal of Information Technology*, vol.3, n°1, pp.33-42, 2011.
- (Comber, & al., 2007) Comber, A.-J., Fisher, P.-F., Wadsworth, R.-A., User-focused metadata for spatial data, geographical information and data quality assessments. *10th AGILE International Conference on Geographic Information Science*, Aalborg University, Denmark, 2007.
- (Corby, & al., 2000) Corby, O., Dieng, R., Hebert, C., A conceptual graph model for w3c resource description framework, *proceedings of the international conference on conceptual structures*, LNCS 1867, Springer-Verlag, Darmstadt, Germany, pp. 468-482, 2000.
- (Corby et Faron 2002) Corby, O., Faron, C., Corese : A Corporate Semantic Web Engine, Proceedings of the International Workshop on Real World RDF and Semantic Web Applications, 11th International World Wide Web Conference, Hawaii, USA, May, 2002.
- (Cox et Cox 1994) Cox, T.-F., Cox, M.-A., Multidimensional Scaling. 1994, *Chapman & Hall* .
- (Cuenca-Grau, & al., 2005) Cuenca-Grau, E.-S.-B., Parsia B., Kalyanpur, A., Automatic partitioning of owl ontologies using e-connections. International Workshop on Description Logics (DL-2005).
- (Davidovsky, & al., 2010) Davidovsky, M., Ermolayev, V., Tolok, V., Evaluation of Semi-Automated Ontology Instance Migration. In M. Essaïdi et al. (Eds.), *Intelligent*

- Distributed Computing IV, SCI 315*, pp. 179–190, 2010, Springer-Verlag Berlin Heidelberg.
- (Davidovsky, & al., 2012) Davidovsky, M., Ermolayev, V., Tolok, V., A Survey on Agent-based Ontology Alignment, 4th International Conference on Agents and Artificial Intelligence (ICAART), 2, pp. 355-361, 2012.
- (Dhérent, 2005) Dhérent, C., Les métadonnées, à quoi ça sert ?, Journée d'information AFNOR CG 46, 7 juin 2005, Bibliothèque nationale de France.
- (Delteil, & al., 2003) Delteil, A., Faron, C., Dieng, R., Le modèle des graphes conceptuels pour le web sémantique. Extensions de RDF et RDFS basées sur le modèle des graphes conceptuel. *Revue des Sciences et Technologie de l'Information – l'Objet – 9/2003. XML et les objets*, 2003.
- (De Runz, 2008) De Runz, C., Imperfection, temps et espace: modélisation, analyse et visualisation dans un SIG archéologique. Thèse de doctorat en Informatique, Reims Champagne-Ardenne (URCA), 2008.
- (Do et Rahm, 2002) Do, H.-H., Rahm, E., Coma - a system for flexible combination of schema matching approaches. *Dans Proc. 28ième International Conference on Very Large Data Bases(VLDB)*, pp. 610–621, 2002, Hong Kong (CN).
- (Do, & al., 2003) Do, H.-H, Melnik, S., Rahm, E., Comparison of schema matching evaluations, A.B. Chaudhri et al. (Eds.): *Web Databases and Web Services*, LNCS 2593, pp. 221–237, 2003. Springer-Verlag Berlin Heidelberg.
- (Dongilli, & al., 2005) Dongilli, P., Fillottrani, P.-R., Franconi, E., Tessaris, S., A multi-agent system for querying heterogeneous data sources with ontologies. *Proc. Of Thirteenth Italian Symposium on Advanced Database Systems, SEBD-2005*.
- (Dubois et Prade, 2001), Dubois, D., Prade, H., La problématique scientifique du traitement de l'information. *Information-Interaction-Intelligence*, vol. 1, n°2, 2001, Cépaduès édition.
- (Duckham et Worboys, 2007). Duckham, M., Worboys, M., Automated geographic information fusion and ontology alignment. In Belussi, A., Catania, B, Clementini, E., and Ferrari, E. (Eds.), *Spatial data on the Web: Modelling and Management, Chapter 6*, pp. 109-132, 2007, Springer, Berlin.
- (Dung, 1995) Dung., P., On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n person games. *Artificial Intelligence*, vol.77, n°2, pp. 321–357, 1995.

- (Egenhofer, 2002) Egenhofer, M.-J., Toward the semantic geospatial web. *ACM International Symposium on Advances in Geographic Information Systems*, vol.1, pp. 34–43, 2002, McLean, VA.
- (Ehrig 2007) Ehrig, M., *Ontology alignment: bridging the semantic gap*. New-York (NY US): Semantic web and beyond: computing for human experience, 2007. Springer.
- (Erschler, & al., 1993) Erschler J., Fontan G., Mercé C., Approche par Contraintes en planification et ordonnancement de la production. *RAIRO-APII*, vol. 27, n° 6, pp. 669-695, 1993.
- (Euzenat, & al., 2004) Euzenat, J., Bach, T., Barrasa, J., Bouquet, P., Bo, J., Dieng-Kuntz, R., et al., *State of the art on ontology alignment*. 2004. Knowledge web, KWEB/2004/D2.2.3/v1.2.
- (Euzenat, & al., 2006) Euzenat, J., Laera, L., Tamma, V., Viollet, A., Negotiation and argumentation techniques among agents complying to different ontologies. 2006, *Deliverable D2.3.7*, KWEB, v1.0.
- (Euzenat et Shvaiko, 2007) Euzenat, J., Shvaiko, P., *Ontology matching*. 2007, Springer.
- (Euzenat, & al., 2007 a) Euzenat, J., Bach, T. L., Dieng, R., Duc, C. L., Napoli, A., Zimmermann, A., et al., *Knowledge web 2.2: Heterogeneity in the semantic web*. Technical Report, 2007, NoE Knowledge Web project.
- (Euzenat, & al., 2007 b) Euzenat, J., Mocan, A., Scharffe, F., *Ontology Management: Semantic Web, Semantic Web Services, and Business Applications*, ch. Ontology Alignments: an ontology management perspective, 2007, Springer.
- (Feigenbaum, 1984) Feigenbaum, E.-A., *Knowledge engineering: The Applied side of Artificial Intelligence*. Annals of the New York Academy of Science, pp. 91-107, 1984.
- (Feigenbaum et McCorduck, 1984) Feigenbaum, E., McCorduck, P., La cinquième génération, le pari de l'IA à l'aube du 21^{ème} siècle. 1984, Inter-Edition , Paris.
- (Fensel, 2001) Fensel, D., *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*. 2001, Springer.
- (Finin, & al., 1994) Finin, T., Fritzson, R., McKay, D., McEntire, R., KQML: a language and protocol for knowledge and information exchange. In *AAAI Technical Report WS-94-02*, pp. 93-103, 1994.
- (FIPA 2002) FIPA. Communicative Act Library Specification. Standard SC00037J, Foundation for Intelligent Physical Agents, 2002.

- (Frank, & al., 2004) Frank, A.-U., Grum, E., Vasseur, B., Procedure to Select the Best Dataset for a Task. *Third International Conference on Geographic Information Science (GIScience)*, Maryland, LNCS, vol. 3234/2004, pp.81-93, 2004, DOI: 10.1007/978-3-540-30231-5_6
- (Fürst, 2002) Fürst, F., Contribution à l'Ingénierie des Ontologies : une méthode et un outil d'opérationnalisation, Thèse de l'Université de Nantes, 2002.
- (Fürst, 2003) Fürst F., TooCoM : a Tool to Operationalize an Ontology with the Conceptual Graph Model. *Proceedings of the Workshop on Evaluation of Ontology-Based Tools (EON'2003) at ISWC'2003*, pp. 57-70.
- (Fürst, & al., 2004) Fürst, F., Leclère, M., Trichet, F., Operationalizing domain ontologies : a method and a tool, De Mantaras, R.-L., Saitta, L., (eds.), *European Conference on Artificial Intelligence (ECAI'2004)*, IOS Press, pp. 318-322, 2004.
- (Fürst et Trichet, 2006) Fürst, F., Trichet, F., Raisonner sur des ontologies lourdes à l'aide de Graphes Conceptuels -Une application à l'alignement d'ontologies de domaine-, *Actes des journées francophones Inforsid'2006*, vol.2, pp 879-895.
- (Gal et Shvaiko, 2009) Gal, A., Shvaiko, P., Advances in ontology matching, in *Advances in Web Semantics I*, T. S. Dillon, E. Chang, R. Meersman, and K. Sycara, Eds. Springer, pp. 176–198, 2009.
- (Ganascia, 1996) Ganascia, J.-G., *Le Petit Trésor. Dictionnaire de l'informatique et des sciences de l'information*. 1996, Édition Flammarion.
- (Gandon, 2002) Gandon, F.-L., *Distributed Artificial Intelligence and Knowledge Management: ontologies and multi-agent systems for a corporate semantic web*. Thèse de doctorat, INRIA et université de Nice - Sophia Antipolis, 2002.
- (Gandon, 2006) Gandon, F.-L., *Ontologies informatiques, Interstices*. 2006, <http://interstices.info/display.jsp?id=c7672>.
- (Gandon, 2008) Gandon, F.-L., *Graphes RDF et leur Manipulation pour la Gestion de Connaissances*. Mémoire d'Habilitation à Diriger les Recherches, INRIA Sophia Antipolis, Méditerranée Université de Nice, 2008.
- (Garf, 1996) Garf, B., *Lexique de philosophie*. 1996, Paris, Edition du Seuil.
- (Genesereth et Nilso, 1987) Genesereth, M.R., Nilso, N., *Logical Foundations of Artificial Intelligence.*, 1987, Morgan Kaufmann, USA.

- (Genest et Salvat, 1998) Genest, D., Salvat, E., A Platform allowing typed nested graphs : how CoGITo became CoGITaNT. *Proceedings of the 6th International Conference on Conceptual Structures*, LNAI 1453, pp. 154-161, 1998, Springer-Verlag.
- (Genest, 2000) Genest, D., *Extension du modèle des graphes conceptuels pour la recherche d'informations*. Ph. D. Thèse, Université Montpellier II, 2000.
- (Goh, 1997) Goh, C.-H., *Representing and Reasoning about Semantic Conflicts in Heterogeneous Information Sources*. Thèse de doctorat, MIT, Cambridge (MA US), 1997.
- (Gómez-Pérez, 1999) Gómez-Pérez, A., Développements récents en matière de conception, de maintenance et d'utilisation des ontologies. *Actes du colloque Terminologie et Intelligence Artificielle*, Nantes, 10-11 mai, pp. 9-20, 1999.
- (Gómez-Pérez & al., 2004) Gómez-Pérez, A., Fernandez-Lopez, M., Corcho O., *Ontological Engineering, with example from the areas of Knowledge Management, e-Commerce and the Semantic Web*, 2004, Springer-Verlag.
- (Goodchild, 2006) Goodchild, M.-F., Foreword. Dans Rodolphe Devillers et Robert Jeansoulin, Éditeurs, *Fundamentals of Spatial Data Quality*, GIS, pp. 13–16. ISTE, 2006.
- (Guarino et Giaretta, 1995) Guarino, N., Giaretta, P., *Ontologies and Knowledge Bases: Towards a Terminological Clarification*. In N. Mars (ed.) *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing*. IOS Press, Amsterdam: pp. 25-32, 1995.
- (Guarino, 1998) Guarino, N., *Ontology and Information Systems*. In N. Guarino (ed), *Formal Ontology in Information System (FOIS'98)*. Proc of the 1st International Conference, Trento, Italy, 6-8 June 1998, IOS Press, pp. 3-14.
- (Guebaili, 2006) Guebaili R., *Réutilisation des graphes conceptuels dans le web sémantique avec introduction d'information floue*. Mémoire de Magister Département d'Informatique USTHB, Encadreur: Mme Aissani_Mokhtari A., *Co-encadreur* : Akli-Astouati K. (Soutenu en Juillet 2006).
- (Guebaili & al., 2006 a) Guebaili R., Akli-Astouati K., Mokhtari A., *Fuzzy Conceptual graphs and their Adaptation to RDF and RDFS*. In 2006 International Symposium on AI - 50 Years – achievements, Future Directions and Social Impacts, Pékin-Chine, du 1 au 3 août, 2006.
- (Guebaili & al., 2006 b) Guebaili R., Akli-Astouati K., Mokhtari A., *Prise en compte d'informations floues au niveau de RDF, RDF(S) en correspondance avec les graphes*

conceptuels flous. In 9^{ème} Conférence Maghrébine sur les technologies de l'information, MCSEAI'06, Agadir - Maroc, 7-9 décembre, 2006.

(Guebaili & al., 2011) Guebaili-Djider R., Akli-Astouati K., Bellili A., Lacheheb H., *Agents coopératifs pour l'alignement d'Ontologies*. Extraction et Gestion des Connaissances Maghreb - EGC M'2011, 23-25 Novembre 2011, Tanger, Maroc.

(Guinaldo et Haemmerlé, 1997) Guinaldo, O., Haemmerlé, O., *CoGITo : une plate-forme logicielle pour raisonner avec des graphes conceptuels*, Actes du XV^e Congrès Inforsid, pp. 287-306, Toulouse, juin 1997.

(Grüber 1993) Grüber, T.-R., *A translation approach to portable ontology specifications*. Knowledge Acquisition, vol.5, n°2, pp.199–220, 1993.

(Grüber, 1995) Grüber, T.-R., *Toward Principles for the Design of Ontologies Used for Knowledge Sharing*. Revision of paper presented at the international workshop on Formal Ontology, Padova, Italy, March 1993, in Special issue of the International Journal of Human-Computer Studies, vol. 43, n°5-6, 1995, N. Guarino and R. Poli (Eds.).

(Grum et Vasseur, 2004) Grume, E., Vasseur, B., *How to select the Best Dataset for a Task?*, *Proceedings of the International Symposium on Spatial Data Quality*, GeoInfo Series, vol.28b, pp. 197-206, Bruck an der Leitha, Autriche, 15-17 April 2004.

(Hacid et Reynaud, 2003) Hacid, M.-S., Reynaud, C., *Intégration de Sources de données*. Action Scientifique CNRS/STIC Web Sémantique– Rapport Final V2, 2003.

(Hakimpour, 2003) Hakimpour, F., *Using Ontologies to Resolve Semantic Heterogeneity for Integrating Spatial Database Schemata*. Thèse de doctorat, Faculté de mathématique et d'informatique de l'université de Zürich, 2003.

(Hamblin, 1970) Hamblin, C.-L., *Fallacies*. 1970, London: Methuen.

(Hayes, 2004) Hayes, P., *RDF Semantics*, W3C recommendation, World Wide Web Consortium, February 10 2004, <http://www.w3c.org/TR/rdf-mt>.

(Hernandez, 2006) Hernandez, N., *Ontologies de domaine pour la modélisation du contexte en recherche d'information*. Thèse de doctorat, Université Paul Sabatier de Toulouse, 2006.

(Horrocks, & al., 2004) Horrocks, I., Patel-Schneider, PF., Boley, H., Tabet, S., Grosz, B., Dean, M., *SWRL: A semantic web rule language combining OWL and RuleML*, W3C Member submission, 2004.

- (Irandooust, & al., 2001) Irandooust, H., Moulin, B., Belanger, M., De l'usage de l'explication et de l'argumentation dans les systèmes à base de connaissances. Dans *Cognito*, n° 23, pp. 3-22, 2001.
- (Izza, 2006) Izza, S., *Intégration des systèmes d'information industriels: Une approche flexible basée sur les services sémantiques*. Thèse de doctorat, Ecole Nationale Supérieure des Mines de Saint-Etienne, 2006.
- (Jaro, 1989) Jaro, M., Advances in record-linkage methodology as applied to matching the census of Tampa. *Journal of the American Statistical Association*, 84(4), pp.14-420, (1989).
- (Jingshan, & al., 2005) Jingshan H., Zavala-Gutierrez, R.-L., Mendoza-García, B., Huhns, M.-N., (2005), A schema-based approach combined with inter-ontology reasoning to construct consensus ontologies. In American Association for Artificial Intelligence (www.aaai.org).
- (Kalfoglou et Schorlemmer, 2003) Kalfoglou, Y., et Schorlemmer, M., Ontology mapping: the state of the art. *The Knowledge Engineering Review*, vol. 18, n°1, pp.1-31, 2003.
- (Kassel, 2002) Kassel, G., OntoSpec : une méthode de spécification semi-informelle d'ontologies. Dans *Actes des journées francophones d'Ingénierie des Connaissances (IC'2002)*, pp. 75-87.
- (Kayser, 97) Kayser, D., *La représentation des connaissances*, ISBN 2-86601-647-5, 1997, Hermès, Paris.
- KIF. (2004). *KIF, Knowledge Interchange Format Home Page*. <http://logic.stanford.edu/kif/dpans.html>
- (Klein, 2001) Klein, M., Combining and relating ontologies : an analysis of problems and solutions. In *Proceedings de IJCAI Workshop on Ontologies and Information Sharing, 2001*, Seattle (WA US).
- (Knuth et Bendix, 1970) Knuth, D., Bendix, P., Simple word problems in universal algebras. *Computational Problems in Abstract Algebra*, pp. 263-297, 1970.
- (Kuhn, 2007) Kuhn, W., avec la collaboration de Talpin, J-J., Robida, F., pour les textes en français, What do you mean by "Semantics"of Geodata?, la revue du BRGM, pour une terre durable, geosciences, n°6 octobre 2007, Terre virtuelle, Les systèmes d'information Géoscientifique, pp. 14-19.
- (Laera, & al., 2006) Laera, L., Tamma, V., Bench-Capon, T.J.M., Euzenat, J., Agent-based Argumentation for Ontology Alignments, In *Proceeding of the Workshop on Computational Models of Natural Argument (CMNA)*, 2006.

- (Lammare, 2009) Lammare, P., Contributions à la recherche d'information dans des systèmes distribués, ouverts, intégrant des participants autonomes, HDR, au Laboratoire d'Informatique de Nantes Atlantique, 2009.
- (Laublet, & al., 2002) Laublet, P., Reynaud, C., Charlet, J., Sur Quelques Aspects du Web Sémantique. *Actes des deuxièmes assises nationales du GdRI3*, pp. 59-78, 2002.
- (Lefèvre, 2000) Lefèvre, P., La Recherche d'informations. Du texte intégral au thésaurus. Paris, France : Hermès-Lavoisier, 2000, 1er édition. ISBN : 2-7462-0173-9.
- (Li et Yang, 2008) Li, L., Yang, Y., Agent negotiation based ontology refinement process and mechanisms for service applications. *Service Oriented Computing and Applications 2*, pp.15-25, 2008.
- (Lieberman, & al., 2006) Lieberman, J., Pehle, T., Morris, C., Kolas, D., Dean, M., Lutz, M., Probst, F., Klien, E., Geospatial Semantic Web Interoperability Experiment Report, in 2006, Lieberman, J (ed), Open Geospatial Consortium Inc.
- (Lin, 1998) Lin, D., An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning*, pp. 296-304, 1998, Madison (WI US).
- (Liping, et Peisheng, 2008), Liping D., Peisheng Z., Geospatial Semantic Web, Interoperability, *Encyclopedia of GIS*, Part 10, pp.398-403, 2008, DOI: 10.1007/978-0-387-35973-1_516.
- (Longley, & al., 1999) Longley, P.-A., Goodchild, M.-F., Maguire, D.-J., Rhind, D.-W., ed., *Geographical Information Systems*. 1999, John Wiley & Sons.
- (Lopes, & al., 2010) Lopes, N., Lukácsy, G., Polleres, A., Straccia, U., Zimmermann, A., A general framework for representing and reasoning with annotated semantic web data. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI 2010), Special Track on Artificial Intelligence and the Web*, Atlanta, Georgia, USA, July.
- (Luger et Stubblefield, 1993) Luger, G.-F., Stubblefield, W.-A., *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*. Palo Alto, CA: Benjamin Cummings, 1993, 2nd Ed.
- (Lukose, 1997) Lukose, D., CGKEE: Conceptual graph knowledge engineering environment, In *Lecture Notes in Computer Science*, vol.1257/1997, pp.598-602, 1997, DOI: 10.1007/BFb0027904, Springer.
- (Luong, 2007) Luong, P.-H., *Gestion de l'évolution d'un Web Sémantique d'entreprise*. Thèse de doctorat, l'Ecole des Mines de Paris, 2007.

- (Maedche, 2002) Maedche, A., *Ontology Learning for the Semantic Web*. 2002, Kluwer Academic Publishers, Norwell, MA, USA.
- (Maedche, & al., 2002) Maedche, A., Motik, B., Silva, N., Volz, R., *MAFRA — A MApping FRAmework for Distributed Ontologies*. In: Proceedings of the EKAW, 2002.
- (Manola et Miller, 2004) Manola, F., Miller, E., *W3 RDF Primer*. W3C, *working Draft 23 January 2004*, www.w3.org/TR/rdf-primer/.
- (Martin, 1996) Martin, P., *Exploitation de Graphes Conceptuels et de Documents Structurés et Hypertextes pour l'Acquisition de Connaissances et la Recherche d'Informations*. Thèse de Doctorat en Informatique, Université de Nice - Sophia Antipolis, 1996.
- (Martin, 1997) Martin, P., *CGKAT: a Knowledge Acquisition Tool and an Information Retrieval Tool Which Exploits Structured Documents, Conceptual Graphs and Ontologies*. In Proc. of CGTOOLS'97, 2nd Conceptual Graph Tools Workshop, University of Washington, Seattle Washington USA, August 8, 1997.
- (Maudet, 2001) Maudet, N., *Modéliser les conventions des interactions langagières: la contribution des jeux de dialogue*. Thèse de doctorat, Université Paul Sabatier, Toulouse, 2001.
- (Maudet, & al., 2007) Maudet, N., Parsons, S., Rahwan, I., *Argumentation in Multi-Agent Systems: Context and Recent Developments*. In Proc. *Argumentation in MultiAgent Systems (ARGMAS06)*. Post-Proc. *The Third International Workshop*, (Revised), LNCS 4766, Springer-Verlag, pp.1–16, 2007.
- (Mazzieri, 2004) Mazzieri, M., *A fuzzy RDF semantics to represent trust metadata*. In *Proceedings of the In: 1st Workshop on Semantic Web Applications and Perspectives (SWAP2004)*, pp.83-89, 2004, Ancona, Italy:
- (Mazzieri et Dragoni, 2005) Mazzieri, M., Dragoni, A.-F., *A fuzzy semantics for semantic web languages*. *Proceedings of the ISWC Workshop on Uncertainty Reasoning for the Semantic Web*, CEUR Workshop Proceedings, 2005.
- (Mazzieri et Dragoni, 2008) Mazzieri, M., Dragoni, A.-F., *A fuzzy semantics for the resource description framework*. *Workshop on Uncertainty Reasoning for the Semantic Web*. da Costa, P.C.G., d'Amato, C., Fanizzi, N., Laskey, K.B., Laskey, K.J., Lukasiewicz, T., Nickles, M., Pool, M. (eds.) LNCS (LNAI), vol. 5327, pp. 244–261, 2008. Springer, Heidelberg.
- (McCarthy & Hayes, 1969) McCarthy, J., Hayes, P., *Some philosophical problems from the standpoint of artificial intelligence*. In B. Meltzer & D. Michie (Eds), *Machine Learning 4*, Edinburgh University Press, 1969.

- (McCarthy, 1980) McCarthy, J., Circumscription - A Form of Non-Monotonic Reasoning. *Artificial Intelligence*, vol.13, pp.27-39, 1980.
- (Medcraft, & al., 2003) Medcraft, P.-S., Schiel, U., Baptista C.-S., DIA: Data Integration Using Agents. *International Conference on Enterprise Information Systems - ICEIS*, pp. 79-86, 2003.
- (Mellal, 2007) Mellal, N., Réalisation de l'interopérabilité sémantique des systèmes, basée sur les ontologies et les flux d'information. Thèse de doctorat, Polytech'Savoie, 2007.
- (Mhiri et Despres, 2005) Mhiri, S., Despres, S., Intégration d'ontologies : Recherche exploratoire et expérimentation d'outils d'alignement d'ontologies au format OWL. *7ème Journées Doctorales Informatique et Réseau*. Université de Technologie de Troyes, 2005.
- (Minsky, 1975) Minsky, M., *A Framework for Representing Knowledge*. MIT-AI Laboratory Memo 306. Reprinted in *The Psychology of Computer Vision*, 1975, P. Winston (Ed.), McGraw-Hill.
- (Mizuguchi, 2003) Mizuguchi, R., Tutorial on ontological engineering - Part 1: Introduction to Ontological Engineering. *New Generation Computing*, vol.21, n°4, pp. 365-384, 2003.
- (Mochol, & al., 2004) Mochol, M., Oldakowski, R., Heese, R., Ontology-based-Recruitment Process, Proc. of the GI2004 Conference, Ulm, Germany: <http://page.mi.fu-berlin.de/mochol/papers/SemTech.pdf>.
- (Morge, & al., 2006a) Morge, M., Routier, J.-C., Secq, Y., Dujardin, T., How to reach an agreement on representation. In Roberta Ferrario, Nicola Guarino, and Laurent Prevot, editors, *Proceedings of the Workshop Formal Ontologies for Communicating Agents (FOCA)*, August 2006.
- (Morge, & al., 2006b) Morge, M., Routier, J.-C., Secq, Y., Dujardin, T., Comment atteindre un accord sur une représentation?, Dans *Journées Francophones sur les Systèmes Multi-Agents*, Octobre 2006.
- (Morge et Routier, 2007) Morge, M., Routier, J.-C., Debating over heterogeneous descriptions. *Applied Ontology, Special issue on Formal Ontology for Communicating Agents*, 2 (3-4) , pp.333-349, 2007.
- (Morton, 1987) Morton, S.-K., *Conceptual Graphs and fuzziness in artificial Intelligence*. PhD thesis, University of Bristol, 1987.
- (Mugnier et Chein, 1996) Mugnier, M.-L., Chein, M., Représenter des connaissances et raisonner avec les graphes. *Revue d'intelligence Artificielle*, vol.10 ; n°1, pp.7-56, 1996.

- (Nagy et Vargas-Vera, 2010) Nagy, M., Vargas-Vera, M., Towards an Automatic Semantic Data Integration: Multi-agent Framework Approach. In Gang Wu (Eds.) *Book chapter in Semantic Web*, pp.107-134, 2010. In-Tech.
- (Napoli, 1997) Napoli, A., Une introduction aux logiques de descriptions. Rapport de recherche INRIA, n° 3314, Décembre 1997. ISSN 0249-6399.
- (Nebert, 2001) Nebert, D., Developing Spatial Data Infrastructures. *The SDI Cookbook v.1.1, Global Spatial Data Infrastructure*, 2001.
- (Neches & al., 1991) Neches, R., Fikes, R., Finin, T., Gruber, T., Senator, T., Swartout, W., Enabling technology for knowledge sharing. *AI Magazine*, n°12, pp. 36–56, 1991.
- (Newell et Simon, 1976) Newell, A., Simon, H., *Computer science as empirical inquiry. Symbols and search*. ACM Turing award lecture, Communication of the ACM, vol. 19, n°3, pp 113-126, 1976.
- (Newell, 1980) Newell, A., *The Knowledge Level*. Presidential Address, American Association for Artificial Intelligence, AAA'80, Stanford University, 19 Aug. 1980.
- (Noy et Hafner, 1997) Noy, N.-F., Hafner, C., *The State of the Art in Ontology Design: A Survey and Comparative Review*. In *AI Magazine*, vol. 18, n°3, pp. 53-74, 1997.
- (Noy et Musen 2000) Noy, N.-F., Musen, M., Algorithm and tool for automated ontology merging and alignment. Proceedings of the 17th National Conference on Artificial Intelligence, AAAI, Austin, Texas, USA, pp. 450– 455, 2000.
- (Noy et Musen, 2001 a) Noy, N.-F., Musen, M., Anchor-PROMPT: Using non-local context for semantic matching. In *Proc. IJCAI 2001 workshop on ontology and information sharing*, pp. 63–70, 2001. Seattle (WA US).
- (Noy et Musen 2001 b) Noy, N.-F., Musen, M., Using non-local context for semantic matching. Proceedings of the Workshop on Ontologies and Information Sharing at the Seventeenth International Joint Conference on Artificial Intelligence, Seattle 01, pp. 334–350, 2001.
- (Noy et McGuinness, 2001) Noy, N.-F., McGuinness, D.-L., (2001), *Ontology Development 101: A Guide to Creating Your First Ontology*; Stanford Knowledge Systems Laboratory, Technical Report [KSL-01-05](#) and Stanford Medical Informatics, March 2001.
- (Oléron, 1991) Oléron P., *L'intelligence, Que sais-je?*, 5^{ème} édition, ISBN: 978-2-13-043796-3, 1991.

- (Oulefki et Akli, 2008 a) Oulefki, S., Akli_Astouati K., Classification des techniques d'appariement d'ontologies. In International Conference on Web and Information Technologies. 29-30 June, 2008, Sidi Bel Abbes, Algeria.
- (Oulefki, 2008) Oulefki, S., Contribution à l'appariement des ontologies: Classification, Processus et Comparaison d'alignements. Mémoire de Magister, Département d'informatique, USTHB, Directeur de thèse: Akli-Astouati K., (Soutenu en Juillet 2008).
- (Packer, & al., 2009) Packer, H.-S., Gibbins, N., Jennings, N.R., Ontology evolution through agent collaboration. 2009, *Brain 1-4*.
- (Pathan, 2009) Pathan, S.-K., Metadata Standards for Digital Geospatial data. Version 2.0, Prepared by the Working Group on "Metadata Standards" chaired by Dr S K Pathan, SAC ISRO Ahmadabad; Department of Science & Technology Government of India, New Delhi; RESA/ No.09-2/TEC/09/1, 17 Feb. 2009.
- (Peirce, 1882) Peirce, C.-S., Lettre, Peirce to O.-H. Mitchell. In Writtings of Cherles, S.Peirce, vol. 4, pp. 394-399, MA, Indiana University Press, Bloomington IR, 1882
- (Pitrat, 1990) Pitrat, J., Métaconnaissance: Future de l'intelligence artificielle. 1990. Edition Hermès, Paris.
- (Plantin, 2011) Plantin, C., Persuasion or alignment?, Zenker, F. (ed.). *Argument Cultures: Proceedings of the 8th International Conference of the Ontario Society for the Study of Argumentation (OSSA)*, May 18-21 2011, Windsor, pp. 1-11.
- (Pochon, 2006) Pochon, L.-O., De la possibilité d'usage d'ontologies pour la gestion de contenus mathématiques. Document de travail, 06.1007 Octobre 2006, <http://www.irdp.ch/>.
- (Prakken, 2000) Prakken, H., On dialogue systems with speech acts, arguments, and counter arguments. In 7th European Workshop on Logic for Artificial Intelligence, Malaga, LNCS, vol. 1919/2000, pp. 224-238, 2000. DOI: 10.1007/3-540-40006-0_16.
- (Pratt, 1995) Pratt, V., *Machines à penser. Une histoire de l'intelligence artificielle*. Trad. de l'américain par C. PUECH. Paris, Presses universitaires de France, 15 x 21,7, viii-299 p., index (Sciences Modernités Philosophies), 1995.
- (Prie et Garlati, 2003) Prie, Y., Garlati, S., Métadonnées et annotations dans le web Sémantique. Action Scientifique CNRS / STIC, Web Sémantique – Rapport Final, 2003.
- (Prince, 1991) Prince, V., Expertise naturelle, expertise artificielle : vers quels paradigmes cognitifs ? *Intellectica* numéro spécial Expertise et sciences cognitives, n°12, Paris, pp.7-31, 1991.

- (Prud'hommeaux, 2006) Prud'hommeaux, E., SPARQL RDF Query Language, 6 April 2006. <http://www.w3.org/TR/2006/CR-rdf-sparql-query-20060406/>
- (Psyché, & al., 2003) Psyché, V., Mendes, O., Bourdeau, J., Apport de l'ingénierie ontologique aux environnements de formations à distance. *Revue sticef.org*, vol. 10, 2003.
- (Quillian, 1968) Quillian, M.-R., *Semantic memory*. In M., Minsky, (Ed), *Semantic information processing*, pp. 216-260, 1968. Cambridge, MA : MIT Press.
- (Rahm et Bernstein, 2001) Rahm, E., Bernstein, P., A survey of approaches to automatic schema matching. *The VLDB Journal*, vol. 10, n° 4, pp. 334–350, 2001.
- (Rahwan et Simari, 2009) Rahwan, I., Simari, G.-R., *Argumentation in Artificial Intelligence*, (2009), (Eds.). Springer.
- (Resnik, 1995) Resnik, P., Using information content to evaluate semantic similarity in a taxonomy., vol. 10, pp. 448–453, 1995. Montréal, Canada.
- (Robillard, 2006) Robillard, J., *Connaissance et information: une analyse comparative critique*, Séminaires de recherche sur la communication, l'information et la cognition, Équipe de recherche sur les acteurs, les dispositifs et les enjeux de la communication (ERADEC), Télé-université (Université du Québec), 2006; disponible à http://eradec.teluq.quebec.ca/IMG/pdf/Connaissance_et_information-vfin-5.pdf
- (Royer, 1994) Royer, V., *Partage de croyances : condition nécessaire pour un système coopératif ?*. *Systèmes coopératifs: de la modélisation à la conception*, sous la direction de B. Pavard, pp. 53-71, 1994. Toulouse, Octarès eds.
- (Sabouret, 2009) Sabouret, N., *Interactions sur le fonctionnement dans les systèmes multi-agents ouverts et Hétérogènes*. Mémoire d'habilitation à diriger des recherches, Université Pierre et marie Curie, Paris, 2009.
- (Safar, & al., 2007) Safar, B., Reynaud, C., Calvier, F.-E., *Techniques d'alignement d'ontologies basées sur la structure d'une ressource complémentaire*. *Ieres Journees Francophones sur les Ontologies*, 2007.
- (Samyn, 2002) Samyn, O., *Traduction entre niveaux d'abstraction pour des applications de bases de données*. Mémoire d'études approfondies en sciences appliquées, Université Libre de Bruxelles, 2002.
- (Sanchez et Yamanoi, 2006) Sanchez, E., Yamanoi, T., *Fuzzy Ontologies for the Semantic Web*. H. Larsen et al. (Eds.): *FQAS 2006, LNAI 4027*, pp. 691–699, 2006. Springer-Verlag Berlin Heidelberg.

- (Schell, 2007) Schell, D., Geoscientists' geospatial web requires technology convergence. La revue du BRGM, pour une terre durable, geosciences, n°6 octobre, Terre virtuelle, les systèmes d'information Géoscientifique, pp. 98-105, 2007.
- (Scheuer, & al., 2010) Scheuer, O., Loll, F., Pinkwart, N., McLaren, B.,-M., Computer-supported argumentation: A review of the state of the art, *Computer-Supported Collaborative Learning* (2010) 5:43–102, 2010, DOI 10.1007/s11412-009-9080-x
- (Schiffrin, 1987) Schiffrin, D., *Discourse markers*, 1987. Cambridge: Cambridge University Press.
- (Scheuer, & al., 2010) Scheuer, O., Loll, F., Pinkwart, N., McLaren, B.,-M., Computer-supported argumentation: A review of the state of the art, *Computer-Supported Collaborative Learning* (2010) 5:43–102, 2010, DOI 10.1007/s11412-009-9080-x
- (Schmidt, 1991) Schmidt, K., *Cooperative Work: A Conceptual Framework*. Dans Rasmussen J., Brehmer B. et Leplat J. (Eds) *Distributed Decision Making: Cognitive Models for Cooperative Work*, John Wiley & Sons Ltd, pp. 75-110, 1991.
- (Seaborne, 2004) Seaborne, A., A Query Language for RDF, 9 January 2004; <http://www.w3.org/Submission/2004/SUBM-RDQL-20040109/>
- (Searle, 1969) Searle, J., *Speech Acts: An Essay in the Philosophy of Language*. 1969, Cambridge University Press, UK.
- (Shvaiko et Euzenat, 2012) Shvaiko, P., Euzenat, J., *Ontology matching: state of the art and future challenges*. A paraître dans *IEEE Transactions on Knowledge and Data Engineering*.
- (Silva, & al., 2005) Silva, N., Maio, P., Rocha, J., An approach to ontology mapping negotiation. In *Proceedings of the K-CAP, Workshop on Integrating Ontologies Banff, Canada, October 2 2005*.
- (Smith et Welty, 2001) Smith, B., Welty, C., *Ontology: Towards a new synthesis*. *Proceeding of the international conference on Formal Ontology in Information Systems (FOIS '01)*, pp. 3-9, 2001.
- (Sobh, 2009) Sobh, T., Information Fusion Using Ontology-Based Communication between Agents. *International Arab Journal of e-Technology*, vol.1, n°2, 2009.
- (Sowa, 1984) Sowa, J.-F., *Conceptual structures in mind and machine*, 1984. Addison–Wesley Reading, MA.

- (Sowa, 1991) Sowa, J.-F., Toward the Expressive Power of natural Language. In Principles of Semantic Networks – Explorations in the representation of knowledge, pp.177-182, 1991. Edited by John F. Sowa, IBL systems Research.
- (Sowa, 1992) Sowa, J.-F., Conceptual graphs as a Universal Knowledge Representation, In Computers and Mathematics with Applications, vol.2, pp.75-93, 1992. Reprinted in Fritz Lehmann (ed.), Semantic Networks in Artificial Intelligence.
- (Sowa, 2000) Sowa, J.F., Knowledge Representation. (2000). Brooks Cole Publishing, Pacific Grove, CA, USA.
- (Spaccapietra et Parent, 1991) Spaccapietra, S., Parent, C., Conflicts and correspondence assertions in interoperable databases, SIGMOD Record, vol.20, no.4, pp.49–54, 1991.
- (Staab et Maedche, 2000) Staab, S., Maedche, A., *Axioms are objects too: Ontology engineering beyond the modeling of concepts and relations*. (2000), Research report 399, Institute AIFB, Karlsruhe.
- (Straccia, 2009) Straccia U., A Minimal Deductive System for General Fuzzy RDF. TR-ISTI 10 July 2009.
- (Straccia, & al., 2010) Straccia, U., Lopes, N., Luk´acsy, G., Polleres, A., A general framework for representing and reasoning with annotated semantic web. *Proceedings of the AAAI Conference on Artificial Intelligence*, Special Track on Artificial Intelligence and the Web, Atlanta, Georgia, USA, July 2010.
- (Suarez-Figueroa et Gomez-Perez, 2003) Suarez–Figueroa, M.-C., Gomez–Perez, A., Result of taxonomic Evaluation of RDF(S) and DAML+ OIL ontologies using RDF(S) and DAML+ OIL validation Tools and Ontology Platforms import services. EON 2003, *2nd International Workshop on Evaluation of Ontology-based Tools ISWC2003 Workshop*, Sanibel Island, Florida, October 20th.
- (Thomopoulos, & al., 2003) Thomopoulos, R., Buche, P., Haemmerlé, O., Different kinds of comparisons between fuzzy conceptual graphs. *Proceedings of the International Conference on Conceptual Structures*, LNAI 2746, 2003.
- (Trojhan, & al., 2008) Trojahn, dos Santos C., Quaresma, P., Vieira, R., Moraes, M., A cooperative approach for composite ontology mapping. *LNCS Journal on Data Semantic X (JoDS)*, 4900(0), pp.237-263, 2008.
- (Trojhan, & al., 2010) Trojhan, dos Santos C., Quaresma, P., Vieira, R., Isaac, A., Comparing argumentation frameworks for composite ontology matching. In Peter McBurney, Iyad Rahwan, Simon Parsons, and Nicolas Maudet, editors, *Argumentation in Multi-Agent Systems*, 6th International Workshop, ArgMAS 2009, pp.305-320. Springer, 2010.

- (Trojahn et Euzenat, 2010) Trojahn dos Santos, C., Euzenat, J., Consistency-driven argumentation for alignment greement. OM 2010 Proceedings of the 5th International Workshop on Ontology Matching (OM-2010), Shanghai, China, November 7 2010.
- (Udrea, & al., 2006) Udrea, O., Recupero, D.-R., Subrahmanian, V.-S., Annotated RDF, In *The Semantic Web: Research and Applications*, LNCS vol. 4011/2006, pp. 487-501, 2006. DOI: 10.1007/11762256_36, Sure, Y., Domingue, J. (eds.), Springer, Heidelberg.
- (Uschold et Gruninger, 1996) Uschold, M., Gruninger, M., *Ontologies : Principales Methodes and Applications*. In the *Knowledge Engineering Review*, vol.11, n°2, pp.93-136, 1996.
- (Van digglen, & al., 2006) Van digglen J., Beun R.-J., Dignum F., Van Eijk R., Meyer J.-J., ANEMONE: An effective minimal ontology negotiation environment. In *Proceedings of the 5th International Conference on Autonomous Agents and Multi-agent Systems*, pp. 899–906, 2006, ACM Press.
- (Van Eemeren, & al., 1996) Van Eemeren, F.-H., Grootendorst, R.-F., Henkemans, F.-S., *Fundamentals of Argumentation Theory: A Handbook of Historical Backgrounds and Contemporary Applications*. 1996, Lawrence Erlbaum Associates, Hillsdale NJ, USA.
- (Vaneková, & al., 2005) Vaneková, V., Bella, J., Gurský, P., Horváth, T., Fuzzy RDF, in the semantic web: Deduction and induction. In *Proceedings of the 6th Workshop on Data Analysis, WDA'05*, pp.16-29, 2005.
- (Vázquez-Naya & al., 2009) Vázquez-Naya, J.M. et al., *Ontology Alignment Overview*. *Encyclopedia of Artificial Intelligence*, pp. 1283-1289, 2009.
- (Wache, & al., 2001) Wache, H., Voegele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., et al. *Ontology-based integration of information - a survey of existing approaches*. In *Proceedings of the workshop on Ontologies and Information Sharing at the International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 108–117, 2001.
- (Walton et Krabbe, 1995) Walton, D., Krabbe, E., *Commitment in Dialogue*. 1995, SUNY Press.
- (Winkler,1999) Winkler, W., *The of record linkage and current research problems*, Statistics of Income Division. Internal Revenue Service Publication.1999.
- (Wuster, 1981) Wuster, E., *L'étude scientifique générale de la terminologie, zone frontalière entre la linguistique, la logique, l'ontologie, l'informatique et la science des choses*. Dans G. Rondeau & H. Felber, *Coordinateurs, Textes choisis de terminologie.*, vol. I: *Fondements théoriques de la terminologie.*, pp.55–113, 1981. Québec, Canada: Université de Laval - GIRSTERM.

- (Wuwongse et Manzano, 1993), Wuwongse, V., Fuzzy Conceptual Graphs. Proceeding of the 1st International Conference on Conceptual Structures, LNAI n°699, pp.430-449, August 1993, Quebec City, Canada, Springer-Verlag.
- (Zadeh, 1965) Zadeh, L., Fuzzy Set, Information and control, vol.8, pp.338–353, 1965.
- (Zadeh, 1975) Zadeh, L., The concept of a linguistic variable and its application to approximate reasoning-Part I, II, In Information Sciences, 1975. Elsevier.
- (Zaraté, 2005) Zaraté, P., Des Systèmes Interactifs d’Aide à la Décision Aux Systèmes Coopératifs d’Aide à la Décision : Contributions conceptuelles et fonctionnelles. Mémoire d’Habilitation à Diriger des Recherches, Institut National Polytechnique de Toulouse (INPT), 2005.
- (Zghal, & al., 2007) Zghal, S., Kamoun, K., Ben-Yahia, S., Edola : Une nouvelle méthode d’alignement d’ontologies owl-lite. Dans Actes de la quatrième CONFérence en Recherche d’Information et Applications, CORIA, 28-30 mars, Saint-Etienne, pp.450-455, 2007.