RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
Université Scientifique et Technologique Houari Boumèdienne
Faculté de Génie Electrique, Département Informatique

Mémoire de magister

RECHERCHE D'INFORMATION INTELLIGENTE SUR LE WEB : Extension du langage OWL

Hichem ZAIT

Présenté devant le jury composé de :

Prof. Mme. Z. Alimazighi	Présidente
M.C. Mr. O. Nouali	Examinateur
M.C. Mme. N. Bensaou	Examinatrice

Encadré par :

Prof. Aïcha AÏSSANI-MOKHTARI

Juin 2006

Une recherche d'information intelligente est une recherche intelligente de l'information que l'on souhaite 'vraiment' obtenir comme réponse à une requête que l'on formule au langage humain. L'intelligence consiste donc à comprendre cette requête et de chercher la réponse qui soit la plus pertinente possible avec le minimum du bruit et du silence. Je formule cette requête "quel est le degré d'intelligence à lequel vous vous attendez ?" ... c'est relatif !

Hichem ZAIT

Remerciements

Au Prof. Mme Aïcha AÏSSANI-MOKHTARI de m'avoir encadré tout au long de ce magistère et pour son orientation et ses remarques très pointues. Je la remercie surtout de m'avoir encouragé à prendre le choix de ce sujet et ses participations actives dans les présentations organisées au sein du laboratoire.

A Mme. Z.Alimazighi d'avoir accepté de présider le jury.

A Mr. O.Nouali et Melle. N.Bensaou d'avoir accepté d'être membres du jury.

A mes parents pour leur prières et encouragement.

A ma femme de m'avoir aidé à la rédaction de ce mémoire, pour son support moral et sa grande patience.

A mes frères et sœurs.

A mes chères amis : Youcef Salem Atia, Lamine Boulahlib, Youcef Cherfi, Hichem, Baghdad, Abdelhakim, Karim, Lyes et tous les anciens colleagues de la section C de la promotion 2003.

Table des matières

Table des figures

Glossaire

Bruit (*Noise*). Ensemble de documents non pertinents qui sont retournés en réponse à une requête dans un système documentaire.

Classe (*Cluster*). Ensemble de documents (ou de segments) qui présentent des caractéristiques communes.

Classification (*Clustering*). Opération qui consiste à regrouper des documents (ou des segments) ayant des caractéristiques communes.

Collection. Voir corpus.

Corpus (Corpus, Collection). Ensemble de textes qui peuvent posséder des caractéristiques linguistiques communes.

Document. Entité pouvant contenir exclusivement ou de façon combinatoire du texte (document textuel), des images, des sons et des vidéos. Dans cette thèse, un document est exclusivement textuel.

Etiquette. Voir Thème.

Homonymie. Caractère des mots qui ont la même forme graphique mais un sens différent.

Hyperonyme. Se dit d'un terme dont le sens inclut le sens d'autres termes.

Lemmatisation (*Stemming*). Opération qui consiste à réduire un terme en lemme.

Modèle vectoriel (*Vector Space Model*). Modèle qui représente les documents et les requêtes par un vecteur de termes.

Ontologie. Spécification explicite d'une conceptualisation

Pertinence. Ensemble de documents pertinents qui sont retournés en réponse à une requête dans un système documentaire.

Polysémie (*Polysemy*). Propriété d'un terme qui présente plusieurs sens.

Précision. Mesure la capacité d'un système de recherche d'information à retrouver uniquement l'ensemble des documents pertinents en réponse à une requête.

Rappel. Rapport entre le nombre de documents pertinents retournés en réponse à une requête et le nombre total de documents pertinents contenus dans la base documentaire pour la requête.

RI. Recherche d'Information

Segment. Une partie d'un texte. Un segment peut correspondre à un ensemble de mots, une phrase, un paragraphe, une section de taille fixe, etc.

Silence. Ensemble de documents pertinents non retournés en réponse à une requête dans un système documentaire.

Synonymie. Relation entre des mots qui ont un sens très voisin.

Syntagme. Groupe d'éléments formant une unité dans une organisation hiérarchisée.

Terme. Désigne un mot ou un groupe de mots.

Thème (*Topic*). Désigne le syntagme caractérisant une classe.

Chapitre 1
INTRODUCTION GÉNÉRALE

La recherche menée dans ce mémoire s'intéresse à la recherche d'information intelligente sur le Web avec le minimum de bruit et de silence. La grande ambition de rendre le Web d'aujourd'hui intelligent nous a stimulé non seulement à chercher des solutions auprès des ontologies mais aussi à concrétiser les recherches effectuées dans le cadre de la logique de description et d'étendre le nouveau langage d'ontologie Web.

L'accès actuel à l'information via les moteurs de recherche semble satisfaire plusieurs navigateurs dés la première recherche. Que se cache-t-il derrière un moteur de recherche et qu'elles sont ses limites en terme de pertinence ?

Le mot moteur de recherche est souvent lié au mot Google, laissons nous donc voir Google de plus prêt afin de répondre à cette question.

1.1 Problème de pertinence

Les résultats d'appariement des requêtes avec index sont généralement présentés par ordre de pertinence (relevance ranking), le but étant d'afficher sur la première page les 15 à 20 premières réponses qui sont les plus pertinentes car près de 90% des internautes ne consulteront que celle-là. On peut considérer ici la pertinence comme "la nature de la relation entre les documents recherchés et le besoin d'information de l'utilisateur" [GAU03].

On distingue trois méthodes de classement :

1.1.1 Tri par pertinence / indice de pertinence

Cette notion de pertinence est déterminée selon quatre critères qui sont la fréquence d'occurrence du mot dans la base de données, sa densité, sa position

dans le texte, et la similarité des mots du document avec les termes de la requête [GAU03].

L'inconvénient de cette méthode de tri repose sur le fait qu'il est très facile de la corrompre à des fins autres qu'une recherche efficace : c'est ce que l'on appelle le spamming. En effet, il est tout à fait possible, lors de la conception d'une page web, de faire volontairement apparaître de façon redondante certains termes, afin de tromper le moteur de recherche.

1.1.2 Tri par popularité / indice de popularité

Développé pour pallier au tri par pertinence, cette méthode trie en fonction des liens que contient la page web (les hyperliens). On considère que plus une page a de liens plus son indice de popularité est élevé. Ce tri est réalisé à l'aide de la citation ou de la mesure d'audience.

La méthode de co-citation repose sur un algorithme chargé de repérer les liens qui relient le moteur de recherche Google qui, le premier, a utilisé cette méthode avec le PageRank : le moteur de recherche affiche les réponses aux requêtes selon l'ordre de référencement des pages. Ce classement est donc indépendant du contenu, contrairement au classement par pertinence.

L'inconvénient majeur qui se pose est que les pages orphelines ne sont pas valorisées. De plus, il est très facile de corrompre ce système avec des liens artificiels, c'est-à-dire des liens achetés : des enjeux économiques rentrent en ligne de compte.

Exemple

Lors d'une recherche dans Google sur le mot "Carte Graphique" (figure 1.1), le premier résultat de la recherche pointe sur un site d'achat qui propose différentes cartes graphiques à vendre. Le classement tel que présenté par Google ne correspond pas forcément à la requête demandée, l'internaute peut demander des définitions sur les cartes graphiques ou souhaite connaître les différents types des cartes graphiques mais pas les acheter. Le site sur lequel pointe le premier résultat contient une grande liste de liens vers des pages contenant plus de détails d'achat. Comme chaque lien correspond à un type de carte graphique, on peut imaginer le nombre de liens contenus sur la première page de ce site (figure 1.2). C'est le principe du PageRank.

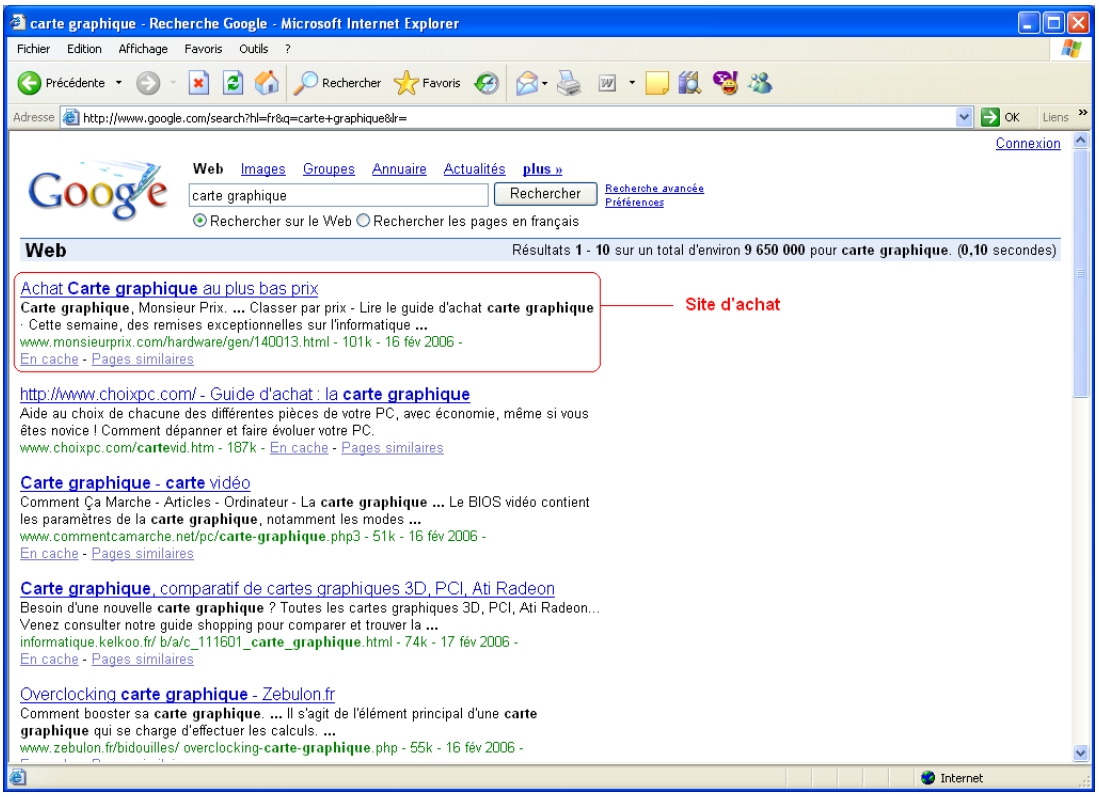

FIG. 1.1 – Exemple de recherche pour “carte graphique”

1.1.3 Tri par calcul dynamique de catégories

Il s'agit de classification des résultats dans des catégories reflétant des concepts liés aux résultats de l'appariement requête/index, catégories qui sont générées automatiquement. On appelle aussi cette technique clustering. Son grand avantage est qu'elle est fondée sur le contenu du document, contrairement aux méthodes de PageRank ou autres tris par popularité.

Néanmoins, il est bon de noter que ces méthodes ne permettent pas un tri sans failles, dû sûrement aux problèmes de manque de structuration de l'objet indexé et aux diversités de contenu de celui-ci. De plus, les “spammeurs” qui détournent les techniques de tri par popularité en multipliant l'apparition de termes à des endroits stratégiques contraignent les moteurs de recherche à une vigilance constante et à développer de nouveaux algorithmes.

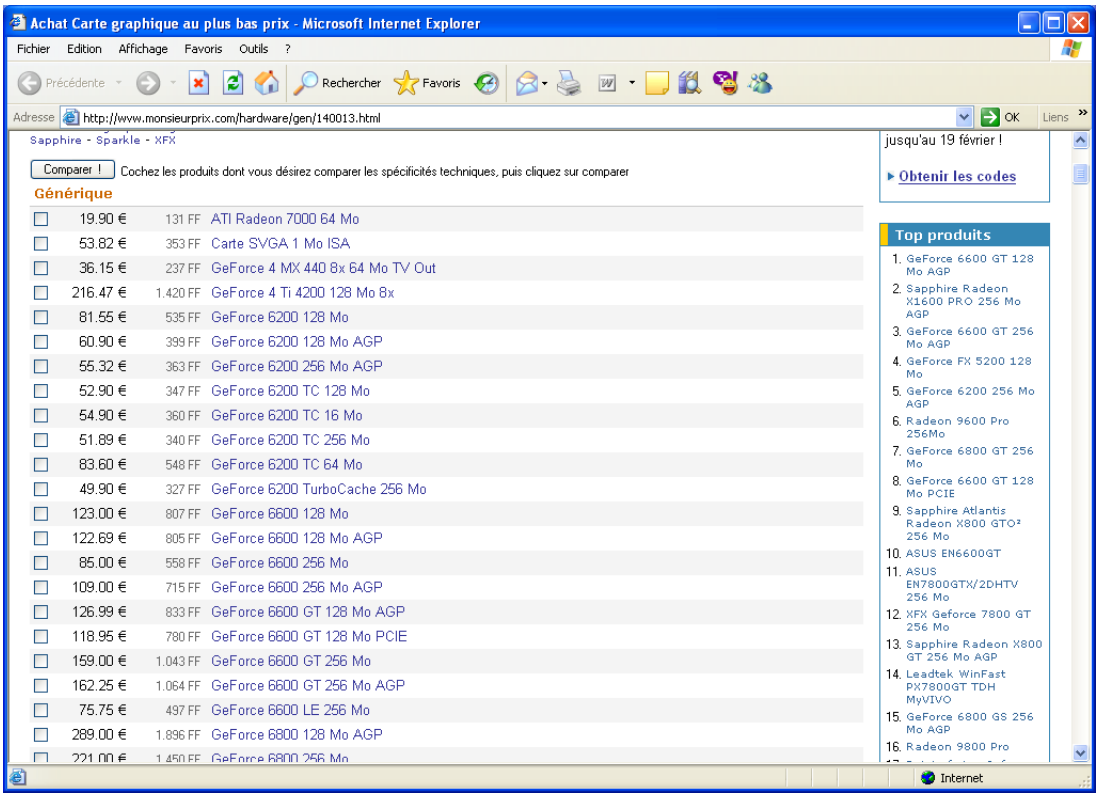

FIG. 1.2 – Exemple d'un site classé premier par ordre de pertinence pour "carte graphique"

1.2 Problème sémantique

Dans cette section, nous étudierons les exemples de recherche en testant les connaissances linguistiques chez Google.

EXEMPLE

Lors d'une recherche des enseignants de l'université de l'USTHB avec la phrase "je cherche les enseignants de l'USTHB" (figure 1.3), nous avons obtenu les résultats suivants ; le premier résultat classé par ordre de pertinence détache complètement le mot "enseignant" du mot "recherche". Le deuxième résultat nous mène vers les recherches de correspondance entre les noms de ville. Le troisième résultat interprète le mot "recherche" comme la recherche pour une aide d'inscription.

Dans ces trois premiers résultats on ne trouve pas le mot USTHB, et ce n'est

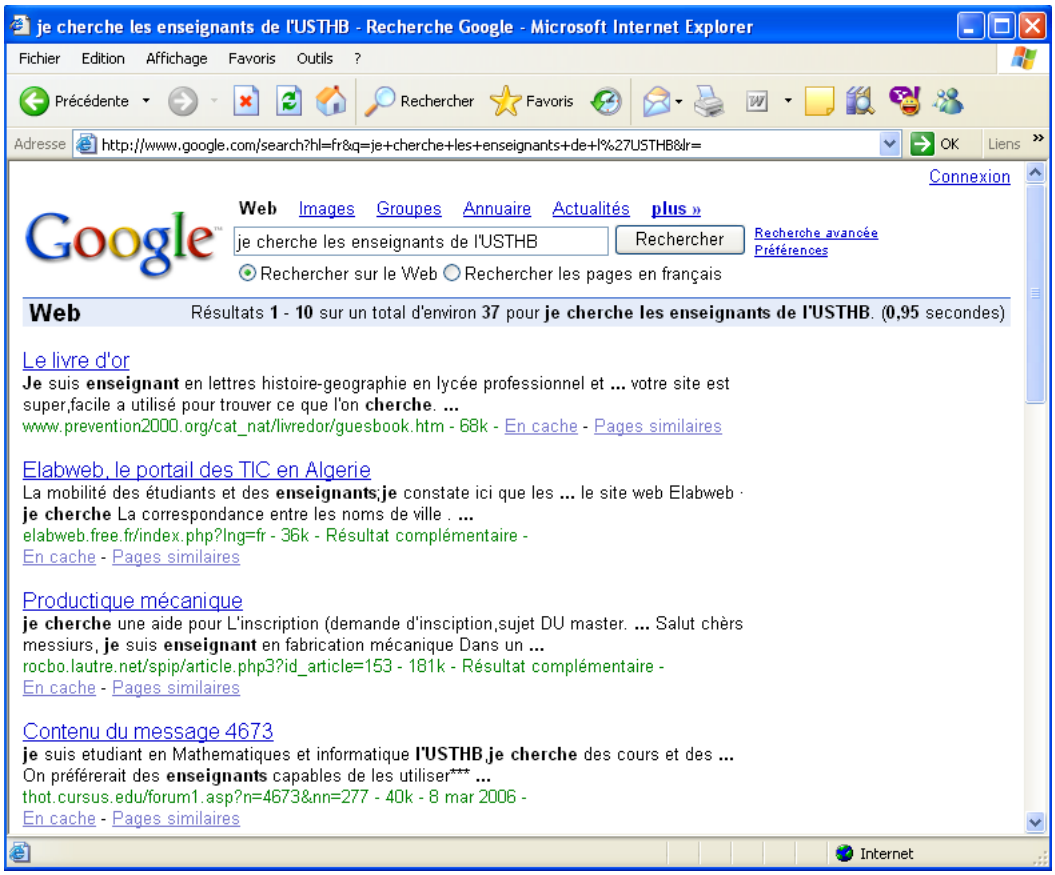

FIG. 1.3 – Exemple de recherche sémantique dans Google

que dans le quatrième résultat qu'on le trouve mais, cette fois-ci, in n'est pas associé à la recherche d'enseignant mais plutôt à une description d'un étudiant qui cherche des cours, tandis que le mot enseignant est utilisé dans un autre contexte complètement à part.

On peut classer la faille dans cet exemple de recherche en deux catégories :

- Le bruit : confusion dans le contexte de la recherche pour les mots "recherche", "USTHB" et "enseignant".
- Le silence : pas de recherche des enseignants de l'USTHB, aucun nom d'enseignant n'a été trouvé. De plus, aucune correspondance entre ce que l'on cherche et ce qu'on a eu comme résultats.

1.3 Motivation

Afin d'atteindre le degré d'intelligence souhaité et indispensable, le Web doit impérativement changer pour qu'il devienne plus basé sur la sémantique, c'est ce qu'on appelle le Web sémantique. Le Web sémantique est un domaine de recherche particulièrement actif aujourd'hui et très vaste, pouvant toucher le champ de l'Intelligence Artificielle en prenant appui sur bien d'autres domaines tels que les bases de données, l'apprentissage automatique, etc.

Le Web actuel est conçu pour être interprété par des humains. Le langage humain, et en particulier l'information présente sur le Web, ne présente pas de signification pour les machines, car il est non formel. Il est donc nécessaire d'ajouter au Web de l'information formelle qui soit interprétable par les machines. Toutefois, les êtres humains ont en général des difficultés à maîtriser parfaitement les langages formels, seuls interprétables par les machines. Il est donc nécessaire de concevoir des systèmes informatiques capables de communiquer avec les êtres humains de façon non formelle, en langage naturel.

De ce qui précède on déduit que, contrairement à l'homme, la connaissance pour un système informatique se limite à la connaissance qu'il peut représenter. Chez l'homme, les connaissances représentables sont complétées par des connaissances non exprimables (sensations, perceptions, sentiments non verbalisables, connaissances inconscientes, etc.). Ces éléments non représentables participent pourtant aux processus de raisonnement et de décision.

Les ontologies informatiques sont des outils permettant en outres de représenter un corpus de connaissances sous une forme utilisable par une machine. En informatique, la notion la plus couramment citée est celle de T. Gruber : "an explicit specification of a conceptualization" [GRU93a]. Une ontologie regroupe ainsi les définitions d'un ensemble structuré de concepts. Ces définitions sont traitables par machine et partagées par une communauté de personnes. Elles doivent, en plus, être explicites, c'est-à-dire que toute la connaissance nécessaire à leur compréhension doit être spécifiée.

Le langage d'ontologie Web le plus récent est OWL [BCH03, BCH04, DS04, DSBH05, HAY04, HP03a], qui est la standardisation donnée par le W3C à DAML + OIL [CON01] issue de la fusion des deux langages DAML-ONT [CON00] et OIL [FHM01], offre aux machines une plus grande capacité d'interprétation du contenu web que RDF et RDFS, grâce à un vocabulaire plus large et à une vraie sémantique formelle [HP03a, RDF00, RDF99, GRA04].

1.3.1 Limites des langages basés sur la LD

OWL et son sous langage OWL-DL basé sur la logique de description excluent, explicitement, les définitions par défauts et les exceptions, comme c'est le cas pour tous les formalismes basés sur la logique. Toutefois, comme peu de concepts sont définissables en utilisant seulement la connaissance stricte (et non pas par défaut), l'utilisation des définitions strictes induit des bases de connaissance terminologiques dont la majorité de ses concepts sont partiellement définis.

Par ailleurs des recherches ont été concentrées sur le problème des définitions par défaut et exception au sein de la logique de description en se basant sur le langage KL-ONE [COU97].

1.4 Contribution

Notre travail s'inscrit dans ce cadre et consiste en l'amélioration du langage d'ontologie Web basé sur la logique de description (OWL-DL) afin d'assurer une représentation terminologique pour tous les concepts utilisés dans l'ontologie. Dans ce contexte, nous avons étudié les solutions proposées pour le langage KL-ONE permettant la définition des concepts avec les définitions par défauts et les exceptions, et nous avons essayé de les projeter sur le langage d'ontologie Web OWL. Ceci nous a amené à proposer un nouveau langage $\text{OWL}_{\delta\epsilon}$, où δ et ϵ correspondent respectivement aux notations default et exceptions. Deux constructeurs sont alors ajoutés dont les représentations en RDF seront présentées.

Nous adoptons une étude comparative entre la représentation dans la logique de description et dans $\text{OWL}_{\delta\epsilon}$. Nous montrons, par proposition et preuve, que la complexité de l'algorithme de subsomption proposé est polynômiale. De plus, nous proposons des solutions pour résoudre les conflits de définition de concepts lors de l'héritage.

1.5 Organisation

Dans le chapitre suivant (chapitre 2) nous introduisons la recherche d'information (RI). D'une manière générale, nous établissons une étude comparative entre les domaines du traitement automatique des langages naturelles (TAL) et les domaines de la RI. À partir de cette comparaison, nous déduisons les techniques du TAL pour la recherche d'information et nous présentons la nouvelle tendance dans la RI.

Le chapitre 3 donne une introduction sur les logiques de description pour le Web sémantique. Après avoir défini la notion du Web sémantique et les ontologies, nous étudions les constructeurs des LDs, les langages terminologiques et assertionnels ainsi que les mécanismes de raisonnement.

Le chapitre 4 présente les ontologies Web basées sur la logique de description. Nous commençons par définir les différentes caractéristiques du langage RDF et son extension RDF schema. Ensuite, nous présentons les influences qui ont mené à la conception de OWL tout en étudiant ses constructeurs et leur utilisation.

Le chapitre 5 traite les limites de OWL. Il pose le problème des concepts partiellement définis et propose la solution pour remédier à ce problème en adoptant le raisonnement par défaut et exception. Nous discutons dans ce chapitres des initiatives prises dans le cadre de la logique de description sur lesquelles nous nous basons dans l'extension du langage OWL pour le nouveau langage $\text{OWL}_{\delta\epsilon}$. Dans ce denier, nous proposons un système equational et un algorithme pour la subsomption. Nous étudions aussi le calcul de complexité avec proposition et preuve et nous discutons des conflits possibles.

Le chapitre 6 conclue sur ce mémoire avec une vision de perspectives et propose quelques axes de recherche dans le domaine du Web sémantique.

Chapitre 2
LA RECHERCHE D'INFORMATION

2.1 Introduction

La recherche d'information est l'un des aspects du langage traité par le traitement automatique des langues (TAL). En effet, le traitement automatique des langues et la recherche d'information sont deux disciplines dont l'interaction, déjà identifiée depuis longtemps, s'est renforcée ces dernières années.

La recherche d'information (RI) est un domaine historiquement lié aux Sciences de l'information et à la bibliothéconomie qui ont toujours eu le souci d'établir des représentations des documents dans le but d'en récupérer des informations, à travers la construction d'index. L'informatique a permis le développement d'outils pour traiter l'information et établir la représentation des documents au moment de leur indexation, ainsi que pour rechercher l'information. On peut aujourd'hui dire que la recherche d'information est un champ transdisciplinaire, qui peut être étudié par plusieurs disciplines, une approche qui devrait permettre de trouver des solutions pour améliorer son efficacité. Au sens large, la recherche d'information inclut deux aspects :

- l'indexation des corpus, et
- l'interrogation du fond documentaire ainsi constitué.

La recherche d'information, dans la mesure où elle travaille aussi sur des textes, s'apparente au TAL [Kar71]. Par conséquent, pour une meilleure prise en compte de la nature linguistique du contexte recherché, le TAL s'impose clairement dans le cadre d'amélioration des résultats de la RI.

Dans un premier temps de ce chapitre, nous étudierons les domaines du TAL. Ensuite, nous étudierons les domaines de la RI où nous présenterons les principes généraux de le recherche d'information ; simplification des documents, l'indexation, traitement et évaluation des résultats des requêtes avec les formules de calcule du

bruit et du silence. Après avoir discuté des techniques du TAL pour la RI, nous présenterons la nouvelle tendance pour la recherche d'information.

2.2 Les domaines du TAL

Le traitement automatique des langues[1] [JAC96, LBB89] est une discipline à la frontière de la linguistique et de l'informatique, qui concerne l'application de programmes et techniques informatiques à tous les aspects du langage humain. Dans la suite, nous discutons les grands domaines du TAL en nous appuyant sur les domaines traitant la syntaxe et la sémantique. Nous ne prétendons pas faire un état de l'art sur le TAL, pour une étude plus détaillée sur le domaine nous conseillons par exemple [JAC96, LBB89, Kar71].

La morphologie concerne l'étude de la formation des mots et de leurs variations de forme ;

La syntaxe (section 2.2.1) s'intéresse à l'agencement des mots et à leurs relations structurelles dans un énoncé ;

La sémantique (section 2.2.2) se consacre au sens des énoncés ;

La pragmatique prend en compte le contexte d'énonciation.

2.2.1 Le domaine de la syntaxe dans le TAL

Pour repérer quels mots fonctionnent ensemble dans une phrase, un premier niveau de modélisation consiste à constituer des classes de mots (catégories syntaxiques, parties du discours) possédant un fonctionnement similaire : Nom (N), Verbe (V), Adjectif (A), etc.

Certaines unités, par accident (homographes : "le", "est") ou de façon plus systématique ("vacataire" : A → N, "étudiant" N → "étudier" V → "étudiant" V), peuvent être ambiguës entre plusieurs catégories (ambiguïté catégorielle ou lexicale). Par exemple, chacune des unités de la phrase "l'étudiant est vacataire" est ambiguë, ce que l'on peut noter :

$$\text{“}L'_{/DET,N,PRO}\ étudiant_{/N,V}\ est_{/A,N,V}\ vacataire_{/A,N}\ \text{”}$$

[1] On trouve aussi Traitement automatique du langage naturel/des langues naturelles (TALN).

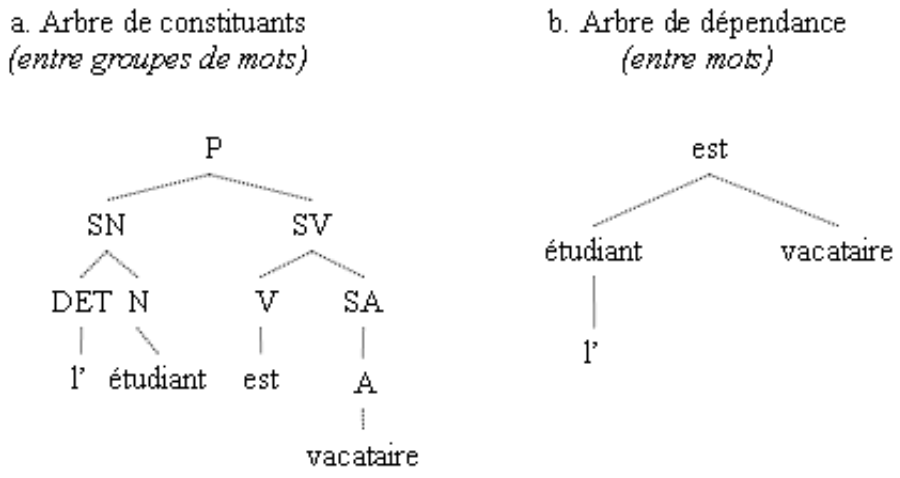

FIG. 2.1 – Représentations syntaxiques d'une phrase

On remarquera que dans le contexte de la phrase entière, aucune de ces unités n'est ambiguë.

Définition

Syntagme : Mot ou groupe de mots constituant un élément grammatical de la phrase.

Les relations syntaxiques entre les mots d'une phrase peuvent se représenter de plusieurs façons. Le modèle en constituants considère des groupes de mots, ou syntagmes, généralement centrés sur un mot de tête (N, V, etc.), et les modélise par des catégories spécifiques (syntagme nominal ou SN, syntagme verbal ou SV, syntagme adjectival ou SA, etc.). Ces syntagmes peuvent eux-mêmes être éléments d'autres syntagmes, et la structure d'une phrase est alors un arbre de constituants (figure 2.1).

Le modèle en dépendance considère directement les mots de tête (recteurs, ou régissants), et leur attache les mots qui en dépendent (régis). La structure d'une phrase est alors un arbre de dépendance (figure 2.1). Des équivalences existent entre les deux modèles.

Même sans ambiguïté lexicale, une phrase peut donner lieu à plusieurs structures syntaxiques (ambiguïté structurelle). Un exemple classique est la phrase "je vois un homme avec un télescope" (figure 2.2), dans laquelle "avec un télescope" peut désigner la manière dont je vois l'homme (2.2-a, attachement au verbe "vois", complément circonstanciel de manière) ou au contraire une caractéristique de l'homme

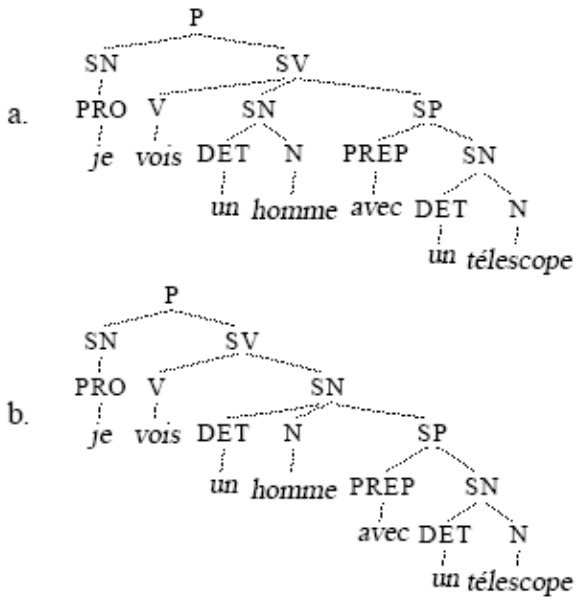

FIG. 2.2 – Ambiguïté structurelle

(2.2-b, avec un attachement au nom “homme”, complément de nom). Des informations sémantiques, voire pragmatiques (comme ce serait le cas ici), sont nécessaires pour déterminer l'interprétation la plus appropriée de ce genre de phrase.

Des relations plus précises entre mots ou syntagmes sont utiles à l'interprétation des phrases. Les relations grammaticales classiques (sujet-verbe, verbe-objet, verbe-objet-indirect, nom-modifieur, etc.) permettent de représenter la fonction des groupes de mots les uns par rapport aux autres. Les relations entre pronom et antécédent, et plus généralement entre anaphore (pronom, mais aussi nom) et antécédent, mobilisant encore davantage sémantique et pragmatique, assurent des mises en relation qui peuvent se situer à distance plus grande et qui sont très utiles en recherche d'information.

Les propriétés intrinsèques des mots restreignent le type de relations syntaxiques qu'ils peuvent avoir. C'est en particulier le cas des verbes, qui régissent ou sous catégorisent de zéro à trois ou quatre “arguments” :

“Il pleut.” pleuvoir()
“Hichem dort.” dormir(X)
“Hichem rédige son mémoire.” rédige(X, Y)
“Hichem présente le mémoire aux jurys.” présente(X, Y, Z)

2.2.2 Le domaine de la sémantique dans le TAL

De même que pour la syntaxe, un premier niveau de modélisation consiste à constituer des classes de mots (*catégories sémantiques*). Ces classes regroupent des mots dont le sens est proche, ou au minimum (pour des classes générales) des mots qui possèdent certaines propriétés sémantiques communes.

Cependant, si en syntaxe on arrive à s'accorder sur des jeux de catégories relativement consensuels (il s'agit d'une vue d'ensemble ; de près, le tableau est beaucoup plus polychrome) [PAR98], en sémantique aucune classification universelle n'existe (la constitution d'une classification universelle risque même d'être théoriquement impossible). Les classifications que l'on pourra utiliser (par exemple, les catégories générales de WordNet [MIL90, FEL98]) reflètent nécessairement un point de vue, une prise de position culturelle ou ontologique spécifique.

Un mot, même syntaxiquement non ambigu, pourra posséder plusieurs sens. Par exemple, on pourra distinguer l'"artère"-vaisseau sanguin de l'"artère"-avenue, même si le second est un sens figuré du premier. Le contexte permet en général de déterminer quel sens est à l'oeuvre dans un énoncé.

Dans un énoncé, les relations grammaticales sont le support de relations sémantiques syntagmatiques. Par exemple, les différents actants d'un événement jouent différents rôles thématiques : agent, thème, source, destination, etc. Ainsi, dans "Hichem présente le mémoire aux jurys.", les rôles par rapport à l'événement "présente" pourront être :

"$Hichem_{/agent,source}$ *présente le mémoire*$_{/thème}$*aux jurys*$_{/destination}$"

Un mot qui désigne un événement possède des propriétés combinatoires restreintes : il sélectionne comme actants certains types de mots (restrictions de sélection). Ces types restreints peuvent être exprimés en termes de classes sémantiques. On pourra par exemple poser pour le verbe "présenter quelque chose à quelqu'un" présenter (animé, objet, animé).

La représentation sémantique finale que l'on vise à associer à un énoncé dans un système de TAL dépend de l'objectif de ce système. Cet objectif peut être l'extraction d'informations spécifiques (section 2.3.2), comme c'est le cas dans les tâches définies dans les campagnes MUC (Message Understanding Conferences) d'évaluation de systèmes d'analyse de textes [Kauf93, Kauf95]. Pour une recherche d'un éventail d'informations plus large, la représentation doit être plus complète, comme dans le système MENELAS [PIE95].

2.3 Les domaines de la Recherche d'Information

Nous discutons rapidement des principes généraux de la recherche d'information (RI) (section 2.3.1). Nous passons ensuite en revue différents raffinements qui peuvent lui être apportés. Nous présentons finalement la notion d'extraction d'information, une tâche du traitement automatique des langues proche de la RI (section 2.3.2).

2.3.1 Principes généraux de la recherche d'information

La recherche d'information cherche des documents répondant à un besoin informationnel, ou sujet (figure 2.3), exprimé à l'aide d'une requête. Les documents sont au préalable indexés : chaque mot de chaque document est répertorié dans une table inverse, avec ou sans conservation des positions des mots dans le texte d'origine. L'appariement entre la requête et l'index va déterminer les documents qui sont considérés comme répondant le mieux au besoin informationnel initial.

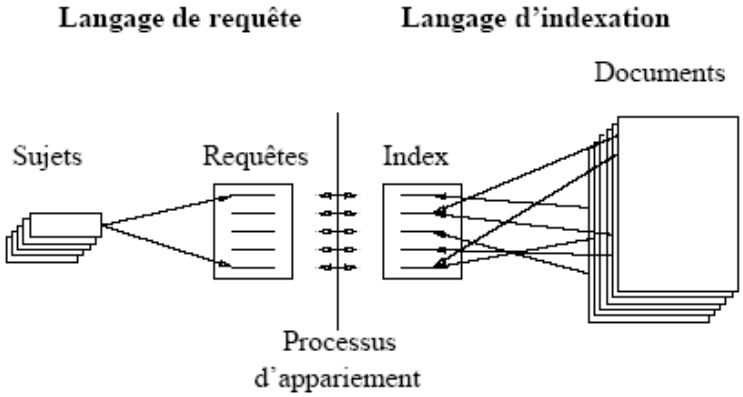

FIG. 2.3 – Schéma général de la recherche d'information

Une extension de ce schéma permet d'effectuer de la recherche d'information interlangue : le sujet de recherche est formulé dans une langue (par exemple, français) différente de celle des documents (par exemple, anglais).

Dans ce cas, le système de RI inclut une étape de traduction du sujet en une requête dans la langue cible. Les documents trouvés peuvent en retour être également traduits dans la langue source.

2.3.1.1 Simplification de documents

Avant d'être traités, la requête comme les documents sont "simplifiés". Cette simplification vise à rendre plus pertinent et plus efficace le processus d'appariement entre requête et index. Elle s'effectue selon les étapes suivantes :

1. Suppression des "mots stop" (mots grammaticaux, mots fréquents, mots sans pouvoir discriminateur...),
2. Racinisation, Stemming (section, réduction des mots de la même famille morphologique à une racine commune),
3. Transformation du texte en un ensemble de mots,
4. Amalgame (conflation) des mots synonymes.

Dans un modèle d'appariement sur la base de mots communs, tel que le modèle vectoriel, la suppression des mots fréquemment partagés tend à éloigner et à mieux séparer les documents considérés comme différents. A l'inverse, le regroupement des mots synonymes ou co-occurrents tend à rapprocher les documents semblables.

Le modèle vectoriel

Dans un système de recherche documentaire, le modèle vectoriel introduit par [SM83] est généralement utilisé. Dans ce modèle, chaque document de la collection initiale est représenté par un ensemble de termes extraits lors d'une phase d'indexation. C'est le modèle que nous utiliserons dans cette étude. Un poids est attribué à tous les termes de chaque document. Un document est alors représenté par un vecteur dont la dimension est le nombre de termes différents de la collection et dont les composantes représentent les poids des termes présents dans le document. L'ensemble de ces vecteurs forme ainsi une matrice dite " documents/termes ". Dans un processus de recherche documentaire classique, la requête (composée de plusieurs termes) est également représentée par un vecteur du même espace. Ce vecteur est alors comparé à tous les vecteurs de la matrice à l'aide d'une fonction de similarité (ou d'une distance). Le calcul de toutes les mesures de similarité entre la requête et l'ensemble des vecteurs de document va permettre par la suite d'ordonner les documents en fonction de leur ressemblance avec la requête.

2.3.1.2 Indexation

L'indexation peut se faire sur mots simples ou sur syntagmes. Dans ce dernier cas, des groupes de mots constituent des index du document. Ces syntagmes peuvent être obtenus par des techniques symboliques : par étiquetage et filtrage sur la base de patrons syntaxiques (spécification d'une structure syntaxique plus ou moins précise) ou par analyse syntaxique de surface (section 2.4.1.2). Ils peuvent aussi être obtenus par des techniques statistiques, en étudiant les mots co-occurrents dans des documents ou dans des fenêtres ; ou grâce à des patrons appris sur corpus et combinant des informations syntaxiques et lexicales.

2.3.1.3 Traitement et appariement des requêtes

On retrouve les mêmes traitements que sur les documents. Cependant, en raison de leur petite taille, les requêtes peuvent être analysées par des procédures plus lentes et complexes que celles traitant les documents ; et en raison de leur syntaxe pauvre, les requêtes sont analysées par des procédures symboliques aux contraintes syntaxiques lâches. Une fois traitées, les requêtes sont appariées avec l'index des documents.

Le modèle booléen suit une approche du type base de données : les documents sont recherchés sur la base d'une formule logique sur les descripteurs, et les réponses sont de la forme Oui/Non. C'est le modèle classique en recherche bibliographique, où l'on interroge sur le contenu des champs Auteur, Titre, etc. Dans le modèle vectoriel, plus un document partage des descripteurs avec la requête, meilleur il est. Les documents sont présentés par ordre décroissant de proximité avec la requête : les réponses sont qualifiées par un pourcentage exprimant leur pertinence. Le modèle probabiliste effectue un apprentissage sur les documents : il complète le modèle vectoriel en calculant la pertinence de chaque index pour un document en fonction de documents répondant à des requêtes sur une base documentaire comparable. Ici aussi, un pourcentage quantifie la pertinence des réponses.

2.3.1.4 Evaluation des résultats d'une requête (Pertinence)

Un système de recherche fournit donc, pour une requête donnée par l'utilisateur, un ensemble de documents dont il est nécessaire d'évaluer la pertinence (relevance en anglais). Cette évaluation se fait par comparaison entre les réponses trouvées et celles considérées comme idéales.

Il existe dans la littérature différents critères d'évaluation très largement utilisés pour mesurer la qualité (ou pertinence) d'une recherche. Ces critères sont intéressants pour mesurer la qualité d'une classification, par exemple, la précision qui mesure la quantité de documents pertinents retrouvés.

Dans le cas d'un système de recherche, une liste triée par ordre de pertinence de documents est proposée à l'utilisateur à partir du corpus de travail. Cette liste contient généralement des documents pertinents et non pertinents pour une requête donnée. Dans cette liste, certains documents pertinents peuvent être oubliés : le système n'a pas pu les retrouver.

Ainsi, pour une requête donnée, le corpus se divise en quatre catégories de documents (voir figure 2.4), l'objectif principal étant de retrouver le maximum de documents pertinents et un minimum de non pertinents. Un autre objectif est de retrouver tous les documents pertinents dans les premiers résultats. D'un autre point de vue (celui des critères), il s'agit de diminuer le nombre de documents non pertinents trouvés ainsi que les documents pertinents non trouvés.

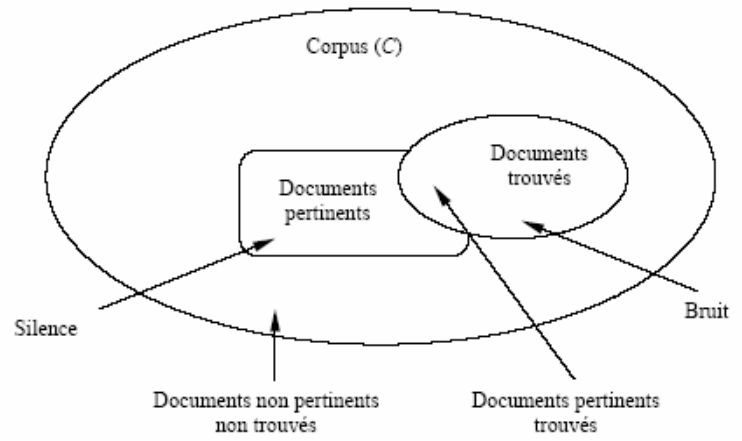

FIG. 2.4 – Pertinence : découpage du corpus pour une requête donnée

Définition du bruit à partir de la précision

La précision est le rapport entre le nombre de documents pertinents trouvés et le nombre de documents trouvés.

$$précision = \frac{\#documents\ pertinents\ trouvés}{\#documents\ trouvés}$$

Il est possible de déterminer la précision à différents niveaux : la précision sur les dix premiers documents de la liste par exemple. En effet, il est intéressant de déterminer la précision sur les n premiers documents présentés à l'utilisateur. Ce dernier ne regarde généralement que la première page de résultats d'une requête [SHM98].

La Fig 2.5 suivante montre un exemple de courbe de précision et la courbe optimale. La courbe optimale est horizontale sur l'intervalle [0, P] où au-delà de P, la courbe décroît : il ne reste que des documents non pertinents.

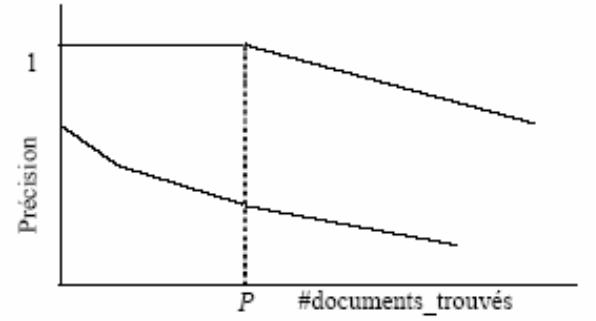

FIG. 2.5 – Exemple de la courbe de précision et de la courbe optimale

Le bruit est le rapport entre le nombre de documents pertinents non retrouvés et le nombre total de documents trouvés. Par conséquent, le bruit est défini comme suit :

$$bruit = 1 - précision$$

Définition du silence à partir du rappel

Le rappel ne tient compte que du nombre de documents pertinents trouvés par rapport au nombre de documents pertinents pour une requête donnée.

$$rappel = \frac{\#documents\ pertinents\ trouvés}{\#documents\ pertinents}$$

Le silence, qui correspond aux documents retrouvés non pertinents, est défini comme suit :

$$silence = 1 - rappel$$

2.3.2 Extraction d'information

Alors que la recherche d'information recherche des documents (ou des "passages" de documents), l'extraction d'information vise à extraire des informations spécifiques et structurées d'un texte sur un domaine particulier. Elle a été popularisée par une série de compétitions organisées entre systèmes d'analyse de textes par l'agence américaine DARPA, les conférences MUC (*Message Understanding Conferences*), qui ont également promu une évaluation rigoureuse des systèmes de traitement automatique des langues.

2.4 Techniques de TAL pour la Recherche d'Information

Les techniques du TAL qui ont un impact actuel ou attendu sur la recherche d'information sont les suivants :

Palier morphologique

- Segmentation en unités linguistiques
- Racinisation

Palier syntaxique

- Etiquetage ou désambiguïsation syntaxique
- Analyse "peu profonde", ou "surfacique"
- Indexation sur les syntagmes et variation
- Reconnaissance des entités nommées

Paliers sémantique et pragmatique

- Etiquetage sémantique
- Résolution d'anaphores

Techniques transversales

- Statistiques textuelles
- Traduction automatique et recherche d'information interlangue

Nous avons vu dans la section 2.2.1 (resp. section 2.2.2) le domaine syntaxique (resp. sémantique). De même, nous allons parler dans la section 2.4.1 et la section 2.4.2 respectivement de l'impact des paliers syntaxique et sémantique sur la recherche d'information.

2.4.1 Palier syntaxique

2.4.1.1 Etiquetage ou désambiguïsation syntaxique

L'étiquetage syntaxique, ou désambiguïsation syntaxique, vise à associer à chaque mot, en contexte, une "étiquette" syntaxique. Cette étiquette indique la catégorie syntaxique et éventuellement les traits morphosyntaxiques du mot. Par exemple, "$L'_{/DET,N,PRO}$ $étudiant_{/N,V}$ $est_{/A,N,V}$ $vacataire_{/A,N}$ ".

L'étiquetage syntaxique est une étape intermédiaire de nombreux systèmes d'analyse surfacique ou partielle, c'est pourquoi nous le présentons ici. La plupart des méthodes cherchent à obtenir cet étiquetage en examinant le contexte immédiat du mot à étiqueter (quelques mots à gauche et à droite).

Les méthodes à base de règles appliquent aux mots ambigus des règles de désambiguïsation, qui (selon la méthode) interdisent ou autorisent sélectivement certaines séquences d'étiquettes [CHA95, SIL93]. Les méthodes probabilistes apprennent des modèles de Markov cachés sur des corpus préalablement étiquetés [WEI93]. La méthode de Brill [BRI92, BRI95] apprend sur un corpus étiqueté des règles de correction d'erreurs d'étiquetage. Enfin, l'application d'un véritable analyseur syntaxique sur une phrase a pour effet de bord de désambiguïser les mots de la phrase [VER99]. Ce dernier type de méthode n'est réellement utile dans le contexte de l'étiquetage que si l'analyse syntaxique appliquée n'a pas une complexité trop grande.

Le choix des étiquettes, et en particulier leur finesse, conditionne les performances des étiqueteurs, qui atteignent 90-98 étiquetés selon le jeu de catégories, le corpus, etc. La taille limitée du contexte examiné pour effectuer la désambiguïsation place une limite théorique sur la précision de l'étiquetage effectué [VER98]. Par ailleurs, la plupart des mots peuvent changer de catégorie syntaxique (conversion d'un adjectif en nom, etc.) ; de ce fait, il est difficile de supposer que toutes les catégories syntaxiques possibles d'un mot se trouvent dans le lexique utilisé.

2.4.1.2 Analyse "peu profonde", ou "surfacique"

Depuis le milieu des années 1980, le modèle d'analyse syntaxique dominant, fondé sur l'emploi de formalismes grammaticaux élaborés et d'analyseurs mettant en oeuvre ces formalismes, a été sérieusement concurrencé dans l'analyse de grands

documents par des méthodes d'analyse simplifiées. Ces méthodes, au moins en première intention, visent des analyses moins "profondes" ou moins complètes que les précédentes.

L'analyse partielle ne cherche pas à traiter l'ensemble d'une phrase, mais seulement à analyser certains segments utiles et potentiellement plus faciles à reconnaître (syntagmes nominaux, syntagmes non récursifs et autres "chunks" [ABN91]). Une méthode souvent employée est l'identification de patrons syntaxiques (typiquement, automates à états finis) dans des textes préalablement étiquetés (voir section 2.4.1.1).

Une stratégie d'analyse robuste fait en sorte de toujours donner un résultat, même incomplet, pour l'analyse d'une phrase. Les analyseurs "classiques" peuvent en général se replier sur une analyse partielle lorsqu'une analyse complète n'est pas obtenue (par exemple, avec les méthodes "tabulaires"). Les analyseurs qui identifient progressivement des segments de phrases "sûrs" et les relations entre ces segments sont par nature robustes.

Enfin, l'identification de *cooccurrences* (statistiques), obtenues en recherchant des mots se retrouvant fréquemment conjointement dans une fenêtre, un paragraphe, un document, peut constituer un substitut de l'analyse syntaxique pour détecter des syntagmes élémentaires.

2.4.1.3 Indexation sur les syntagmes et variation

Une fois que l'on a identifié des syntagmes, on peut s'en servir pour indexer les documents dans lesquels ils apparaissent. L'indexation sur les syntagmes ("phrase indexing") a pour but d'augmenter la précision des index en diminuant leur ambiguïté. L'identification des coocurrences est utilisée en RI pour faire de l'indexation sur des groupes de mots sans avoir recours à des techniques symboliques de TAL plus coûteuses à mettre en oeuvre.

En concurrence, on trouve des techniques d'analyse robuste et superficielle en TAL appliquées à l'indexation pour la RI [DEB82, FAG87, ARA97, ARA98]. Ces techniques doivent être capables de regrouper les variantes d'un syntagme de base qui peut être modifié ou transformé pour produire des syntagmes de sens proche. Il est utile de savoir reconnaître ces variations pour pouvoir apparier une requête qui contient l'une des formes avec un document qui en contient une variante [JAC96]. Par exemple, à partir du syntagme de base "diffusion de la lumière", on repérera "diffusions de la lumière", "diffusion dépolarisée de la lumière", "diffuse une lumière" et "émission de lumière". Ces variantes peuvent être obtenues par

génération dynamique de patrons de variantes (par exemple, à l'aide de métarègles) ou par simplification des structures syntaxiques des termes observés.

Parmi les enjeux de la reconnaissance de variantes, on peut citer la difficulté à couvrir exactement les variantes pertinentes et le coût computationnel de la production contrôlée de ces variantes.

2.4.1.4 Reconnaissance des entités nommées

La notion d'*entités nommées*, introduites dans le cadre de l'extraction d'information (section 2.3.2), se réfère à des concepts uniques et partagés. Les entités nommées comprennent les organisations (entreprises, administrations, musées, etc.), les lieux (villes, régions, fleuves, etc.), les personnes (hommes politiques, vedettes, chefs d'entreprise, inconnus, etc.) et les numériques (poids, longueurs, valeurs monétaires, pourcentages, etc.). Les entités nommées peuvent constituer des index très discriminants, et sont souvent des informations demandées. Par exemple, plusieurs entités nommées sont en jeu pour répondre à la question "*Quel est le nom de l'encadreur de Hichem en 2006 ?*".

La reconnaissance des entités nommées s'appuie sur des méthodes symboliques et numériques. Le premier type de méthode repose sur des dictionnaires (de nombreuses listes d'entités nommées sont accessibles en ligne : noms de lieux, annuaires divers, etc.) et des patrons syntaxiques. Ceux-ci sont appliqués sur des textes préalablement étiquetés (section 2.4.1.1) et peuvent utiliser des repères lexicaux internes (par exemple, unités pour les mesures) ou externes (par exemple, titres honorifiques pour les personnes) [MCD93, WAC97]. Le second type de méthode effectue un apprentissage de contextes et de structures, par exemple avec des modèles à apprentissage statistique comme les modèles de Markov cachés [BIK97, MIK99].

2.4.2 Paliers sémantique et pragmatique

2.4.2.1 Etiquetage sémantique

De même que l'étiquetage syntaxique (section 2.4.1.1) vise à associer à chaque mot une étiquette syntaxique, l'étiquetage sémantique cherche à associer à chaque mot, en contexte, une étiquette sémantique. Cette étiquette sémantique peut être une catégorie sémantique générale (par exemple, animé, événement, mouvement,

etc.) ou un sens de mot (par exemple, "artère"- vaisseau sanguin vs "artère"- avenue). Pour une partie des mots, la désambiguïsation syntaxique peut aider : on en sait davantage sur le sens de "livre" si l'on connaît son genre ("*un* $livre_{/Nms}$" vs "*une* $livre_{/Nfs}$"). Par ailleurs, des méthodes similaires à celles employées en syntaxe sont applicables (chaînes de Markov, etc.). Encore plus que pour les travaux en étiquetage syntaxique, le choix des étiquettes a une influence fondamentale sur la nature de la tâche. Les travaux en désambiguïsation sémantique sont relativement récents, mais possèdent une forte dynamique.

2.4.2.2 Résolution d'anaphores

Définition

Anaphore : Répétition d'un mot en tête de phrases.

La résolution d'anaphores consiste à relier entre elles les références à une même entité au sein d'un texte. On distingue plusieurs types d'anaphore. L'anaphore pronominale emploie un pronom pour faire référence à une expression antérieure : "Le stage est effectué au sein de l'USTHB. Il est inscrit dans le cadre de préparation du magistère.". L'anaphore par reprise partielle reprend une partie de l'expression antérieure, comme dans "... les modèles d'ontologies pour concevoir des ontologies WEB sont issus d'approches basées sur la logique. Ces approches exploitent l'expressivité de la logique ...". L'anaphore par lien sémantique ne reprend pas directement un mot de l'antécédent, mais un terme sémantiquement lié (ici, plus générique) : "Le centre de calcul est fabriqué de béton... Cet institut a été remplacé par un nouveau institut...".

De façon générale, la plupart des expressions nominales (syntagme nominal défini, pronom) sont potentiellement des anaphores et potentiellement des antécédents d'anaphores. La résolution d'anaphores requiert des informations aussi bien syntaxiques (genre, nombre) que sémantiques (relation d'hyperonymie, etc.) et s'appuie sur des considérations pragmatiques (entités les plus saillantes au fil du texte, ou "focus") [HIR81, CAR88, BOT96, MIT98].

La résolution d'anaphores est une technique dont l'apport est important dans de nombreuses applications. En extraction d'information, elle permet de garnir une structure d'information avec la référence initiale complète à une entité.

2.5 La nouvelle tendance dans la Recherche d'information

Le Web actuel basé sur une RI classique est conçu pour être interprété par des humains. Par contre, le langage humain ne présente pas de signification pour les machines, du fait qu'il n'est pas formel. Nous avons vu que l'utilisation du TAL peut fortement améliorer la qualité de recherche par son langage formel. Cependant, les êtres humains ont en général des difficultés à maîtriser parfaitement les langages formels, seuls interprétables par les machines. Aujourd'hui, la nouvelle tendance de la RI est de rendre cette recherche plus intelligente, ici la notion de l'intelligence artificielle s'impose.

Le terme intelligence artificielle couvre actuellement des sujets de recherche aussi variés que la robotique, le traitement de langage naturel, les réseaux neuronaux, les systèmes experts, etc. Parmi les aspects fondamentaux d'un système intelligent se trouve la représentation de la connaissance du domaine de l'application, les techniques de raisonnement que le système utilise pour inférer des nouvelles connaissances et la communication homme-machine, en particulier pour que le raisonnement suivi soit compris par l'utilisateur.

Depuis la naissance de l'intelligence artificielle, des techniques de représentation de connaissances et des mécanismes de raisonnement associés ont été développés. Parmi ces techniques et mécanismes, on voit apparaître les systèmes de logique classique qui raisonnent par des inférences logiques monotones, les systèmes à base de règles, les réseaux sémantiques [WOO75] et les graphes conceptuels [SOW84] qui permettent de représenter la connaissance par des graphes bipartites et les logiques terminologiques ou logique de description [NAP97, ATT86, BAA03].

Les logiques terminologiques[2] ont pour base les schémas et réseaux sémantiques. Ces systèmes s'appuient sur un modèle logique pour bâtir des représentations autour de la notion de concept structuré et pour organiser des concepts dans une hiérarchie de subsomption ; ils manipulent et mettent à jour la hiérarchie de concepts par un algorithme de classification de concepts, le "classifieur". Le précurseur de cette famille est le système KL-ONE développé par Brachman en 1978 [BRA85] et complété par un mécanisme de classification et une sémantique formelle en 1983 [SCH83]. Le souci de complétude d'une part et d'efficacité du raisonnement de l'autre ont motivé le développement de toute une famille de systèmes : certains ajoutent des éléments

[2]aussi appelés systèmes hybrides, systèmes de représentation de connaissances basés sur la classification (CBS), langages de subsomption de termes (TSL), langages terminologiques, systèmes de classification de termes (TC) ou encore famille de KL-ONE.

à la représentation pour augmenter l'expressivité, comme KRYPTON qui introduit les ABOXs ou paquets d'assertions, et LOOM [McGRE91] qui ajoute les règles et les contraintes ; d'autres augmentent l'efficacité de la classification, en réduisant les capacités d'expression du langage comme dans le cas du langage de bases de donnés CLASSIC [BRA91] ou en sacrifiant la complétude comme dans les langages NIKL [KAZ86] et BACK [NEB88].

2.6 Conclusion

L'accès au contenu des documents textuels est le domaine de la recherche d'information. Les moteurs de recherche disponibles sur le Web montrent les limites des techniques classiques de la RI. Nous avons vu comment les techniques du traitement automatique des langues peuvent jouer des rôles clé dans les tâches de RI, mais restent néanmoins de l'utopie malgré les énormes progrès dans ce domaine.

Nous avons vu que la nouvelle tendance cherchant à ramener le Web actuel à un Web plus intelligent, Web sémantique, se base sur l'apport du domaine de la RI et notamment les techniques parmi lesquels on trouve la logique de description (LD).

Dans le chapitre suivant, nous discutons des critères qui ont accompagné notre intérêt pour les LDs tout au long de ce mémoire. Les exemples qui seront ainsi présentés sont issue d'une recherche approfondie dans le cadre de contribution dans l'extension des langages basés sur la LD afin d'assurer une reconnaissance terminologique pour tous les concepts utilisés dans l'ontologie (discuté plus loin dans le chapitre 5).

Chapitre 3
LES LOGIQUES DE DESCRIPTION POUR LE WEB SÉMANTIQUE

3.1 Introduction

La conclusion à laquelle on est arrivé dans le chapitre précédent est que pour arriver à un degré de correspondance entre le langage machine et le langage naturel, il faut doter la RI de capacité d'intelligence plus évoluée . La tendance accompagnant cette idée est de changer le Web actuel en Web sémantique.

Pour arriver à un Web sémantique, les programmes doivent être capables de reconstituer des raisonnements et des actions intelligentes sur les données, cette nécessité a donné naissance au domaine des ontologies. Une ontologie a ainsi la charge de permettre à une machine de comprendre des données. Par cette capacité, les ontologies apparaissent comme le moyen le plus prometteur pour résoudre des problèmes dans plusieurs domaines et particulièrement la recherche sur le Web [BER01]. A cet effet, plusieurs langages de définition d'ontologies notamment ceux basés sur les logiques de description se sont développés (OWL par exemple).

Dans la section suivante, nous introduisons le Web sémantique et la notion d'ontologie. Nous étudions ensuite plus en détails les constructeurs et langages de la logique de description.

3.2 Introduction au Web sémantique

Selon Tim Berners Lee [BER01], le Web Sémantique doit permettre de porter le Web actuel à son plein potentiel. Le Web actuel est principalement tourné vers les êtres humains : le langage HTML est un langage de présentation et de formatage

de documents, et les machines ou agents informatiques se limitent à mettre en page les pages HTML. L'objectif du Web Sémantique est de rendre le contenu du Web compréhensible à des machines. Le procédé consiste à exploiter :

- des ontologies. Une ontologie est un vocabulaire constitué de concepts, de relations, voire d'axiomes, liés à un certain domaine. [GRU93b] définit une ontologie comme étant "la spécification d'une conceptualisation".
- un langage commun pour exprimer les ontologies et décrire des annotations utilisant les termes de ces ontologies,
- des moteurs de raisonnement permettant d'inférer sur les annotations d'après les axiomes déclarés dans les ontologies,

3.2.1 Les ontologies

Dans le cadre de notre travail, nous avons choisi la définition suivante : "la conceptualisation des objets reconnus comme existants dans un domaine, de leurs propriétés et de connaissances heuristiques associées". Une ontologie regroupe ainsi les définitions d'un ensemble structuré de concepts.

3.2.1.1 Structures de représentation

La conception d'une ontologie se fait en instanciant un modèle d'ontologies. De nombreux modèles existent, suivant différentes approches et présentant des constructeurs très différents. Ceux-ci peuvent cependant tous être associés à l'une des trois dimensions [PIE04] autour desquelles la connaissance peut s'articuler :

1. *La connaissance structurelle ;* les objets du domaine d'étude sont regroupés en classes organisées par des relations ; la plus couramment utilisée est la relation de subsomption ;
2. *La connaissance descriptive ;* les objets des différentes classes sont regroupés parce qu'ils présentent des caractéristiques communes ; la définition de ces caractéristiques est faite en utilisant des propriétés ;
3. *La connaissance procédurale ;* les caractéristiques d'une classe peuvent être déduites à partir d'autres informations ; elles peuvent également être soumises à des contraintes.

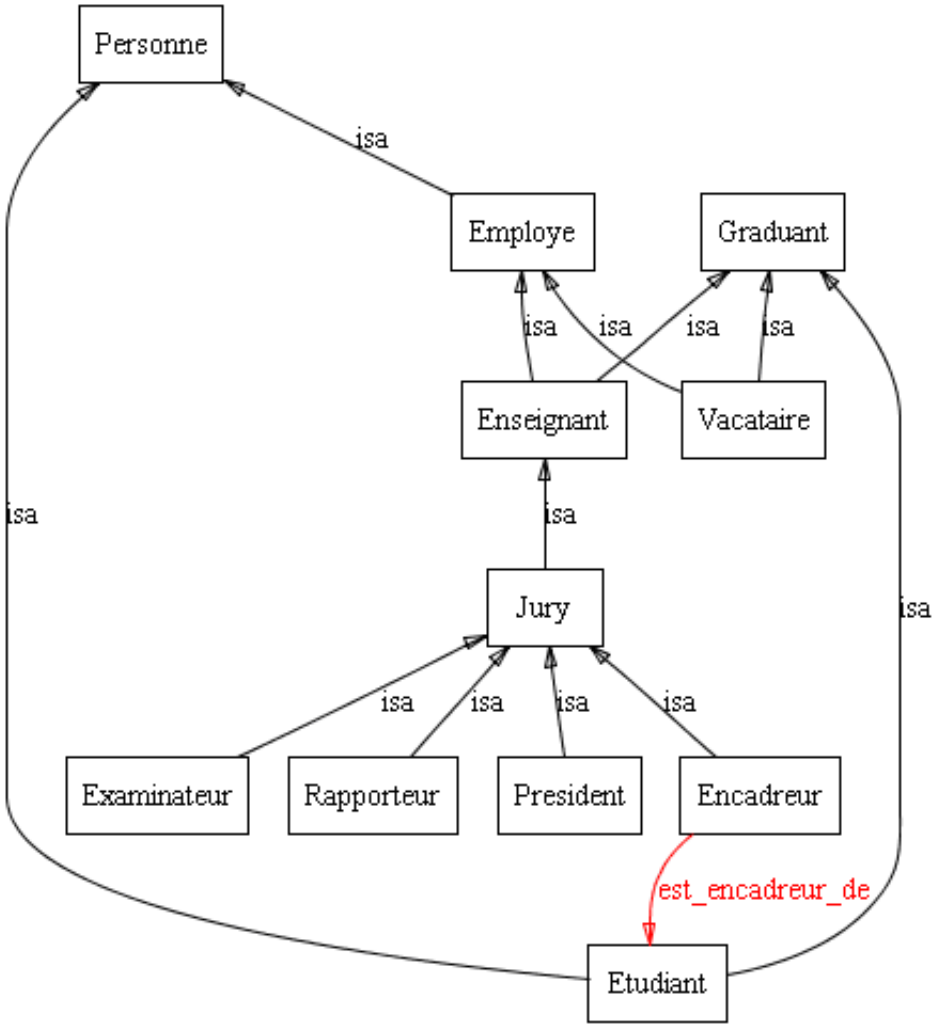

FIG. 3.1 – Modèle de représentation de l'ontologie Université

EXEMPLE

Par exemple, dans une ontologie pour une université, les constructeurs associés à la dimension structurelle nous permettront de regrouper l'ensemble des enseignants dans une classe Enseignant subsumée par la classe Employé. De son côté, la classe Employé est subsumée par la classe Personne. Aussi, nous pouvons ajouter la propriété qu'un étudiant est encadré par un encadreur qui fera parti de l'ensemble des jurys lors de la soutenance. On peut ajouter la contrainte qu'un étudiant doit finir son stage de magistère dans un délai maximum de deux ans.

La représentation de cette ontologie au sein de l'outil Protégé est montrée dans la figure 3.1. La relation de subsomption est représentée par le lien "isa" (est un).

3.2.1.2 Problèmes à résoudre avec les ontologies

Dans [GBMS99], quatre types de conflits ont été identifiés : conflits de représentation, conflits de nom, conflits de contexte et conflits de mesure de valeur.

Conflits de représentations (conflit structurel)

Ils se retrouvent quand des attributs différents ou des schémas différents sont utilisés pour décrire le même concept.

Conflits sémantiques

1. Les conflits de noms : se retrouvent lorsque les différents schémas utilisent des noms différents pour représenter le même concept ou propriété (synonyme), ou des noms identiques pour des concepts (et des propriétés) différents (homonymes).
2. Les conflits de contexte : se retrouvent dans le cas où les concepts semblent avoir la même signification dans deux schémas mais sont différents à cause de leur contexte. Par exemple le poids d'une personne dépend de la date où elle se pèse.
3. Les conflits de mesure de valeur : sont liés à la manière de coder la valeur d'un concept du monde réel dans un système de mesure. Ils se retrouvent par exemple dans le cas où on utilise des unités différentes pour mesurer la valeur d'une même propriété dans différentes sources (dans une source, on utilise le dollar, et dans une autre on utilise l'euro).

3.2.1.3 Les domaines d'utilisation d'ontologies

Traitement du langage naturel

La reconnaissance de similitudes conceptuelles entre des mots permet essentiellement d'améliorer la recherche documentaire. Dans les moteurs de recherche actuels, une requête est composée d'un ensemble de mots éventuellement connectés par les opérateurs logiques OU, ET et NON. Le moteur produira sa réponse en fonction des mots contenus dans les documents parcourus. L'utilisation d'ontologies par ces moteurs va améliorer sensiblement "la qualité" du résultat.

Web sémantique

La popularisation des accès haut débit au réseau Internet provoque un fort engouement sur ce média de communication. Autrefois réservé aux professionnels de l'informatique avec un contenu essentiellement scientifique, celui-ci est aujourd'hui accessible à tous pour une utilisation très variée. Que ce soit pour faire ses démarches administratives, faire ses courses en ligne, rechercher un emploi ou bien d'autres choses, tout le monde a perçu les bénéfices de ce vecteur de communication.

Cette formidable source d'informations souffre pourtant d'un défaut majeur qui décourage bien des débutants. Alors que l'on parle d'une toile pour décrire ce réseau, l'ensemble des services qui y sont offerts sont complètement isolés. En conséquence, pour arriver au résultat escompté, une personne doit soit avoir une connaissance approfondie du Web, soit passer par une longue période fastidieuse d'errance sur différents sites. L'évolution très rapide et incontrôlée des services conduits souvent à la deuxième solution même pour des personnes habitués du Web.

L'utilité majeure d'une ontologie est son exploitation par le moteur de recherche. L'affectation d'un document à une classe se fait par le calcul d'une mesure de similarité entre le document et le domaine de la plus proche.

La recherche d'information

Un des apports du Web Sémantique se situe au niveau de la recherche d'informations. Actuellement la recherche d'informations sur le Web est essentiellement basée sur des technologies plein texte, comme dans les principaux moteurs de recherche, par exemple Google et Altavista.

La recherche sur le Web Sémantique peut, en plus de l'exploitation plein texte comme dans le Web actuel, exploiter les annotations des documents et les ontologies. Cela permet d'accéder aux ressources selon leur contenu (si les annotations représentent le contenu) plutôt que par mots-clés. Une telle recherche d'informations, que l'on appelle 'guidée par les ontologies' ou 'sémantique' (voir figure 3.2), a déjà été étudiée dans de nombreux projets et pour différents langages, comme Shoe [HHS98] [LSRH97], Ontobroker [FENS98] (Frame Logic) ou WebKB [MAR97] (Graphes Conceptuels). Ces projets ont démontré l'utilité et l'apport de l'approche de la recherche d'informations sémantique. Nous préconisons l'utilisation de la logique de description.

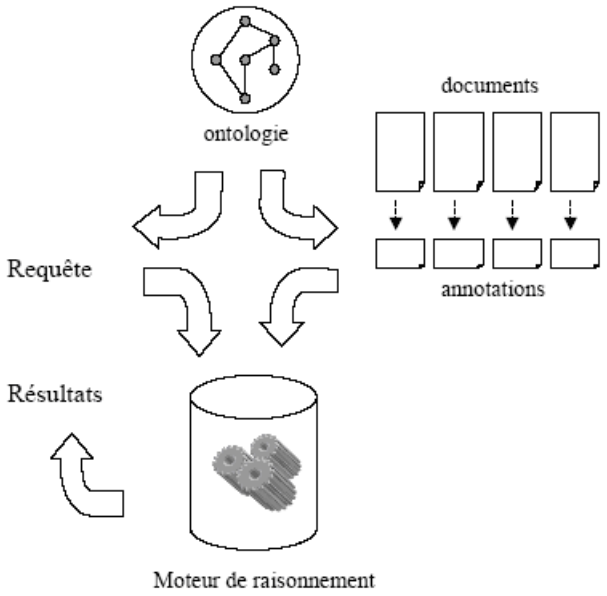

FIG. 3.2 – La recherche d'informations sémantique, ou guidée par les ontologies

3.3 Les logiques de description

Les logiques de description (LDs) découlent directement des travaux fondateurs de Brachmann et de son système KL-ONE [NAP97, ATT86, BAA03]. Depuis le début des années 90, la recherche en logique de descriptions s'est considérablement développée. Les logiques de description peuvent être considérées comme un fragment de la logique du premier ordre, dans lequel les formules ont une variable libre pour les descriptions de concepts et deux variables libres pour les descriptions de relations [BDS93]. Les logiques de description peuvent aussi être considérées comme des logiques multi-modales (car elles possèdent plusieurs relations d'accessibilité) propositionnelles.

3.3.1 Les constructeurs des LDs

Une LD est inductivement définie à partir d'un ensemble P_C de concepts primitifs, un ensemble P_r de rôles primitifs, des constantes $\top$ et $\bot$ et des règles de syntaxe suivantes du langage de description standard ALN :

$C, D \rightarrow \top \mid \bot$

$\mid P$	concept primitif
$\mid C \cap D$	conjonction de concepts
$\mid C \cup D$	disjonction de concepts
$\mid \neg C$	Négation
$\mid \forall r.C$	restriction universelle
$\mid \exists r.C$	restriction existentielle
$\mid \leq_n r.C$	cardinalité maximum
$\mid \geq_n r.C$	cardinalité minimum

$r \rightarrow q$

$\mid r_1 \cap r_2$	conjonction de rôles
$\mid r_1 \cup r_2$	disjonction de rôles
$\mid q^{-1}$	inverse de rôles
$\mid r_1 \circ r_1$	composition de rôles

Les constructeurs utilisés dans cette syntaxe déterminent la puissance d'expression de la LD ainsi définie. Par exemple, la description de concept suivante décrit toutes les personnes qui sont membre du jury pour les étudiants suivant des stages à l'université :

$$Person \cap \exists juryDe.(Etudiant \cup \exists suiviDans.Université)$$

Selon les applications considérées, les constructeurs sont plus ou moins utiles. Ainsi pour décrire des objets, les constructeurs de cardinalité maximum et minimum peuvent être parfois très utiles. Pour décrire des actions et des processus, les constructeurs de composition, de conjonction et de disjonction de rôles semblent nécessaires. Une logique de description contenant ces constructeurs est par exemple CIQ [DGL96].

Les logiques terminologiques incluent deux langages, le langage terminologique (TBox) et le langage assertionnel (ABox). Le langage terminologique sert à décrire les termes ou concepts du monde à partir de concepts plus simples et d'opérateurs de formation de structures ; le langage assertionnel décrit, à partir des termes, des propositions de croyance sur l'état du monde. Ainsi l'expression "un encadreur dont

les stagiaires sont des étudiants" décrit un terme tandis que la phrase "Hichem est un étudiant avec une taille de 1m80" indique une assertion, une proposition dont la valeur de vérité dépend du contexte d'évaluation.

3.3.2 Langage terminologique

Les logiques terminologiques structurent la connaissance autour des objets ; le principal type d'objet de représentation est le concept ; un concept est la description d'une structure potentiellement complexe. Un concept est composé d'un ensemble de rôles (propriétés, parties, etc.) et un ensemble de conditions structurelles qui expriment des relations entre les rôles. Le langage terminologique comporte un ensemble limité d'opérateurs ("spécialisation", "restriction", "différentiation", etc) qui permettent de définir un concept complexe à partir de concepts plus simples. Un concept représente une classe d'individus, la classe des individus qui satisfont la structure et les contraintes du concept. KL-ONE distingue deux types de concepts : les concepts génériques qui peuvent décrire plusieurs individus dans des contextes (états du monde) différents et les concepts individuels qui décrivent un seul individu dans un contexte précis et qui correspondent donc à une classe ayant un seul membre. A part cette distinction ensembliste de concepts, la plupart des logiques terminologiques distinguent les concepts primitifs des concepts définis. Les concepts primitifs établissent des conditions nécessaires mais pas de conditions suffisantes pour l'appartenance d'un individu au concept, tandis que les concepts définis donnent des conditions nécessaires et suffisantes.

Exemple

Le concept primitif "personne" peut être défini comme "être vivant et ayant un esprit" ; toute personne doit satisfaire ces contraintes mais le fait de les satisfaire ne suffit pas pour dire qu'il s'agit d'une personne ; par contre, la définition du concept défini étudiant_vacataire, donnée par : "étudiant_vacataire = étudiant + vacataire" indique que tout étudiant vacataire doit être un étudiant (satisfaire les contraintes d'étudiant) et a le poste du vacataire et de plus que toute personne étant un étudiant et ayant le poste de vacataire est un étudiant_vacataire. L'appartenance d'un individu à un concept primitif doit être établie par l'utilisateur, celle d'un concept défini est établie par le système.

3.3.2.1 La relation de subsomption

Les concepts sont organisés en une hiérarchie de généralisation induite par la relation de subsomption : un concept A subsume un concept B si l'ensemble

d'individus dénoté par A contient l'ensemble d'individus dénoté par B (A est appelé le subsumant et B le subsumé) [LEV87].

Par opposition à cette définition extensionnelle de la subsomption, des définitions logique et intensionnelle on été proposées [WOO91]. Du point de vue logique, la définition d'un concept correspond à un prédicat logique unaire qui détermine l'appartenance d'un individu au concept ; à partir de cette définition, un concept A subsume un concept B si "être un individu décrit par B" entraîne logiquement "être un individu décrit par A", c'est à dire si $\forall x, B(x) \Rightarrow A(x)$ [McGRE88].

La subsomption intensionnelle

La subsomption intensionnelle concerne la structure des concepts : A subsume B si tout individu décrit par B l'est aussi par A ; autrement dit, si l'ensemble de propriétés d'un individu dont la description est définie par B contient l'ensemble des propriétés qui sont spécifiée par A. Tout concept se compose donc des propriétés héritées de ses subsumants, qu'il peut affiner, et des propriétés définies localement. Une description subsume (au niveau intensionnel) une autre description composite pour toute combinaison des raisons suivantes :

1. Une catégorie primaire dans l'une des descriptions est plus générale que dans l'autre :
   [une personne dont les fils sont docteurs] subsume
   [une femme dont les fils sont docteurs]
2. Un modificateur de relation dans l'un des deux est plus générale que dans l'autre :
   [une personne dont les fils sont professionnels] subsume
   [une personne dont les fils sont docteurs]
3. Une condition générale dans l'un des deux est plus générale que dans l'autre :
   [un enfant dont un des parents nettoie sa chambre] subsume
   [un enfant dont la mère nettoie sa chambre]
4. La description la plus spécifique inclut une catégorie, modificateur ou condition qui n'est pas présent dans la description la plus générale :
   [une personne dont les fils sont docteurs] subsume
   [une personne dont les fils sont docteurs et qui aime conduire]

La relation de subsomption est une relation d'ordre (réflexive, antisymétrique et transitive) qui induit une structure taxinomique des concepts. Cette structure a comme élément maximal le concept THING qui subsume tous les autres concepts de la base. Les concepts primitifs concernent les types primitifs du domaine et sont en général près de la racine THING de la base ; les concepts définis, construits à partir d'autres concepts plus simples sont localisés plus bas dans la hiérarchie.

3.3.3 Langage assertionnel

La partie assertionnelle du système utilise des termes du langages de description pour faire des propositions, des assertions sur le monde. Le langage assertionnel permet de décrire tout ce qui peut être déduit sans servir à la classification. Un monde peut être vu selon différents contextes composés de nexus. Un nexus est une entité regroupant toutes les descriptions qui font référence au même objet du monde ; l'existence d'un individu satisfaisant une assertion sur le contexte courant du monde est établie en le connectant au nexus correspondant à l'intérieur du contexte. Ainsi par exemple, si Hichem est le stagiaire du Prof. Aïssani, alors les assertions "Hichem est un étudiant" et "le stagiaire de Prof Aïssani travaillant sur la recherche d'information intelligente sur le Web" référencent le même objet.

3.3.4 Mécanismes de raisonnement

Le mécanisme de raisonnement de base des logiques terminologiques est la classification de concepts, réalisée par un algorithme de classification, appelé le classifieur [SCH83]. Le classifieur prend une nouvelle description de concept et la place à l'endroit correct dans la hiérarchie. Pour trouver la place appropriée pour le nouveau concept, l'algorithme de classification détermine les relations de subsomption entre ce concept et les autres concepts de la hiérarchie ; ces relations peuvent être spécifiées directement, trouvées par transitivité ou bien calculées à partir de la sémantique des conditions des rôles [1] (en prenant la subsomption intensionnelle). La recherche de la place correcte pour le concept comporte trois étapes : la recherche des subsumants les plus spécifiques SPS (concepts qui subsument le concept à classer et dont les sous-concepts ne le subsument pas), la recherche des subsumés les plus généraux SPG (concepts subsumés par le concept à classer et dont les sur-concepts ne sont pas subsumés par lui) et puis l'insertion du concept dans la hiérarchie (figure 3.3).

1. La première étape se fait en profondeur à partir de la racine : tant qu'un concept subsume le concept à classer, ses sous-concepts sont considérés. Le résultat de cette étape est une coupe de la taxinomie au-dessus de laquelle tous les concepts subsument le concept à classer.
2. La deuxième étape considère les sous-graphes des SPS et détermine, parmi les concepts ayant au moins les mêmes propriétés du concept à classer, les subsumés les plus généraux.

[1] Les algorithmes de classification utilisent pour leurs calculs la subsomption intensionnelle.

3. Une fois la position trouvée, la troisième étape de l'algorithme insère la nouvelle description dans la hiérarchie, en l'attachant en dessous des subsumants les plus spécialisés SPS et au dessus des subsumés les plus généraux SPG et en éliminant les liens redondants.

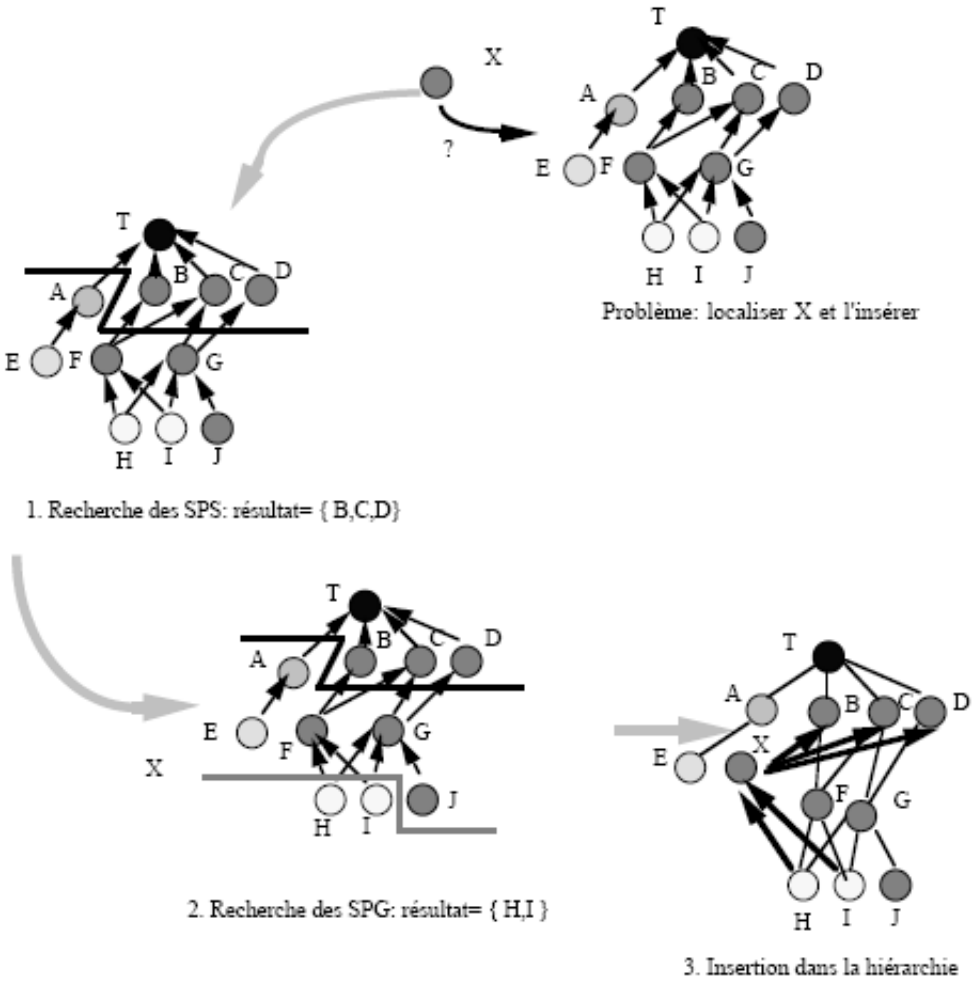

FIG. 3.3 – La classification d'un nouveau concept

Avantages

Par opposition aux réseaux sémantiques dans lesquels les liens et les noeuds sont utilisés avec différentes sémantiques, dans les logiques terminologiques les concepts et les liens ont une sémantique bien définie équivalente à un sous-ensemble de la logique du premier ordre. Un exemple de cette équivalence est le système

OMEGA [ATT86], qui, sans être une logique terminologique, décrit des composants terminologiques à l'aide de la logique classique. Par ailleurs, la taxinomie de concepts permet une organisation adéquate de la connaissance. Enfin, l'algorithme de classification atteint dans certains systèmes comme CLASSIC une grande efficacité.

Inconvénients

Le souci de complétude du raisonnement a entraîné le développement de systèmes ayant une trop faible expressivité et ne pouvant pas représenter des problèmes intéressants. De plus, le langage de représentation des logiques terminologiques introduit des notions qui ne sont pas toujours faciles à comprendre et compliquent le développement de la base. C'est le cas des distinctions entre propriétés définitionnelles et propriétés contingents ou déduites ; entre propriétés nécessaires et propriétés suffisantes et entre subsomption extensionnelle et subsomption intensionnelle. Un aspect particulièrement difficile à comprendre et donc à expliquer, comme l'a souligné Swartout [PAT90], est la distinction entre le raisonnement terminologique et le raisonnement assertionnel ; en effet, une affirmation du style "ce téléphone est rouge" peut être interprétée comme une assertion sur un individu particulier du concept "téléphone", indiquant qu'il est rouge, ou bien comme une assertion affirmant l'existence d'un individu du concept "téléphone rouge". Dans le premier cas, la définition du concept n'inclut pas la couleur ; la propriété couleur ne fait pas partie de la définition ; c'est une propriété contingente. Dans le deuxième cas, seuls les téléphones rouges satisfont le concept ; la couleur fait partie des conditions d'appartenance au concept.

Au niveau du raisonnement, utiliser la classification de concepts pour classer des instances est coûteux et n'est pas cohérent avec la sémantique des instances ; la classification d'instances comme un mécanisme particulier permet une relation d'appariement plus efficace que la subsomption. Ce mécanisme est encore assez peu utilisé, entre autre parce que la plupart des logiques terminologiques n'observent pas l'instance mais le concept.

3.4 Conclusion

Nous avons montré dans ce chapitre l'impact des ontologies dans la construction du Web sémantique, nous avons donc présenté les structures de représentation et avons mis en évidence les problèmes à résoudre avec les ontologies ainsi que leurs domaines d'utilisation.

Nous avons ensuite montré, à travers quelques exemples, que la logique de description se compose de deux langages ; un langage terminologique qui permet

la structuration de la connaissance autour des objets, et un langage assertionnel qui utilise des termes du langages de description pour faire des propositions et des assertions sur le monde.

La combinaison de la logique de description avec les langages d'ontologie Web est représentée dans les langages qui existent déjà tels que OWL, qui sera discuté dans le chapitre suivant.

Chapitre 4
LES ONTOLOGIES WEB BASÉES SUR LA LOGIQUE DE DESCRIPTION

4.1 Introduction

OWL est, tout comme RDF, un langage XML profitant de l'universalité syntaxique de XML. Fondé sur la syntaxe de RDF/XML, OWL offre un moyen d'écrire des ontologies Web. OWL se différencie du couple RDF/RDFS en ceci que, contrairement à RDF, il est justement un langage d'ontologies. Si RDF et RDFS apportent à l'utilisateur la capacité de décrire des classes (ie. avec des constructeurs) et des propriétés, OWL intègre, en plus, des outils de comparaison des propriétés et des classes : identité, équivalence, contraire, cardinalité, symétrie, transitivité, disjonction, etc. Ainsi, OWL offre aux machines une plus grande capacité d'interprétation du contenu web que RDF et RDFS, grâce à un vocabulaire plus large et à une vraie sémantique formelle

4.2 Historique

RDF est le premier langage apparu pour définir la sémantique de sources WEB. Sa structure, que nous étudierons dans les sections suivantes, est issue des réseaux sémantiques [WOO75] introduits dans le domaine de la représentation de connaissances.

Ses constructeurs étant insuffisants pour exprimer la sémantique précise des données, de nouveaux langages ont été conçus suivant une structure nommée "système de frames" [MIN75]. Cette structure vise à organiser la connaissance selon sa dimension structurelle et descriptive. Le premier modèle d'ontologies WEB à l'avoir mis en place est SHOE [HEF99]. Cette structure a été ensuite adoptée par RDF Schema conçu comme extension de RDF.

Deux initiatives ont ensuite été lancées pour étendre le pouvoir d'expression de RDF Schema en utilisant les logiques de description :

- DAML-ONT [CON00] : une initiative américaine lancée par le DARPA ;
- OIL [FHM01] : une initiative sponsorisée par la communauté européenne.

La fusion de ces deux langages a donné naissance à DAML+OIL [CON01] dont la standardisation par le W3C est connu sous le nom de OWL.

4.3 RDF

4.3.1 Influences

RDF a été conçu pour exprimer des assertions sur des ressources WEB. A l'origine, c'était essentiellement pour décrire des pages WEB comme, par exemple, son titre ou son auteur. Puis, le concept de ressource WEB a été étendu à toute information qui peut être identifiée sur le WEB.

4.3.2 Caractéristique d'une ontologie RDF

Un document RDF est composé d'un ensemble de triplets (Sujet Prédicat Object). L'Objet d'un triplet peut être le Sujet d'un autre. Cette caractéristique permet de relier les triplets entre eux. En représentant les Sujets et Objets par des noeuds et les Prédicats par des arrêtes, une ontologie RDF se représente sous la forme d'un graphe. Niveau sémantique, chaque triplet exprime une assertion. En conséquence, la signification d'un tel graphe est l'ensemble de ces assertions.

4.3.3 Constructeurs du modèle pour définir des ressources

4.3.3.1 Identification des ressources

RDF utilise le mécanisme d'URI pour identifier Sujets, Prédicats et Objets et plus précisément de référence par URI :

$$\underbrace{http://www.example.org/index.html}_{\text{URI}}\#\underbrace{section1}_{\text{Reference}}$$

4.3.3.2 Description des ressources

Les ressources définies par une URI sont décrites par les assertions dans lesquelles elles figurent comme Sujet. L'Objet d'une assertion peut aussi être une valeur primitive en utilisant les types de données définis par les XML Schema qui respectent les 3 conditions :

- définir les valeurs acceptées (ex : l'ensemble des dates du calendrier) ;
- définir la syntaxe de ces valeurs (ex : 1999-12-01) ;
- définir la sémantique de ces valeurs (ex : 1999 = Année, 12 = mois, 01 = jours).

Un Objet peut également servir à décrire une structure complexe comme le montre la figure 4.1. Un tel noeud est appelé "noeud blanc".

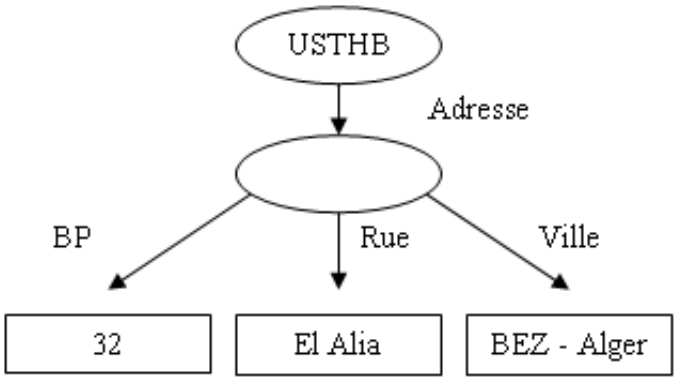

FIG. 4.1 – Représentation du concept Adresse par un nœud blanc

Enfin, RDF fournit des mécanismes pour décrire des assertions, c'est à dire exprimer qu'une ressource WEB est une assertion en indiquant son Sujet, son Prédicat et son Objet. Cette description est appelée une réification. Cette possibilité fait de RDF un langage logique d'ordre 2.

4.4 Sémantique formelle

La sémantique d'un document RDF est définie en s'appuyant sur la théorie du modèle. Celle-ci consiste à définir des structures mathématiques permettant de donner une interprétation du vocabulaire introduit par le langage. Ces structures fournissent le moyen de vérifier formellement qu'une assertion est vraie dans un document RDF. Voici les principales structures qui permettent de définir l'interprétation (I) d'un vocabulaire (V) constitué d'URI et de littéraux (L) :

- IR : ensemble des Ressources (Sujet, Objet) ;
- IP : ensemble des Prédicats ;
- IEXT : application de signature $IP \longrightarrow 2^{IR \times IR1}$ qui associe un Prédicat à ses instances ;
- IS : application de signature $URI \longrightarrow IR \cap IP$ qui interprète une URI comme Prédicat ou Ressource ;
- IL : application de signature $L \longrightarrow IR$ qui interprète un littéral comme une Ressource. I est alors définit par la suite d'équations :

$$I(e) = IS(e) \Longleftrightarrow e \in URI. \tag{4.1}$$

$$I(e) = IL(e) \Longleftrightarrow e \in L. \tag{4.2}$$

$$\begin{aligned} I(e) = TRUE \Longleftrightarrow e = (s,p,o) \wedge s,p,o \in V \wedge I(p) \in IP \\ \wedge (I(s), I(o)) \in IEXT(I(p)). \end{aligned} \tag{4.3}$$

La sémantique du vocabulaire défini par le langage RDF est donnée par des axiomes. Voici quelques axiomes définissant le constructeur ***type*** :

$$type \in IP \tag{4.4}$$

$$p \in IP \Longleftrightarrow (p, predicat) \in IEXT(type) \tag{4.5}$$

4.5 RDF Schema

4.5.1 Une extension de RDF

RDF Schema introduit la connaissance structurelle en définissant la notion de classe, sous-classe et instance de classe. Ces classes sont hiérarchisées par liaison de subsomption. L'héritage multiple est permis. Il introduit également la connaissance

descriptive en définissant les Prédicats comme des propriétés dont le domaine et le codomaine sont des classes.

Ces propriétés présentent deux particularités. D'abord, elles sont, comme les classes, hiérarchisées par des liens de subsomption. Par exemple, la propriété *est-délégué-de* est une *sous-propriété* de *est-membre-de* puisque tout délégué d'une formation est membre de celle-ci. Ensuite, le domaine d'une propriété peut être défini par une liste de classes. Son domaine sera alors composé des instances qui appartiennent à l'ensemble de ces classes.

4.5.2 Extension de la sémantique

Voici les nouvelles structures mathématiques qui permettent l'interprétation du vocabulaire RDF-S :

- IC : ensemble des Classes ;
- ICEXT : application de signature $IC \longrightarrow 2^{IR}$ qui associe une classe à ses instances. Elles sont définies par :

$$x \in ICEXT(C) \Longleftrightarrow (x, C) \in IEXT(type). \tag{4.6}$$

$$IC = ICEXT(Class) \tag{4.7}$$

Les nouveaux axiomes sont les triplets présents dans les trois tableaux présentés en 4.1. Chaque ligne des tableaux est un triplet dont le prédicat est dans l'en-tête. Par exemple, les deuxièmes lignes des deux tableaux les plus à gauche correspondent aux triplets : (domain, domain, property) et (domain, range, Class). Ils indiquent que le domaine de la propriété *domain* est l'ensemble des propriétés et que son codomaine est l'ensemble des classes.

Le nouveau vocabulaire est également défini par des théorèmes mathématiques tels que :

$$(pf, pm) \in IEXT(subPropertyOf) \Rightarrow IEXT(pf) \subset IEXT(pm). \tag{4.8}$$

$$(CF, CM) \in IEXT(subClassOf) \Rightarrow ICEXT(CF) \subset ICEXT(CM). \tag{4.9}$$

Ces équations expriment la sémantique de la subsomption pour les propriétés et classes.

Dans un langage de programmation classique, la création de classes permet de vérifier qu'une instance est conforme à cette définition. Dans un langage de

Domain		Range		Type	
type	resource	type	Class	Resource	Class
domain	property	domain	Class	Class	Class
range	property	range	Class	Literal	Class
subPropertyOf	property	subPropertyOf	Property	XMLLiteral	Class
subClassOf	class	subClassOf	Class	Datatype	Class
subject	Statement	subject	Resource	Seq	Class
predicate	Statement	predicate	Resource	Bag	Class
object	Statement	object	Resource	Alt	Class
member	Resource	member	Resource	Container	Class
first	List	first	Resource	List	Class
rest	List	rest	List	Property	Class
seeAlso	Resource	seeAlso	Resource	Statement	Property
isDefinedBy	Resource	isDefinedBy	Resource	domain	Property
comment	Resource	comment	Literal	range	Property
label	Resource	label	Literal	subPropertyOf	Property
value	Resource	value	Resource	subClassOf	

Tab. 4.1 – Axiomes du vocabulaire RDF Schema

modélisation tel que RDF, les classes permettent également de faire de l'inférence. C'est-à-dire déduire de nouvelles assertions à partir d'assertions connues. Voici quelques règles de déduction :

$$(p, C) \in IEXT(domain) \wedge (u, v) \in IEXT(p) \Rightarrow u \in ICEXT(C). \tag{4.10}$$

$$(p, C) \in IEXT(range) \wedge (u, v) \in IEXT(p) \Rightarrow v \in ICEXT(C). \tag{4.11}$$

$$p \in IP \Rightarrow (p, p) \in IEXT(subPropertyOf). \tag{4.12}$$

$$\begin{aligned}(p1, p2) \in IEXT(subPropertyOf) \wedge (p2, p3) \in IEXT(subPropertyOf) \\ \Rightarrow (p1, p3) \in IEXT(subPropertyOf).\end{aligned} \tag{4.13}$$

$$C \in IC \Rightarrow (C, C) \in IEXT(subClassOf). \tag{4.14}$$

$$\begin{aligned}(C1, C2) \in IEXT(subClassOf) \wedge (C2, C3) \in IEXT(subClassOf) \\ \Rightarrow (C1, C3) \in IEXT(subClassOf).\end{aligned} \tag{4.15}$$

Les équations 4.10 et 4.11 permettent de déduire l'appartenance d'une ressource à une classe lorsqu'elle est utilisée dans une propriété comme membre de son domaine ou codomaine. Les équations 4.12 et 4.13 exploitent le fait que le constructeur *subPropertyOf* est réflexif et transitif. Les équations 4.14 et 4.15 font de même avec *subClassOf*.

4.5.3 Limites de RDF Schema

RDF Schema fournit des mécanismes de bases tels que l'identification des concepts d'un domaine. Il ne permet cependant pas de :

- définir des contraintes de cardinalité sur les propriétés ;
- qualifier les propriétés (transitives, uniques ...) ;
- construire des classes à partir d'autres ;
- identifier les ressources identiques mais ne portant pas la même URI ;
- regrouper les concepts pour pouvoir les importer ou référencer ailleurs ;
- gérer la version des concepts.

C'est pour cela qu'une nouvelle analyse des besoins [HEF04] a été faite pour la spécification d'un langage de définition d'ontologies Web. Elle a donné naissance au langage OWL.

4.6 OWL

OWL est décomposé en trois sous-langages : Lite, DL et Full. Il est défini par une syntaxe abstraite et une syntaxe concrète. De plus, deux sémantiques formelles basées sur la théorie du modèle lui sont associées. Cette multiplicité s'explique par les besoins et influences qui ont motivé sa conception [HP03a].

4.6.1 Les influences qui ont guidé la conception de OWL

La principale source d'influences de OWL est issue de ses prédécesseurs. Parmi les impératifs de conception de ce langage figure la compatibilité avec RDF. Celle-ci se traduit par l'utilisation de RDF/XML comme syntaxe concrète de OWL. Seulement, cette compatibilité ne doit pas être uniquement syntaxique mais aussi sémantique. Ce problème a été résolu en s'inspirant des solutions trouvées par d'autres prédécesseurs de OWL tels que SHOE [HHS99], OIL [FEN01] et DAML+OIL [CHM01].

Le domaine de la logique de description (Description Logics) est la seconde source d'influences de OWL. Elle a, d'une part, influencé le choix de spécifier la sémantique du langage par la théorie du modèle. D'autre part, les études sur la complexité menées dans ce domaine ont également orienté les choix des constructions du langage. En effet, un autre objectif important de OWL est de pouvoir assurer qu'il est possible de créer un moteur d'inférences décidable basé sur ce langage. Ceci signifie qu'il existe un algorithme pour évaluer la subsomption entre deux ontologies.

Enfin, la troisième source d'influences est le domaine des Frames (Frames paradigm) qui consiste à regrouper l'ensemble des informations qui se rattachent à un même élément. Ceci a influencé la conception de la syntaxe abstraite du langage présentée dans la section 4.6.5.

4.6.2 Caractéristiques d'une ontologie OWL

OWL n'est pas dédié à la conception d'ontologies d'une des deux catégories présentées dans la classification donnée en [ZAI04]. Il permet de construire des ontologies qui peuvent être linguistiques, conceptuelles ou intermédiaire entre ces

deux catégories. Cette particularité vient du fait que OWL permet de définir à la fois des concepts primitifs et définis.

Quelque soit sa nature, une ontologie OWL possède les caractéristiques suivantes :

- multilingue ; OWL utilise les mêmes mécanismes que RDF Schema ;
- modulaire ; une ontologie peut importer l'ensemble des concepts qui sont définis dans une autre ontologie ;
- compatible avec RDF ; les outils créés pour RDF peuvent traiter une ontologie OWL ; ils ne permettent bien entendu pas d'en exploiter la sémantique ;
- interopérable ; les ontologies peuvent être reliées en statuant que deux concepts définis dans deux ontologies sont identiques ;
- déductive ; l'application des règles de déduction permet d'ajouter de nouvelles caractéristiques aux instances liées à l'ontologie ; la cohérence et la non redondance d'une ontologie peuvent également être vérifiées par les mêmes mécanismes.

4.6.3 Les trois versions de OWL

L'impossibilité d'obtenir un langage à la fois compatible avec RDF et décidable a poussé les concepteurs de OWL à en spécifier trois versions [DS04, HP03a, BCH03, BCH04].

4.6.3.1 OWL DL

Une première nommée DL, pour Description Logics, se définit par la syntaxe abstraite. Seuls les graphes RDF résultant de la conversion entre cette syntaxe et des triplets RDF sont acceptés dans cette version. De tels graphes sont qualifiés de bien formés. Les constructeurs définis par la syntaxe abstraite ont été choisis de manière à ce que cette version de OWL reste décidable. Le problème d'inférences en OWL DL a ainsi pu être classé comme étant aussi difficile que celui de l'inférence dans les langages de la famille SHOIN(D) [HP04]. Des travaux de recherche ont montré que ces langages sont de complexité au pire des cas de temps exponentiel non déterministe (NExpTime) [TOB01]. En pratique, il n'existe pas d'algorithme connu pour implémenter un moteur d'inférences sur de tels langages satisfaisant aux exigences usuelles en terme de qualité et temps de réponse.

4.6.3.2 OWL Lite

OWL Lite est le sous langage de OWL le plus simple. Il est destiné aux utilisateurs qui ont besoin d'une hiérarchie de concepts simple.

Ce langage résulte de la conversion d'une syntaxe abstraite simplifiée de OWL vers des triplets RDF. Cette version a la même complexité que les langages de la famille SHIF(D) : temps exponentiel déterministe *(ExpTime)*. En pratique, deux implémentations de moteur d'inférences sur de tels langages ont été utilisées avec succès dans des applications en production et sur de larges ontologies. Il s'agit de FaCT [Hor98] et RACER [HAA01].

4.6.3.3 OWL Full

La dernière version de OWL se nomme OWL Full. Elle se caractérise par une compatibilité totale avec RDF et RDF Schema. Cette version ne permet pas, tout comme RDF, la distinction entre instances et classes. Elle autorise également l'utilisation des constructeurs OWL comme paramètres de ses constructeurs. Il est par exemple possible d'indiquer qu'une classe donnée ne doit pas avoir plus de 2 super-classes. Ce pouvoir d'expression fait que OWL Full est indécidable. Dans la suite de ce chapitre, sauf indication contraire, nous présenterons la version DL de OWL.

4.6.4 Constructeurs du modèle pour définir des concepts

4.6.4.1 Identification d'un concept

OWL, comme RDF, utilise les URI pour l'identification des concepts.

4.6.4.2 Définition d'un concept : connaissance structurelle

Une première méthode pour construire une classe OWL est d'en énumérer les instances en utilisant le mot clé *oneOf*. La seconde méthode est d'utiliser une restriction OWL qui définit une classe anonyme en spécifiant une contrainte associée à une propriété. Cette classe est l'ensemble des instances satisfaisant cette contrainte. Celle-ci peut être de trois sortes :

- une contrainte de cardinalité ; elle définit le nombre minimum, exact ou maximum de valeurs que doivent définir les instances pour la propriété contrainte ;
- une contrainte de codomaine ; *allValuesFrom* (resp. *someValuesFrom*) permet de créer une classe dont les instances ne peuvent prendre pour valeur de la propriété contrainte que des (resp. au moins une) instances d'une classe spécifiée ;
- une contrainte de valeur ; *hasValue* permet de créer une classe dont les instances ont une valeur spécifiée pour la propriété contrainte.

La dernière méthode pour créer une classe OWL est d'appliquer des opérations booléenne à une ou plusieurs classes déjà définies (*intersectionOf , unionOf , complementOf*).

En plus de *subClassOf* définit par RDF Schema, OWL fournit deux autres constructeurs pour lier des classes : *equivalentClass* et *disjointClass*. Ils permettent d'indiquer que les deux classes ont les mêmes ensembles d'instances ou que ceux-ci sont disjoints.

4.6.4.3 Définition d'un concept : connaissance descriptive

OWL distingue deux catégories de propriétés en fonction de leur codomaine : *DatatypeProperty* et *ObjectProperty*. Le codomaine d'une propriété peut ainsi être un type de données ou une classe. Ces propriétés peuvent être hiérarchisées comme en RDF Schema par le constructeur *subPropertyOf*.

Le modèle OWL fournit également un moyen de les qualifier. Ainsi, une propriété pourra être une :

- *TransitiveProperty* ou *SymetricProperty* ce qui correspond aux notions mathématiques de symétrie et transitivité ;
- *FunctionalProperty* c'est-à-dire une fonction mathématiques ;
- *InverseFunctionalProperty* : une application injective.

4.6.5 La syntaxe de OWL

Pour la compatibilité avec RDF, la syntaxe RDF/XML a été choisie pour OWL.

Constructeurs OWL	Interprétation
$Class(ccompletedesc_1...desc_n)$	$EC(c) = EC(desc_1) \cap ... \cap EC(desc_n)$
$Class(cpartialdesc_1...desc_n)$	$EC(c) \subset EC(desc_1) \cap ... \cap EC(desc_n)$
$oneOf(i_1...i_n)$	$i_1...i_n$
restriction(p allValuesFrom(r))	$\{x \in O \mid (x,y) \in ER(p) \Rightarrow y \in EC(r)\}$
restriction(p someValuesFrom(e))	$\{x \in O \mid \exists (x,y) \in ER(p) \wedge y \in EC(e)\}$
restriction(p value(i))	$\{x \in O \mid (x,i) \in ER(p)\}$
restriction(p minCardinality(n))	$\{x \in O \mid card(\{y \mid (x,y) \in ER(p)\}) \geq n\}$
restriction(pmaxCardinality(n))	$\{x \in O \mid card(\{y \mid (x,y) \in ER(p)\}) \leq n\}$
restriction(p cardinality(n))	$\{x \in O \mid card(\{y \mid (x,y) \in ER(p)\}) = n\}$
complementOf(c)	$O - EC(c)$
$unionOf(c_1...c_n)$	$EC(c_1) \cup ... \cup EC(c_n)$
$intersectionOf(c_1...c_n)$	$EC(c_1) \cap ... \cap EC(c_n)$
$DisjointClasses(d_1...d_n)$	$EC(d_i) \cap EC(d_j) = \{\} 1 \leq i < j \leq n$
$EquivalentClasses(d_1...d_n)$	$EC(d_i) = EC(d_j) 1 \leq i < j \leq n$
$SubClassOf(d_1 d_2)$	$EC(d_1) \cup EC(d_2)$

TAB. 4.2 – Sémantique des constructeurs de classes OWL

EXEMPLE

Pour exprimer qu'un encadreur ne peut encadrer qu'un seul étudiant, les représentaions en LD et OWL sont les suivantes :

$$Encadreur = Jury \cap \leq\ 1\ est\text{-}encadreur\text{-}de$$

```
<owl :Class rdf :ID="Encadreur">
  <rdfs :subClassOf>
    <owl :Restriction>
      <owl :onProperty>
        <owl :DatatypeProperty rdf :ID="est-encadreur-de"/>
      </owl :onProperty>
      <owl :maxCardinality rdf :datatype=http ://www.w3.org/2001/
        "XMLSchema#int">1
      </owl :maxCardinality>
    </owl :Restriction>
  </rdfs :subClassOf>
  <rdfs :subClassOf>
    <owl :Class rdf :about="#Jury"/>
  </rdfs :subClassOf>
</owl :Class>
```

Pour que ces triplets RDF puissent être convertis en syntaxe RDF/XML, il faut que tous les termes présents dans ces triplets soient définis dans cette syntaxe. C'est pour cela que l'ensemble des constructeurs de OWL est défini en RDF Schema [BCH04]. Voici la définition des termes OWL utilisés pour définir une restriction :

```
<rdfs :Class rdf :ID="Restriction">
  < rdfs :label> Restriction < /rdfs :label>
  <rdfs :subClassOf rdf :resource="#Class″/ >
< /rdfs :Class>
<rdf :Property rdf :ID="onProperty">
  < rdfs :label> onProperty < /rdfs :label>
  <rdfs :domain rdf :resource="#Restriction"/ >
  <rdfs :range rdf :resource="& rdf ;Property"/ >
< /rdf :Property>
<rdf :Property rdf :ID="minCardinality">
  <rdfs :label> minCardinality < /rdfs :label>
  <rdfs :domain rdf :resource="#Restriction"/ >
  <rdfs :range rdf :resource="& xsd ;nonNegativeInteger"/ >
< /rdf :Property>
```

Remarque

Comme toute ontologie OWL doit faire référence à ces définitions, un espace de noms vers ce vocabulaire est souvent défini dans l'en-tête d'une ontologie OWL exprimée en RDF/XML.

4.7 Conclusion

Dans ce chapitre, nous avons discuté d'une façon détaillée la représentation dans RDF et RDF schema avec quelques exemples de définitions de concepts dans OWL. Ceci, tout en mettant en évidence les caractéristiques et la syntaxe d'une ontologie OWL.

Malgré la diversité de représentation que OWL nous offre, nous pensons que la syntaxe OWL reste encore loin à répondre à tous les besoins de représentations possibles dans les LDs. Par exemple, pour dire qu'une propriété est exceptionnelle pour un concept donné il n'est pas évident de trouver une représentation immédiate et nous force à rechercher des solutions en utilisant les restrictions, ce qui n'est pas toujours évident.

Dans ce contexte, nous étudierons dans le chapitre suivant les différents types de représentation nous permettant d'avoir une définition complète de concept et comment on peut les représenter dans OWL.

Chapitre 5
EXTENSION OWL AVEC LES DÉFINITION PAR DÉFAUT ET EXCEPTION

5.1 Introduction

Le nouveau langage des ontologies web (OWL) et son sous langage basé sur la description logique (OWL-DL) excluent, explicitement, les définitions par défauts et les exceptions, comme c'est le cas pour tous les formalismes basés sur la logique. Toutefois, comme peu de concepts sont définissables en utilisant seulement la reconnaissance stricte (et non pas par défaut), l'imposante des définitions strictes induit des bases de connaissances terminologiques dont la majorité de ses concepts sont partiellement définis.

Notre contribution s'appuie sur l'idée de discuter les solutions proposées dans ce contexte pour les LDs, et les appliquer sur les ontologies OWL-DL. Dans la section 5.2, nous présentons l'intérêt des définition par défaut et exception. Dans la section 5.3, nous étudierons l'extension du langage de description standard pour les LDs. Dans la section 5.4, nous proposons l'extension du langage OWL avec étude de la subsomption, la complexité et la résolution des conflits.

5.2 L'intérêt des définitions *par défaut* et *exception*

5.2.1 Le raisonnement par défaut

Nous essayons, dans cette section, d'introduire l'aspect du défaut. Nous commençons par donner un rappel sur la subsomption et la représentation en OWL. Ensuite, nous présentons les différents types de concepts. Finalement, Nous discutons le besoin de l'aspect du défaut dans la définition des concepts.

Subsomption

Les concepts sont ordonnés par des relations de subsomption : Un concept A est subsumé par un concept B ssi A est plus général que B.

$B \subset A$ (i.e. $B \cap A = B$)

EXEMPLE

$Enseignant \subset Employe$

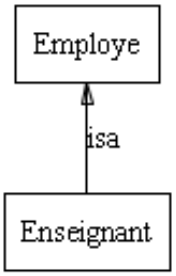

FIG. 5.1 – Exemple de subsomption

La représentation en OWL est :

<rdfs :Class rdf :ID="Enseignant">
<rdfs :label>Enseignant</rdfs :label>
<rdfs :subClassOf rdf :resource="#Employe"/>
</rdfs :Class>

Types de concepts

On définit deux types de concept selon la façon de représentation choisie :

1. **Primitifs**

   Les concepts primitifs se caractérisent par des définitions *incomplètes*. Ce type de concept est représenté par des Conditions Nécessaires mais non suffisantes (CN).

   EXEMPLE

– une *personne* a une *date de naissance*.
– un animal a une *date de naissance* mais n'est *pas une personne*.

Dans cet exemple, une date de naissance est une condition nécessaire pour chaque personne mais n'est pas suffisante pour dire que tout être qui a une date de naissance est une personne, comme c'est le cas des animaux par exemple.

2. **Définis**

Les concepts définis se caractérisent par des définitions *complètes*. Ce type de concept est représenté par des Conditions Nécessaires et Suffisantes (CNS).

Exemple

– un père a des enfants.

Dans cet exemple, un père a forcément des enfants. D'un autre côté, tout homme qui a des enfants est un père.

Discussion

Afin d'arriver à une définition complète de tous les concepts, on propose de compléter les définitions en utilisant les connaissances par défaut. Cette complétude nous aide aussi dans la classification des termes, cette dernière ne peut opérer que sur des concepts bien définis. Dans l'exemple suivant, on montre comment de compléter la définition des concepts personne et animal par des définitions plus détaillées :

– une personne a par défaut deux pieds
– un animal a par défaut quatre pattes

De ce qui précède, on déduit que la connaissance peut être stricte (basée sur des concepts primitifs) ou par défaut (basée sur des concepts définis). En utilisant la connaissance par défaut dans l'exemple ci-dessus, il est impossible de faire une confusion entre une personne et animal puisque il y'aura pas une contradiction entre le fait d'avoir deux pieds et d'avoir quatre pattes.

5.2.2 Le raisonnement d'*exception*

5.2.2.1 Définition d'exception

Une exception est une expression d'un cas particulier (i.e. exceptionnel) ou spécialisation pour une propriété d'un concept. Cette propriété peut être une exception pour un concept et au même temps une simple propriété pour un autre concept. Dans un exemple plus bas, on définit un Éléphant par sa couleur grise. Un Éléphant Royal est un Éléphant connu par défaut avec une couleur Blanche et qui peut exceptionnellement avoir la couleur Grise. Ici, la propriété Gris sera une exception pour l'Éléphant Royal. Le concept Éléphant Royal est donc défini avec plus de précision.

Il est à noter que ce qui s'applique sur les définitions par défaut s'applique aussi bien sur les exceptions. Les exceptions nous permettent non seulement de compléter la définition des concepts, mais aussi de générer des spécialisations et d'accepter quelques cas exceptions pour quelques concepts, ce qui élargie encore d'avantage le domaine de définition du concept avec plus de complétude. Dans la prochaine section, nous illustrons les notions de défaut et exception avec plus de détails en utilisant le langage $AL_{\delta\epsilon}$ [COU97].

5.2.3 Motivation

En logiques de description, la connaissance par défaut n'est pas traitée dans la définition des concepts. De plus, peu de concepts sont définissables en utilisant seulement la connaissance stricte.

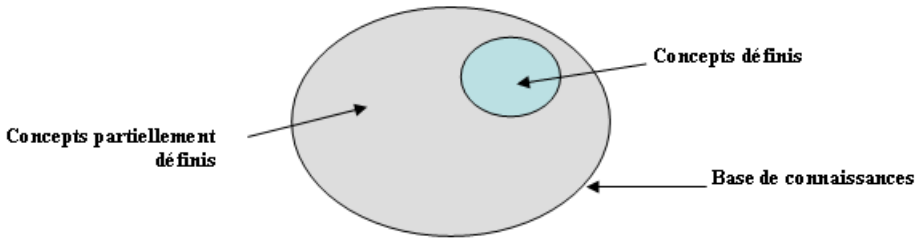

FIG. 5.2 – Représentation des concepts dans la base de connaissance

A partir de la figure 5.2, on peut voir que l'utilisation des définitions strictes induit des bases de connaissance terminologiques dont la majorité de ses concepts

sont partiellement définis. De là, et comme la classification ne peut opérer que sur des concepts bien définis, la classification ne peut donc opérer sur la majorité des concepts.

Des recherches ont été menées en LDs afin d'étendre le langage avec les définitions par défaut [COU97], ces travaux consistaient à étendre le langage de description standard *ALN*. Toutefois, le raisonnement par défaut reste, explicitement, exclut dans OWL et son sous langage OWL-DL.

Dans le reste de notre travail, on se basera sur les travaux menés en LD et nous essayons de les adapter sur les ontologies OWL. Nous montrerons comment il est possible d'étendre le langage OWL et nous discutons de l'impact de la subsomption sur la définition multiple (par défaut et exception) de concepts et le traitement de types de conflits.

5.3 Extension du langage *ALN*

5.3.1 Le langage $AL_{\delta\epsilon}$

Dans ce qui suit nous présentons le langage $AL_{\delta\epsilon}$ [COU97], un langage basé sur le langage *ALN* (section 3.3.1). La nouvelle grammaire adoptée dans ce langage nous permet d'exprimer des concepts complétés avec les définitions par défaut et exception.

Dans [COU97] le langage $AL_{\delta\epsilon}$ est défini comme suit :

C,D $\rightarrow$	$\bot$	concept le plus général
	$\|P$	concept primitif
	$\|\neg P$	négation du concept primitif
	$\|C \sqcap D$	conjonction des concepts
	$\|\forall\exists R : C$	C est une restriction de valeur pour tous les rôles de $R(> 0)$
	$\|\delta C$	concept par défaut
	$\|C^{\epsilon}$	Exception du concept C

R = ensemble des rôles primitifs

P = ensemble des concepts primitifs
$\forall\exists$: remplace, At_Least, At_Most

EXEMPLE

$$Arbre \equiv \delta Avec\text{-}branches \cap Avec\text{-}tronc \cap Avec\text{-}racinces$$

$$Scion \equiv \delta Age\text{-}un\text{-}an \cap Arbre \cap Avec\text{-}branches^{\epsilon}$$

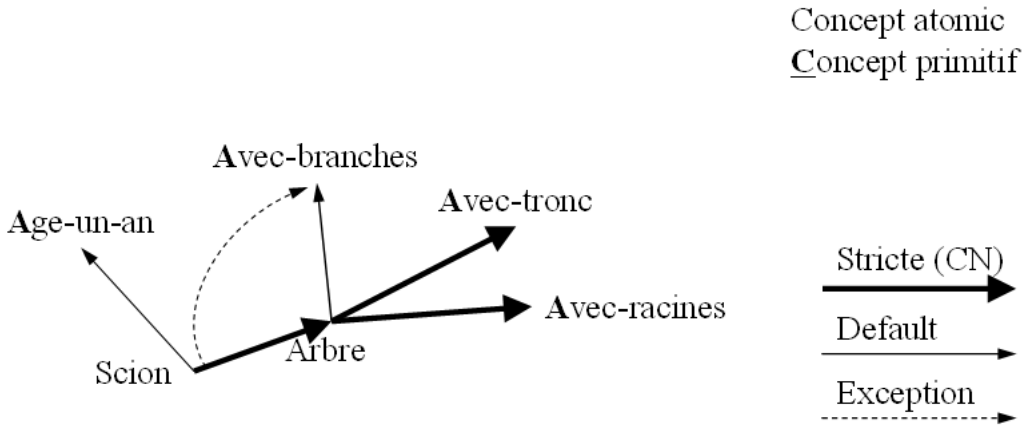

FIG. 5.3 – Exemple de définition par défaut et exception

Dans cet exemple, un arbre a un tronc, des racines et par défaut des branches. Un scion est un arbre qui a par défaut un an d'age et qui a exceptionnellement des branches (figure 5.3). Le concept *Scion* hérite donc du concept *Arbre* toutes les caractéristiques avec exception sur la propriété *Avec-branches*. C'est une sorte de redéfinition du concept *Arbre* utilisé au sein du concept *Scion*.

5.3.2 Système équationnel pour $AL_{\delta\epsilon}$

Le système équationnel met en évidence les principales propriétés des connecteurs utilisés et donne une relation d'équivalence entre les termes conceptuels. Nous considérons ici l'ensemble des équations pour le langage $AL_{\delta\epsilon}$, où A, B et C appartiennent à $AL_{\delta\epsilon}$.

$\sqcap$:	associativité	$(A \sqcap B) \sqcap C = A \sqcap (B \sqcap C)$
	commutativité	$A \sqcap B = A \sqcap B$
	idempotence	$A \sqcap A = A$
	élément neutre	$\top \sqcap A = A$
ϵ	$(\epsilon 1)$	$(\delta A)^{\epsilon} = A^{\epsilon}$
δ	distributivité $(\delta 1)$	$\delta(A \sqcap B) = (\delta A) \sqcap (\delta B)$
	$(\delta 2)$	$A \sqcap \delta A = A \quad (A \subset \delta A)$
	$(\delta 3)$	$A^{\epsilon} \sqcap \delta A = A^{\epsilon} \quad (A^{\epsilon} \subset \delta A)$
	idempotence$(\delta 4)$	$\delta\delta A = \delta A \quad (A \subset \delta A)$

5.3.3 Définitions

Nous donnons ci-dessous les principales définitions, vous pouvez trouver les détails et les preuves de définitions dans [COU97]. Nous les donnons ici juste à titre d'utilisation et substitution dans les formules de subsomption.

Déf1. $A \sqcap \neg A = \bot$

Déf2. $A^{\epsilon^{\epsilon}} = \delta A$

Déf3. $A \sqcap A^{\epsilon^{\epsilon}} = A^{\epsilon^{\epsilon}} \quad (A^{\epsilon^{\epsilon}} \subset A)$

Déf4. $A \sqcap A^{\epsilon} = \bot$

Déf5. $(A \sqcap B)^{\epsilon} = A^{\epsilon} \cup B^{\epsilon}$

Déf6. $\delta A = \bigcup_{i \geq 0} A^{\epsilon^i}$

REMARQUE. $A^{\epsilon^{\epsilon}}$ est utilisée pour exprimer une exception d'une exception. Ce type de co-exception peut se présenter lors d'héritage des concepts (subsomption).

5.3.4 Subsomption dans $AL_{\delta\epsilon}$

Dans cette section nous allons voir comment on remplace les définitions par défaut par les exceptions dans le cas d'un héritage.

EXEMPLE

Prenant l'exemple du concept Elephant-Royal (ER) (figure 5.4) qui hérite de Elephant (E). Gris est une propriété par défaut de Elephant, comme ER hérite de E alors Gris devient une propriété par défaut de ER. D'un autre côté, ER a une exception pour la propriété Gris, nous avons donc la propriété "Gris" définie comme propriété par défaut dans l'héritage et comme exception dans la définition du concept même. L'idée ici est de remplacer la définition par défaut par l'exception. ER devient donc un cas (case-of) de Elephant.

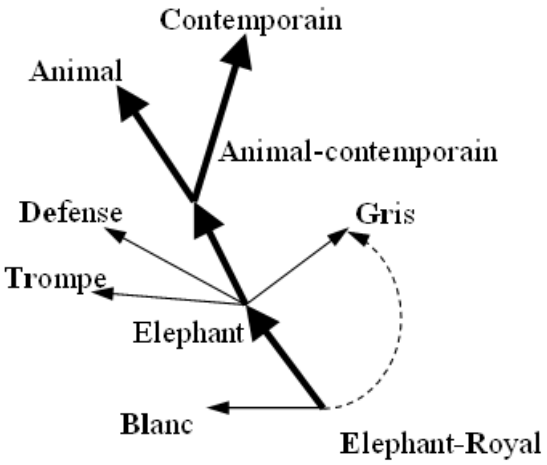

FIG. 5.4 – Exemple de subsomption dans $AL_{\delta\epsilon}$

Ce graphe s'exprime dans le langage $AL_{\delta\epsilon}$ comme suit :

Définitions

$Ac \equiv A \sqcap C$

$E \equiv \delta Gr \sqcap \delta De \sqcap \delta Tr \sqcap Ac$

$ER \equiv E \sqcap \delta Bl \sqcap Gr^{\epsilon}$

$ER \equiv \delta Gr \sqcap \delta De \sqcap \delta Tr \sqcap \delta Bl \sqcap Gr^{\epsilon}$

En remplaçant la propriété δGr par Gr^{ϵ}, on obtient :

$ER \equiv \delta De \sqcap \delta Tr \sqcap \delta Bl \sqcap Gr^{\epsilon}$

5.4 Le langage OWL$_{\delta\epsilon}$

Dans cette section nous proposons une extension du langage OWL en ajoutant de nouveaux constructeurs au langage. Ces nouveaux constructeurs vont nous permettre une plus grande flexibilité de représentation des concepts grâce à l'introduction des nouveaux aspects de défaut et exception précédemment discutés.

Le constructeur "default" en RDF

<rdfs :Class rdf :ID="default">
<rdfs :label>default</rdfs :label>
<rdfs :subClassOf rdf :resource="#Class"/ >
</rdfs :Class>

Le constructeur "exception" en RDF

<rdfs :Class rdf :ID="exception">
<rdfs :label>exception</rdfs :label>
<rdfs :subClassOf rdf :resource="#Class"/ >
</rdfs :Class>

5.4.1 Comparaison entre la représentation $AL_{\delta\epsilon}$ et OWL$_{\delta\epsilon}$

On reprend ici les définitions en AL$_{\delta\epsilon}$ des deux concepts *Arbre* et *Scion* discutés dans l'exemple de la figure 5.3 et nous donnons leur représentation correspondante en OWL$_{\delta\epsilon}$. Soit la représentation en AL$_{\delta\epsilon}$ suivante :

$Arbre \equiv \delta Avec\text{-}branches \cap Avec\text{-}tronc \cap Avec\text{-}racinces$

Représentation en OWL$_{\delta\epsilon}$

<owl :Class rdf :ID='Arbre'>
<owl :intersectionOf rdf :parsetype='Collection'>
<owl :default>
<owl :Class rdfs :about='Avec-branches'>
</owl :default>
<owl :Class rdfs :about='Avec-tronc'>
<owl :Class rdfs :about='Avec-racines >
</owl :intersectionOf>

</owl :Class>

Représentation en $AL_{\delta\epsilon}$

$Scion \equiv \delta Age\text{-}un\text{-}an \cap Arbre \cap Avec\text{-}branches^{\epsilon}$

Représentation en $OWL_{\delta\epsilon}$

```
<owl :Class rdf :ID='Scion'>
  <owl :intersectionOf rdf :parsetype='Collection'>
     <owl :default>
        <owl :Class rdfs :about='Age-un-an'>
     </owl :default>
     <owl :Class rdfs :about='Arbre'>
     <owl :exception>
        <owl :Class rdfs :about='Avec-branches'>
     </owl :exception>
  </owl :intersectionOf>
</owl :Class>
```

5.4.2 Subsomption dans $OWL_{\delta\epsilon}$

Comme pour $AL_{\delta\epsilon}$, lors de la subsomption, des concepts par défaut hérités doivent être substitués par les concepts définis en exception dans la classe courante.

Exemple

Dans l'exemple précédent, si on remplace la définition du concept *Arbre* dans la définition du concept *Scion*, on obtient la définition suivante :

$Scion \equiv \delta Age\text{-}un\text{-}an \cap \delta Avec\text{-}branches \cap Avec\text{-}tronc \cap Avec\text{-}racinces \cap Avec\text{-}branches^{\epsilon}$

Ici, le concept *Avec-branches* est utilisé par défaut et comme une exception en même temps. La représentation en $OWL_{\delta\epsilon}$ est la suivante :

```
<owl :Class rdf :ID='Scion'>
  <owl :intersectionOf rdf :parsetype='Collection'>
     <owl :default>
        <owl :Class rdfs :about='Age-un-an'>
     </owl :default>
     <owl :intersectionOf rdf :parsetype='Collection'> // début du Arbre
           <owl :default>
              <owl :Class rdfs :about='Avec-branches'>
           </owl :default>
           <owl :Class rdfs :about='Avec-tronc'>
           <owl :Class rdfs :about='Avec-racines >
     </owl :intersectionOf>                          // fin du concept Arbre
     <owl :exception>
        <owl :Class rdfs :about='Avec-branches'>
     </owl :exception>
  </owl :intersectionOf>
</owl :Class>
```

On remarque qu'on a obtenu deux intersections imbriquées une dans l'autre. Il faut d'abord réduire les intersections avant de procéder à la substitution du concept *Avec-branches*. Deux étapes sont alors nécessaires et seront discutées en détail dans les sections suivantes :

- Réduction des opérateurs
- Remplacer les définitions par défaut par les définitions d'exception

5.4.3 Système équationnel pour $OWL_{\delta\epsilon}$

Le système équationnel pour $OWL_{\delta\epsilon}$ nous permet de mettre en évidence les principales propriétés des connecteurs et de réduire au maximum possible le nombre de propriétés utilisées. Nous considérons un ensemble d'équations pour le langage $OWL_{\delta\epsilon}$, où c appartient à $OWL_{\delta\epsilon}$:

• IntersectionOf

Associativité

$intersectionOf(c_1, c_2, c_3) = intersectionOf(intersectionOf(c_1, c_2), c_3)$

$intersectionOf(intersectionOf(c_1, c_2), c_3) = intersectionOf(c_1, intersectionOf(c_1, c_2))$

Commutativité

$intersectionOf(c_1, c_2) = intersectionOf(c_1, intersectionOf(c_1, c_2))$

Idempotence

$intersectionOf(c_1, c_2) = (intersectionOf(c_2, c_1))$

• Exception

$exception(default(c)) = exception(c)$

• Default

Distributivité (default1)

$default(intersectionOf(c_1, c_2)) = intersectionOf(default(c_1), default(c_2))$

default2

$intersectionOf(c, default(c)) = c$

default3

$intersectionOf(exception(c), default(c)) = exception(c)$

Idempotence (default4)

$default(default(c)) = default(c)$

Exemple

En appliquant la première règle d'associativité de *intersectionOf* dans l'exemple précédent, on obtient l'écriture suivante :

```
<owl :Class rdf :ID='Scion'>
  <owl :intersectionOf rdf :parsetype='Collection'>
     <owl :default>
        <owl :Class rdfs :about='Age-un-an'>
     </owl :default>
     <owl :default>
        <owl :Class rdfs :about='Avec-branches'>
     </owl :default>
     <owl :Class rdfs :about='Avec-tronc'>
     <owl :Class rdfs :about='Avec-racines >
     <owl :exception>
        <owl :Class rdfs :about='Avec-branches'>
     </owl :exception>
  </owl :intersectionOf>
</owl :Class>
```

Pour remplacer la définition par défaut du concept *Avec-branches* par l'exception, on utilisera la règle *default3*. On obtient donc :

```
<owl :Class rdf :ID='Scion'>
  <owl :intersectionOf rdf :parsetype='Collection'>
     <owl :default>
        <owl :Class rdfs :about='Age-un-an'>
     </owl :default>
     <owl :Class rdfs :about='Avec-tronc'>
     <owl :Class rdfs :about='Avec-racines >
     <owl :exception>
        <owl :Class rdfs :about='Avec-branches'>
     </owl :exception>
  </owl :intersectionOf>
</owl :Class>
```

Dans la section suivante, nous donnons la fonction g qui permet la substitution des concepts par défaut par leur exception.

5.4.4 L'algorithme de substitution dans $OWL_{\delta\epsilon}$

Le principe de la fonction g est d'avoir en entrée un concept c et deux ensembles d_δ et d_ϵ. L'ensemble d_δ contient la liste des définitions par défaut utilisées dans le concept c, tandis que l'ensemble d_ϵ contient la liste des exceptions définis dans le concept c. Pour chaque concept a de l'ensemble d_ϵ, on cherche s'il existe un concept b de l'ensemble d_δ tel que $a = b$.

$g : D \times 2^D \times 2^D \rightarrow D$ **such that**
$g(c, d_\epsilon, d_\delta) =$
if $d_\epsilon = \emptyset$
then return c
else $res \leftarrow c$
for all $a \in d_\epsilon$
if there exists $b \in d_\delta$ **such that** $a.rdf : about = b.rdf : about$
then $d_\delta \leftarrow d_\delta \setminus b$
$res \leftarrow g(res, d_\epsilon \setminus a, d_\delta)$
endfor

5.4.5 Complexité de calcul dans $OWL_{\delta\epsilon}$

Dans cette section, nous étudierons la complexité de calcul dans l'algorithme de substitution pour la fonction g et nous montrerons que la complexité est polynômiale.

Proposition 1. La fonction g est d'une complexité polynômiale.

Preuve. Soit n et m les longueurs des ensembles d_ϵ et d_δ respectivement. Nous savons que la fonction g est une fonction récursive et que la recherche se fait sur une taille décroissante des ensembles, i.e. à chaque substitution, l'ensemble d_δ sera décrémenté (i.e. $m-1$) et la recherche se poursuivra sur le reste de l'ensemble d_ϵ sans le concept remplacé (i.e. $n-1$).

NOTATION. Notons $\overline{g}$ la complexité de calcul de la fonction g, et $\overline{g}_i$ la complexité de calcul de la fonction g à l'étape i de substitution.

$$
\begin{aligned}
&\overline{g}_1 \leq n \times m \\
&\overline{g}_2 \leq (n-1) \times (m-1) \\
&\vdots \\
&\overline{g}_n \leq (n-(n-1)) \times (m-(n-1)) \quad (avec\ m > n-1) \\
&\quad \leq (1) \times (m-n+1) \\
&\quad \leq m-n+1
\end{aligned}
$$

Nous savons que :

$$\overline{g} = \overline{g}_1 + \overline{g}_2 + \ldots + \overline{g}_n$$

ce qui veut dire que :

$$
\begin{aligned}
\overline{g} \ &\leq (n \times m) + ((n-1) \times (m-1)) + \ldots + (m-n+1) \\
&\quad \leq (n \times m) \times n \qquad (n\ fois) \\
&\quad \leq mn^2
\end{aligned}
$$

5.4.6 Résolution des conflits

Un des grands problèmes auxquels on est confronté dans le processus d'héritage est le cas de conflit de définition qui peut se présenter lors de l'héritage des concepts. Pour illustrer ce problème, nous proposons d'étudier l'exemple suivant.

EXEMPLE

Dans la figure 5.5, *mammifère* est définit par défaut comme *ne vie pas dans la mer*. De son côté, *baleine* est définit par défaut comme *vie dans la mer*. Lors de l'héritage du concept par défaut *ne vie pas dans la mer*, *baleine* se trouve dans une contradiction de définition du concept par défaut.

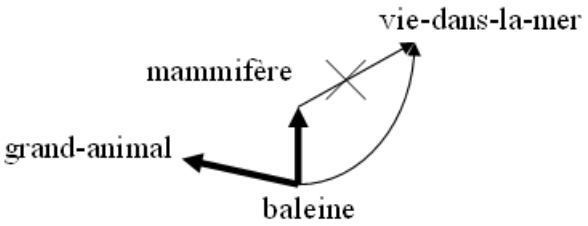

FIG. 5.5 – Exemple du conflit de définition

Pour remédier à ce problème, l'idée consiste de ne considérer que le concept le plus spécifique lors d'un conflit. Dans notre cas, le concept le plus spécifique pour *baleine* est *vie dans la mer* puisque c'est une propriété directe et non pas héritée.

5.5 Conclusion

Dans ce chapitre, nous venons d'introduire la notion de "défaut" et "exception". Nous avons vu que le langage d'ontologie Web OWL et son sous system OWL-DL excluent, explicitement, les définitions par défaut et les exceptions, alors que ce problème est traité dans la logique de description.

Nous avons commencé par discuter la solution proposée dans la LD avec le nouveau langage $AL_{\delta\epsilon}$. Nous avons ensuite présenté $\text{OWL}_{\delta\epsilon}$ comme étant l'extension de OWL, nous avons donc proposé d'ajouter deux nouveaux constructeurs *default* et *exception* tout en présentant une comparaison avec $AL_{\delta\epsilon}$. Nous avons aussi étudié la subsomption dans $\text{OWL}_{\delta\epsilon}$ et nous avons proposé un système équationnel et un algorithme de substitution. Nous avons aussi prouvé que la complexité de calcul pour l'algorithme de substitution est polynômiale, et nous avons également étudié le cas de conflit de définition.

Nous rappelons qu'une définition complète de concepts permet de mieux répondre aux interrogations des internautes, i.e. minimiser les bruits et les silences.

Chapitre 6
CONCLUSION ET PERSPECTIVES

6.1 Conclusion

Dans ce mémoire, nous nous sommes intéressés à la recherche d'information intelligente sur le Web, avec le minimum de bruit et de silence, en utilisant les ontologies basées sur la logique de description. Pour mieux s'approcher d'un Web sémantique nous avons proposé d'étudier le langage OWL, un des langages les plus récents et les plus utilisés pour décrire les ontologies Web. Nous avons alors souligné quelques problèmes et lacunes liés à ce langage et nous avons proposé des solutions pour élargir son contexte. Le but est de construire des ontologies Web avec les définitions les plus complètes possibles. Nous avons focalisé nos recherches sur les solutions proposées dans le cadre des logiques de description et nous avons proposé d'étendre le langage OWL pour le langage $\text{OWL}_{\delta\epsilon}$. Cette extension consiste à prendre en compte les aspects de défaut et d'exception qui manquaient à OWL.

Dans $\text{OWL}_{\delta\epsilon}$, nous avons proposé deux nouveaux constructeurs et un système équationnel. Nous avons aussi prouvé que la complexité de calcul de l'algorithme proposé est polynômiale et nous avons proposé des solutions pour le cas de conflit de définitions.

Par ailleurs, la recherche menée tout au long de ce mémoire nous a permis de :

- Cerner le problème de recherche d'information : connaître les principes des moteurs de recherches, étudier les technique du TAL pour la recherche d'information et voir la nécessité de rendre le Web plus sémantique.
- Prendre une connaissance plus approfondie des ontologies informatiques et acquérir une connaissance riche sur les logiques de description. Cette double connaissance nous a permis de voir l'intérêt de combiner les ontologies avec les logiques de description.

– Étudier en détail le principe de fondement du OWL sur RDF/XML et les différentes caractéristiques de ce langage.

6.2 Perspectives

Le travail que nous envisageons de faire en premier lieu consiste à développer un pseudo moteur de recherche dans lequel l'algorithme proposé sera intégré. Nous avons vu que la complexité de cet algorithme est polynômiale et donc l'ajout des aspects défaut et exception ne risque pas d'altérer sensiblement la complexité du moteur. Par contre, la pertinence de la recherche, prenant en compte les valeurs du bruit et du silence, ne peut que s'améliorer.

Nous étudierons les différents cas possibles de combinaisons entre les différents constructeurs et nous élargirons encore d'avantage notre système équationnel. Les constructeurs proposés peuvent nécessiter une définition plus large afin de les adapter aux différentes combinaisons possibles (avec $unionOf$ par exemple).

Ramener l'étude au domaine des instances pour détecter les concepts dont l'objet est une instance et développer un raisonnement pouvant vérifier l'état des instances : sûre, probable, typique ou exceptionnel.

Tous ces points ne vont que contribuer à développer un système complet permettant de mettre en pratique un langage d'ontologie Web intelligent. L'intelligence ainsi obtenue est-elle celle voulue ... c'est relatif !

REFERENCES

[ABN91] Steven P. Abney. *Parsing by chunks*. In Robert C. Berwick, Steven P. Abney et Carol Tenny, éditeurs, Principle-Based Parsing : computation and psycholinguistics, pages 257-278. Kluwer Academic Publisher, Boston, MA, 1991.

[ARA97] A. T. Arampatzis, C. H. A. Koster et T. Tsoris. *IRENA : Information retrieval engine based on natural language analysis*. In Proceedings, Intelligent Multimedia Information Retrieval Systems and Management (RIAO'97), pages 159-175, Montreal, 1997.

[ARA98] A. T. Arampatzis, T. Tsoris, C. H. A. Koster et Th. P. van derWeide. *Phrase-based information retrieval*. Information Processing and Management, 34(6) :693-707, 1998.

[ATT86] G. Attardi, M. Simmi *A Description-Oriented Logic for Building Knowledge Bases*. In Proceedings of the IEEE, vol. 74, n°. 10, pp.1335-1344, octobre 1986.

[BAA03] F. Baader, D. Calvanese, D. Mcguinness, D. Nardi Et P. Patel Schneider, *The description logic handbook*, Cambridge (UK) : Cambridge university press, 2003.

[BCH03] S. Bechhofer, I. Horrocks, P. F. Patel-Schneider, *Tutorial on OWL* , ISWC, Sanibel Island, Florida, USA, october 2003

[BCMNP03] F. BAADER, D. CALVANESE, D. MCGUINNESS, D. NARDI et P. PATEL SCHNEIDER, Eds. *The description logic handbook*. Cambridge (UK) : Cambridge university press, 2003.

[BDS93] M. Buchheit, F. Donini, and A. Shaerf. *Decidable reasoning in terminological knowledge representation systems*. Journal of Artificial Intelligence Research, 1 :109-138, 1993.

[BER01] T. Berners-Lee, J. Hendler, and O. Lassila. *The semantic web : A new form of web content that is meaningful to computers will unleash a revolution of new possibilities*, Scientific American, - :35-43, May 2001.

[BHMP92] M.J. Blosseville, G. Hébreil, M.G. Monteil, et N. Pénot. *Automatic classification : natural language procedding, statstical analysis and expert techniques used together*. Dans : Conference on Research and Development in Information Retrieval, pages 51-58, 1992.

[BIK97] D.M. Bikel, S. Miller, R. Schwartz and R. Weischedel. *Nymble : a high-performance learning name-finder*. In Proceedings, 5th Conference on Applied Natural Language Processing (ANLP'97), pages 194-201, Washington, 1997.

[BL99] T. Berners-Lee. *Weaving the Web*. Harper, San Francisco, 1999.

[BLHL01] T. Berners-Lee, J. Hendler, and O. Lassila. *The semantic web : A new form of web content that is meaningful to computers will unleash a revolution of new possibilities*. Scientific American, - :35-43, May 2001.

[BMT99] F. Baader, R. Molitor, and S. Tobies. *Tractable and decidable fragments of conceptual graphs*. In Proceedings of ICCS'99, Blacksburg, VA, USA, volume LNAI 1640, pages 480-493. Springer-Verlag, 1999.

[BOT96] S.P. Botley, éditeur. *Approaches to Discourse Anaphora : Proceedings of the DAARC Colloquium*. Lancaster University, Lancaster, 1996.

[BOU93] D. Bourigault. *An Endogenous Corpus-based Method for Structural Nous Phrase Disambiguation*. In Proceedings of the 6th Conference of the European Chapter of the Association of Computational Linguistics (EACL '93), Utrecht, pp. 81-86, 1993.

[BRA85] R.J. BRACHMAN, *I Lied about the Trees Or, Defaults and Definitions in Knowledge Representation*. The A.I. Magazine, vol. 6, n°. 3, pp.80-93, 1985.

[BRA91] R.J. BRACHMAN, *LIVING WITH CLASSIC : When and How to Use a KL-ONE-like Language, in Principles of Semantic Networks*. Explorations in the Representation of Knowledge. J.Sowa, Morgan Kaufman Publi., chapitre 14, pp.401-456, 1991.

[BRI92] E. Brill. *A simple rule-based part of speech tagger*. In Proceedings, 3rd Conference on Applied Natural Language Processing (ANLP'92), pages 152-155, Trento, 1992.

[BRI95] E. Brill. *Transformation-based error-driven learning and natural language processing : A case study in part-of-speech tagging*. Computational Linguistics, 21(4) :543-565, 1995.

[BS00] F. Baader and U. Sattler. *Tableau algorithms for description logics*. In In

Proceedings of Tableaux 2000, University of St Andrews, Scotland. Springer-Verlag, 2000.

[CAL88] P. Ceeseman et al. "AutoClass : a Bayesian classification system". Dans : Fifth International Conference on Machine Learning, pages 53-56. Morgan Kaufmann Publishers, Inc., 1988.

[CAR88] J.G. Carbonell et R.D. Brown. *Anaphora resolution :A multistrategy approach*. In Proceedings, 12th International Conference on Computational Linguistics (COLING'88), pages 96-101, Budapest, 1988. ACL.

[CF98] P. Coupey and C. Faron. *Towards correspondence between conceptual graphs and description logics*. In Proceedings of ICCS'98, Montpellier, France, volume LNAI 1453, pages 165-178. Springer-Verlag, 1998.

[CHA95] J-P. Chanod et P. Tapanainen. *Statistical and constraintbased taggers for French*. In Proceedings of the 7 th EACL, Dublin, Ireland, 1995.

[CLT03] N. Cullot, C. Parent, S. Spaccapietra, C. Vangenot, *A contribution to the DL/DB debate*, Proceedings ot the first International Workshop on Semantic Web and Database (SWDB'20003), Co-located VLDB'2003, Berlin, Germany, September 2003, pp 109-130.

[CMS98] M. Chein, M.-L. Mugnier, and G. Simonet. *Nested graphs : A graph-based knowledge representation model with fol semantics*. In Proceedings of KR'98, Trento, Italy, pages 524-534. Morgan Kaufmann Publishers, 1998.

[CO90] F. Can and E.A. Ozkarahan *Concepts and effectiveness of the cover coefficient based clustering methodology for text databases*. Dans : ACM Transactions on database Systems, volume 15, pages 482-517, 1990.

[CON91] P. Constant. *Analyse syntaxique par couche*. PhD thesis, Ecole Nationale Supérieure des Télécommunications, Paris, 1991.

[CON95] P. Constant. *L'analyseur linguistique Sylex*. Ecole d'été du CENT, 1995.

[COU97] P. Coupey and C. Fouqueré. *Extending Conceptual Definitions with Default Knowledge*. In Computational Intelligence, Volume 13, Number 2 (1997)

[CSS98] Cohn, L. Schubert, and S. C. Shapiro, editors, *Principles of Knowledge Representation and Reasoning*. Proceedings of the Sixth International Conference (KR'98), pages 636-647. Morgan Kaufmann Publishers, San Francisco, California, June 1998.

[DEB82] F. Debili. *Analyse Syntaxico-Sémantique Fondée sur une Acquisition Automatique de Relations Lexicales-Sémantiques*. Thèse de Doctorat d'Etat en

Sciences Informatiques, Université of Paris XI, Orsay, 1982.

[DGL96] G. De Giacomo and M. Lenzerini. *Tbox and abox reasoning in expressive description logics*. In In Proc. of the 5th International Conference on Principles of Knowledge Representation and Reasoning (KR'96), pages 316-327. Morgan Kaufmann Publishers, 1996.

[DGLL00] G. De Giacomo, Y. Lesperance, and H. Levesque. *Congolog : a concurrent programming language based on the situation calculus*. Artificial Intelligence Journal, pages 169-209, 2000.

[DS04] M. Dean and G. Schreiber, eds. *OWL Web Ontology Language Reference*. W3C Recommendation 10 February 2004.

[FAG87] J.L. Fagan. *Experiments in Automatic Phrase Indexing for Document Retrieval : A Comparison of Syntactic and Non-syntactic Methods*. PhD Thesis in Philosophy, Cornell University, 1987.

[FDES98] D. Fensel, S. Decker, M. Erdmann, and R. Studer. *Ontobroker : Or how to enable intelligent access to the www*. In Proceedings of KAW 98, Banff, Canada, 1998.

[FEL98] C. Fellbaum, éditeur. *WordNet : An Electronic Lexical Database*. MIT Press, Cambridge,MA, 1998.

[FEN01] D. Fensel, F. van Harmelen, I. Horrocks, D.L. McGuinness, P.F. Patel-Schneider,*OIL : an ontology infrastructure for the Semantic Web*, IEEE Intell. Syst. 16 (2) (2001) 38-45.

[FENS98] D. Fensel, S. Decker, M. Erdmann, and R. Studer. *Ontobroker : Or how to enable intelligent access to the www*. In Proceedings of KAW 98, Banff, Canada, 1998.

[FHM01] D. Fensel, F. van Harmelen, I. Horrocks, D.L. McGuinness, P.F. Patel-Schneider. *OIL : an ontology infrastructure for the Semantic Web*. IEEE Intell. Syst. 16 (2) (2001) 38-45.

[GAU03] E. Gaussier, M-H. Stefanini. *Assistance intelligente à la recherche d'informations*. Edition Hermes, 2003, 319 pages.

[GBMS99] C.H. Goh, S. Bressan, E. Madnick, M.D. Siegel. *Context interchange : New features and formalisms for the intelligent integration of information*. ACM Transactions on Information Systems, vol. 17, n°3, 1999, p. 270-293.

[GRU93a] T.R. Gruber. *Towards principles for the design of ontologies used for knowledge sharing*. In Roberto Poli Nicola Guarino, editor, International

Workshop on Formal Ontology, Padova, Italy, 1993.

[GRU93b] T. Gruber, *A translation approach to portable ontology specification*, Knowledge Acquisition, 7, 1993.

[HAA01] V. Haarslev, R. Möller, *RACER system description*, In Proceedings of the International Joint Conference on Automated Reasoning (IJCAR 2001), Berlin, 2001.

[HAR92] D. Harman. *Ranking Algorithms*. In Frakes and Baeza-Yates, chapter 14, 1992.

[HEF99] J. Heflin, J. Hendler, S. Luke. *SHOE : A Knowledge Representation Language for Internet Applications*. Technical Report CS-TR-4078, Department of Computer Science, University of Maryland, 1999.

[HHS98] J. Heflin, J. Hendler, and Luke S. *Reading between the lines : Using shoe to discover implicit knowledge from the web*. In Proceedings of the AAAI Workshop on Artificial Intelligence and Information Integration, pages 51-57. AAAI Press, 1998.

[HHS99] J. Heflin, J. Hendler, S. Luke, *SHOE : A Knowledge Representation Language for Internet Applications* ,Technical Report CS-TR-4078, Department of Computer Science, University of Maryland, 1999.

[HIN90] D. Hindle. *Noun Classification from Predicate-argument Structures*. Meeting of the Association for Computational Linguistics, pages 268-275, 1990.

[HIR81] G. Hirst. *Anaphora in Natural Language Understanding : A Survey*. Springer-Verlag, Berlin, 1981.

[Hor98] I. Horrocks. *Using an expressive description logic : FaCT or fiction ?* In Proceedings of the Sixth International Conference (KR'98), pages 636-647. Morgan Kaufmann Publishers, San Francisco, California, June 1998.

[HP03a] I. Horrocks, P.F. Patel-Schneider, F. van Harmelen. *From SHIQ and RDF to OWL : the making of a Web Ontology Language*, Journal of Web Semantics, July 2003.

[HP04] I. Horrocks, P.F. Patel-Schneider. *Reducing OWL entailment to description logic satisfiability*. In Proc. of the 2003 Int. Semantic Web Conf. ISWC (2003).

[HST99] I. Horrocks, U. Sattler, and S. Tobies. *Practical reasoning for expressive description logics*. In Proceedings of the 6th International Conference on Logic for Programming and Automated Reasoning, pages 161-180. Springer-Verlag, LNAI 1705, 1999.

[JAC96] C. Jacquemin. *What is the tree that we see through the window : A linguistic approach to windowing and term variation.* Information Processing and Management, 32(4) :445-458, 1996.

[Kar71] S-J. Karen. *Automatic Keyword Classification for Information Retrieval.* Butterworth, London, 1971.

[Kauf93] M. Kaufmann. *Defense Advanced Research Projects Agency.* Fifth Message Understanding Conference (MUC-5), San Francisco, Ca, 1993.

[Kauf95] M. Kaufmann. *Defense Advanced Research Projects Agency.* MUC-6 : Proceedings of the Sixth Message Understanding Conference, Columbia, Maryland, 1996.

[KAZ86] T. Kazmarek, R. Bates, G. Robins. *Recent Developments in NIKL.* In Proceedings AAAI 86, Philadelphia, PA, pp.978-987, 1986.

[KOH95] T. Kohonen. *Self Organizing Maps.* Springer, 1995.

[LAM02] G. Lame. *Construction d'ontologies à partir de textes : une ontologie du droit dédiée à la recherche d'informations sur le Web.* PhD thesis, Ecole des Mines de Paris, 2002.

[LBB89] D. Lewis, W. Bruce Croft et N. Bhandaru. *Language oriented information retrieval.* International Journal of Intelligent Systems, 4 :285-318, 1989.

[LEC97] M. Leclère. *Reasoning with type definitions.* In Proceedings of ICCS'97, Seattle, WA, volume LNAI 1257, pages 401-415. Springer-Verlag, 1997.

[LEF00] P. Lefèvre. *La recherche d'information, du texte intégral au thésaurus.* Hermes Science, Paris, 2000.

[LH00] A. Lelu et M. Hallab. *Consultation floue de grandes listes de formes lexicales simples et composes : un outil préparatoire pour l'analyse de grands corpus textuels.* Actes des JADT'2000, Lausanne, Mars 2000.

[LR96] A. Levy and M.-C. Rousset. *Carin : A representation language combining horn rules and description logics.* In Proceedings of ECAI'96, pages 323-327, 1996.

[LSRH97] S. Luke, L. Spector, D. Rager, and J. Hendler. *Ontology based web agents.* In Proceedings of the 1st International Conference on Autonomous Agent, 1997.

[LUH57] H.P Luhn. *A statistical approach to mechanized encoding and searching of literary information.* IBM Journal of Research and Development, 4(4), 600-605, 1957.

[MAL83] M.L. Malagnini. *A classification algorithm for star-galaxy counts*. In Statistical Methods in Astronomy, pages 69-72, November 1983.

[MAR97] P. Martin. *The webkb set of tools : a common scheme for shared www annotations, shared knowledge bases and information retrieval*. In Proceedings of 5th International Conference on Conceptual Structures (ICCS 97), volume LNAI 1257, pages 585-588, Seattle, USA, 1997. Springer Verlag.

[MC96] M.-L. Mugnier and M. Chein. *Représenter des connaissances et raisonner avec des graphes*. Revue d'Intelligence Artificielle, 10(1) :7-56, 1996.

[MCD93] D. McDonald. *Internal and external evidence in the identification and semantic categorization of proper names*. In Branimir Boguraev et James Pustejovsky, éditeurs, Corpus Processing for Lexical Acquisition, pages 61-76. MIT Press, Cambridge (Mass.), 1993.

[McGRE91] R.M. MAC GREGOR. *Inside the LOOM Description Classifier*. SIGART bulletin, vol. 2, n°. 3, Special Issues on Implemented Knowledge Representation and Reasonong Systems, juin, 1991.

[MER97] Mercier. *Analyse, Indexation documentaire dans un centre de documentation*. 1997.

[MH69] J. McCarthy and P. Hayes. *Some philosophical problems from the standpoint of artificial intelligence*. Machine Intelligence, 4 :463-502, 1969.

[MIK99] Andrei Mikheev. *Named entity recognition without gazetteers*. In Proceedings, 9th Conference of the European Chapter of the Association for Computational Linguistics (EACL'99), pages 1-8, Bergen, 1999. ACL.

[MIL90] George A. Miller. *Wordnet : An on-line lexical database. International* Journal of Lexicography, 3(4), 1990.

[MIN75] M. Minsky. *A Framework for Representing Knowledge*. In Patrick Henry Winston (ed.), The Psychology of Computer Vision, McGraw-Hill, New York, 1975

[MIT98] R. Mitkov, L. Belguith et M. Stys. *Multilingual robust anaphora resolution*. In Proceedings of the Third International Conference on Empirical Methods in Natural Language Processing (EMNLP-3), pages 7-16, Granada, 1998. ACL.

[MOE00] M.-F. Moens. *Automatic Indexing and Abstracting of Document Texts (The Kluwer International Series on Information Retrieval 6)*. Kluwer Academic Publishers : Boston, 2000.

[MSZ01] S. McIlraith, T.C. Son, and H. Zeng. *Semantic web services*. IEEE Intelligent Systems, 16(2) :46-53, 2001.

[MSZ02] S. McIlraith, T.C. Son, and H. Zeng. *Adapting golog for composition of semantic web services*. In Proceedings of the Eighth International Conference on Knowledge Representation and Reasoning (KR2002), Toulouse, France, 2002.

[NAP97] A. Napoli. *Une introduction aux logiques de descriptions*. Rapport de Recherche RR 3314, INRIA, 1997.

[NEB88] B. NEBEL. *Computational Complexity of Terminological Reasoning in BACK*. Artificial Intelligence vol. 34, n°. 3, pp.371-383, 1988.

[OK03] B. Omelayenko et M. Klein, Eds. *Knowledge transformations for the semantic web*. Amsterdam (NL) : IOS press, 2003.

[OUE99] R. Oueslati. *Aide à l'acquisition de connaissances à partir de corpus. Thèse de doctorat*. Université Louis Pasteur Strasbourg, 1999.

[PAR98] P. Paroubek. *GRACE : Grammaires et ressources pour les analyseurs de corpus et leur évaluation*. page WWW http ://m17. limsi.fr/TLP/grace/, LIMSI, 1998.

[PAT90] P.F. Patel-Schneider, B. Qwsnicki-Klewe, A. Kobsa, N. Guarino, R. Mac Gregor, W.S. Mark, D.L. MC Guinness, B.Nebel, A. Schmiedel, J. Yen, *Term Subsumption Languages in Knowledge Representation*. Workshop of TSL'89, AI Magazine, vol. 11, n°. 2, 1990.

[PIE04] G. Pierra, *Introduction au langage EXPRESS - Polycopié*, Cours MO2, janvier 2004.

[PIE95] Pierre Zweigenbaum et Consortium MENELAS. *MENELAS : coding and information retrieval from natural language patient discharge summaries*. In Advances in Health Telematics, pages 82-89, 1995.

[PLM98] P. Poinçot, S. Lesteven and F. Murtagh. *Comparison of Two Documents Similarity Search Engines*. Dans : ASP Conf. Ser. 153 : Library and Information Services in Astronomy III, pages 85+, 1998.

[PTL93] F. Pereira, N. Z. Tishby, and L. Lee. *Distributional Clustering of English Words*. In 30th Annual Meeting of the Association for Computational Linguistics, pages 183-190, Columbus, Ohio, 1993.

[RAS92] E. Rasmussen . *Information Retrieval : Data structures and algorithms*,

chapter 16 : Clustering Algorithms, pages 419, 442 W.B.Frakes and R Baeza-Yates, prentince Hall edition, 1992.

[RFO96] F. Rousselot, P. Frath P and R. Oueslati. *Extracting Concepts and Relations from Corpora*. In Proceedings of the Corpus-Oriented Semantic Analysis Workshop of ECAI'96 Budapest p.74-78, 1996.

[RIJ79] C. J. van Rijsbergen. *Information Retrieval*. Butter-worths, London, 2nd edition, 1979

[SAL71] G. Salton. *The SMART Retrieval System*. Experiments in Automatic Document Processing. Prentice Hall, 1971.

[SB88] G. Salton and C. Buckley. *Term weighting approaches in automatic text retrieval*. Information Processing and Management, vol. 24, no. 5, pages 513-523, 1988.

[SCH83] J.G. Schmolze, T.A Lipkis. *Classification in the KL-ONE Knowledge Representation System*. In Proceedings of the 8th. IJCAI, Karlsruhe, Germany, 1983.

[SHM98] C. Silverstein, M. Henzinger, and H. Marais. *Analysis of a Very Large Altavista Query Log*. Technical note #1998-014, Digital SRC, Oct. 1998.

[SIL93] M. Silberztein. *Dictionnaires électroniques et analyse automatique de textes : Le système INTEX*. Masson, Paris, 1993.

[SM83] G. Salton and M. J. McGill. *Introduction to modern information retrieval*. McGraw-Hill, New York, 1983.

[SM90] F.A. Smadja, K. McKeown. *Automatically Extracting and Representing Collocations for Language Generation*. In Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics, pages 252-259, 1990.

[SOW84] J. Sowa. *Conceptual Structures : Information Processing in Mind and Machine*. Addison-Wesley, Reading, MA, 1984.

[STA05] Université de Stanford. *The Protégé Ontology Editor and Knowledge Acquisition System*. http ://protege.stanford.edu/ (en ligne au 16 juin 2005).

[TES59] L. Tesniere. *Elément de syntaxe structurale, Klincksieck*. Paris, 1959.

[TOB01] S. Tobies, *Complexity Results and Practical Algorithms for Logics in Knowledge Representation*, Ph.D. Thesis, LuFG Theoretical Computer Science, RWTH Aachen, Germany, 2001.

[VAN79] C. J. van Rijsbergen. *Information Retrieval*. Butter- worths, London, 2nd

edition, 1979.

[VER98] J. Vergne and E. Giguet. *Regards théoriques sur le tagging*. In Pierre Zweigenbaum, éditeur, Actes de TALN 1998, pages 22-31, Paris, juin 1998.

[VER99] J. Vergne. *Étude et modélisation de la syntaxe des langues à l'aide de l'ordinateur - analyse syntaxique automatique non combinatoire*. Mémoire d'habilitation à diriger des recherches, Université de Caen, 1999.

[VOL04] R. VOLTZ. *Web Ontology Reasoning with Logic Databases*. Thèse de doctorat, Université Fridericiana zu Karlsruhe, 2004.

[WAC97] N. Wacholder, Y. Ravin et M. Choi. *Disambiguating proper names in text*. In Proceedings, 5th Conference on Applied Natural Language Processing (ANLP'97),Washington, 1997. ACL.

[WEI93] R. Weischedel, M. Meeter, R. Schwartz, L. Ramshaw et J. Palmucci. *Coping with ambiguity and unknown words through probabilistic models*. Computational Linguistics, 19(2) :359-382, 1993. Special Issue on Using Large Corpora : II.

[WES01] M. Wessel, *Obstacles on the way to qualitative spatial reasoning with Description Logics : some undecidability results*, In Proceedings of the 2001 Description Logic Workshop (DL'001), Stanford, California, USA, 2001.

[WOO75] W. A. Woods. *What's in a Link : Foundations for Semantic Networks*. In D.G.Bobrow & A.M.Collins (eds.), Representation and Understanding : Studies in Cognitive Science, 35-82, Academic Press, New York, 1975.

[ZAI04] H. ZAIT. *Conception d'une Architecture d'Integration de Sources de Données Hétérogènes Basée sur Ontologie*, Mémoire du Master Informatique, LISI-ENSMA, 2004.

REFERENCES WEB

[BCH04] S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D.L. McGuinness, P.F. Patel-Schneider,L.A. Stein, *OWL Web Ontology Language Reference*, http ://www.w3.org/TR/owl-ref/, 2004

[CHM01] D. Connolly, F. van Harmelen, I. Horrocks, D.L. McGuinness, P.F. Patel-Schneider, *DAML+OIL Reference Description*, http ://www.daml.org/2001/03/reference, 2001.

[CON00] D. Connolly, L. Stein, D. McGuinness. *DAML-ONT Initial Release*, http ://www.daml.org/2000/10/daml-ont.html

[CON01] D. Connolly, F. van Harmelen, I. Horrocks, D.L. McGuinness, P.F. Patel-Schneider. *DAML+OIL Reference Description*, http ://www.daml.org/2001/03/reference

[DAM00a] DAML. *Darpa Agent Markup Language*. http ://www.darpa.org, 2000.

[DAM00b] DAML+OIL. *Darpa Agent Markup Language + Ontology* Interface Language. http ://www.darpa.org, 2000.

[DC00] DC. *Dublin Core Elements Set*. http ://dublincore.org/documents/dces/, 2000.

[DS02] DAML-S. *Darpa Agent Markup Language Services*. http ://www.daml.org/services/, 2002.

[DSBH05] M. Dean, G. Schreiber, S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D.L. McGuinness, P.F. Patel-Schneider et L.A. Stein. *OWL Web Ontology Language - Reference*. http ://www.w3.org/TR/2004/REC-owl-ref-20040210/ (en ligne au 16 juin 2005). Traduction française : La référence du langage d'ontologie Web OWL, http ://www.yoyodesign.org/doc/w3c/owl-ref-20040210/.

[GRA04] J. Grant, D. Beckett, *RDF Test Cases*, http ://www.w3.org/TR/rdf-testcases.

[HAY04] P. Hayes, I. Horrocks, P.F. Patel-Schneider, *OWL Web Ontology Language Semantics and Abstract Syntax* , http ://www.w3.org/TR/owl-semantics/

[HEF04] J. Heflin. *OWL Web Ontology Language - Use Cases and Requirements*.

http ://www.w3.org/TR/2004/REC-webont-req-20040210/ (en ligne au 16 juin 2005). Traduction française : Les cas et conditions d'utilisation du langage d'ontologie Web OWL,
http ://www.yoyodesign.org/doc/w3c/webont-req-20040210/.

[HP03b] I. Horrocks, P.F. Patel-Schneider, *A Proposal for an OWL Rules Langage* , http ://www.cs.man.ac.uk/~horrocks/DAML/Rules , 2003

[HPB04] I. Horrocks, P. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, M. Dean. *SWRL : A Semantic Web Rule Language Combining OWL and RuleML.* W3C Submission, 2004. http ://www.w3.org/Submission/SWRL/

[ML00] Rule ML. *Rule MetaLanguage.* http ://www.dfki.uni-kl.de/ruleml/, 2000.

[OIL00] OIL. *OIL : Ontology Inference Layer.* http ://www.ontoknowledge.org/oil, 2000.

[OWL02] OWL. *OWL : Ontology Web Language.*
http ://www.w3.org/2001/sw/WebOnt/, 2002.

[RDF00] RDFS. *Resource Description Framework Schema Specification 1.0.*
http ://www.w3.org/TR/2000/CR-rdf-schema-20000327/, 2000.

[RDF99] RDF. *Resource Description Framework Model and Syntax Specification.*
http ://www.w3.org/TR/REC-rdf-syntax/, 1999.

[WSD02] WSDL. *Web Service Description Language.* http ://www.w3.org/TR/wsdl, 2002.

Annexe A

Représentation de l'ontologie Université en Protégé

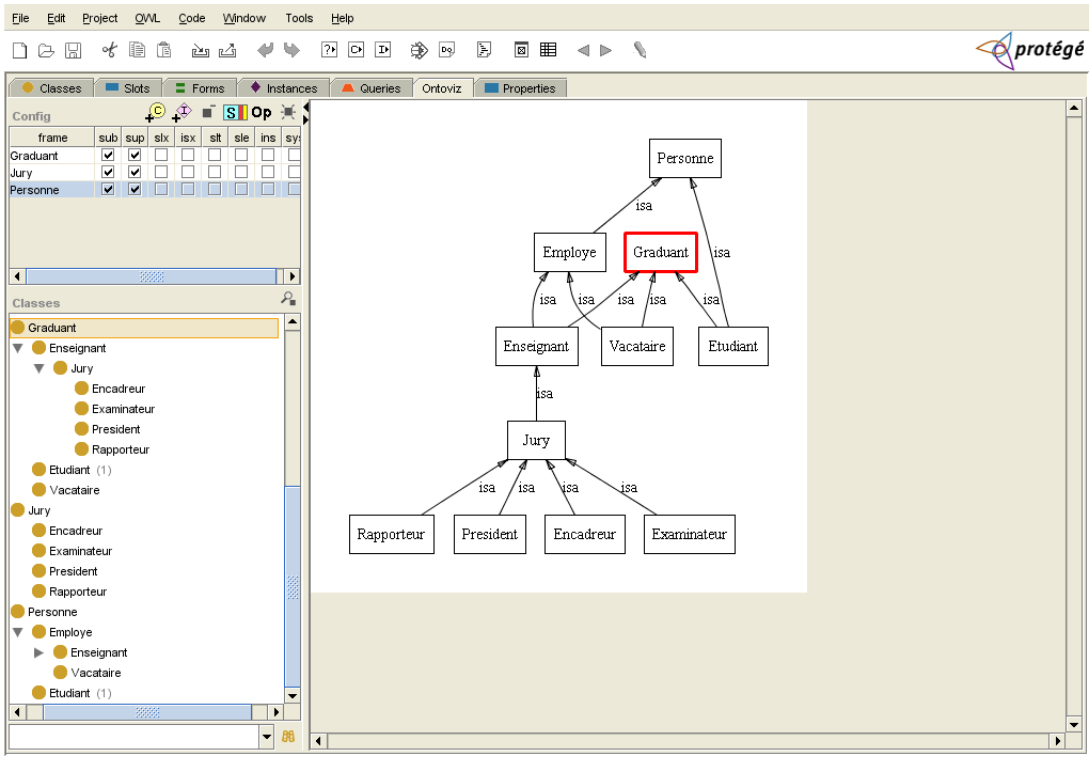

FIG. 0.1 – Représentation de l'ontologie Université dans Protégé

Code source OWL/RDF de l'ontologie Université

```
< ?xml version="1.0" ?>
<rdf :RDF
        xmlns :j.0="http ://protege.stanford.edu/plugins/owl/protege#"
        xmlns :rdf="http ://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns :xsd="http ://www.w3.org/2001/XMLSchema#"
        xmlns :rdfs="http ://www.w3.org/2000/01/rdf-schema#"
        xmlns :owl="http ://www.w3.org/2002/07/owl#"
        xmlns="http ://www.owl-ontologies.com/unnamed.owl#"
      xml :base="http ://www.owl-ontologies.com/unnamed.owl">
      <owl :Ontology rdf :about="Université"/>
      <owl :Class rdf :ID="President">
        <rdfs :subClassOf>
          <owl :Class rdf :ID="Jury"/>
        </rdfs :subClassOf>   </owl :Class>
      <owl :Class>
        <owl :unionOf rdf :parseType="Collection">
        <rdf :Description rdf :about="http ://www.w3.org/2002/07/owl#Thing"/>
          <owl :Class rdf :ID="Etudiant"/>
        </owl :unionOf>
      </owl :Class>
      <owl :Class>
        <owl :unionOf rdf :parseType="Collection">
        <rdf :Description rdf :about="http ://www.w3.org/2002/07/owl#Thing"/>
          <owl :Class rdf :about="#Etudiant"/>
        </owl :unionOf>
      </owl :Class>
      <owl :Class rdf :ID="Rapporteur">
        <rdfs :subClassOf>
          <owl :Class rdf :about="#Jury"/>
        </rdfs :subClassOf>
      </owl :Class>
      <owl :Class rdf :ID="Graduant"/>
      <owl :Class rdf :ID="Enseignant">
        <rdfs :subClassOf>
          <owl :Class rdf :ID="Employe"/>
        </rdfs :subClassOf>
        <rdfs :subClassOf rdf :resource="#Graduant"/>
      </owl :Class>
      <owl :Class rdf :ID="Encadreur">
        <rdfs :subClassOf>
          <owl :Restriction>
```

```
          <owl :onProperty>
            <owl :DatatypeProperty rdf :ID="est-encadreur-de"/>
          </owl :onProperty>
          <owl :maxCardinality rdf :datatype="http ://www.w3.org/2001/
          XMLSchema#int">1</owl :maxCardinality>
        </owl :Restriction>
      </rdfs :subClassOf>
      <rdfs :subClassOf>
        <owl :Class rdf :about="#Jury"/>
      </rdfs :subClassOf>
</owl :Class>
<owl :Class rdf :ID="Examinateur">
      <rdfs :subClassOf>
        <owl :Class rdf :about="#Jury"/>
      </rdfs :subClassOf>
</owl :Class>
<owl :Class rdf :ID="Vacataire">
      <rdfs :subClassOf rdf :resource="#Graduant"/>
      <rdfs :subClassOf>
        <owl :Class rdf :about="#Employe"/>
      </rdfs :subClassOf>
</owl :Class>
<owl :Class rdf :about="#Jury">
      <rdfs :subClassOf rdf :resource="#Enseignant"/>
   <rdfs :subClassOf rdf :resource="http ://www.w3.org/2002/07/owl#Thing"/>
</owl :Class>
<owl :Class>
      <owl :unionOf rdf :parseType="Collection">
      <rdf :Description rdf :about="http ://www.w3.org/2002/07/owl#Thing"/>
        <owl :Class rdf :about="#Enseignant"/>
      </owl :unionOf>
</owl :Class>
<owl :Class rdf :about="#Employe">
      <rdfs :subClassOf>
        <owl :Class rdf :ID="Personne"/>
      </rdfs :subClassOf>
</owl :Class>
<owl :Class>
      <owl :unionOf rdf :parseType="Collection">
      <rdf :Description rdf :about="http ://www.w3.org/2002/07/owl#Thing"/>
        <owl :Class rdf :about="#Rapporteur"/>
      </owl :unionOf>
</owl :Class>
<owl :Class rdf :about="#Etudiant">
```

```
      <rdfs :subClassOf rdf :resource="#Graduant"/>
      <rdfs :subClassOf rdf :resource="#Personne"/>
  </owl :Class>
  <owl :Class>
      <owl :unionOf rdf :parseType="Collection">
      <rdf :Description rdf :about="http ://www.w3.org/2002/07/owl#Thing"/>
        <owl :Class rdf :about="#Encadreur"/>
      </owl :unionOf>
  </owl :Class>
  <owl :DatatypeProperty rdf :ID="Test-Project-Slot-17">
      <rdfs :label rdf :datatype="http ://www.w3.org/2001/XMLSchema#string"
      >Test Project-Slot-17</rdfs :label>
    <rdfs :range rdf :resource="http ://www.w3.org/2001/XMLSchema#string"/>
  </owl :DatatypeProperty>
  <owl :DatatypeProperty rdf :ID="salary">
     <rdfs :range rdf :resource="http ://www.w3.org/2001/XMLSchema#float"/>
  </owl :DatatypeProperty>
  <owl :DatatypeProperty rdf :ID="est-encadre-par"/>
  <owl :DatatypeProperty rdf :about="#est-encadreur-de">
      <rdfs :domain>
        <owl :Class>
          <owl :unionOf rdf :parseType="Collection">
       <rdf :Description rdf :about="http ://www.w3.org/2002/07/owl#Thing"/>
            <owl :Class rdf :about="#Encadreur"/>
          </owl :unionOf>
        </owl :Class>
      </rdfs :domain>
  </owl :DatatypeProperty>
  <j.0 :PAL-CONSTRAINT rdf :ID="PAL-CONSTRAINT-14"/>
  <Etudiant rdf :ID="Etudiant-13"/>
</rdf :RDF>
```