Une approche adaptative pour la recherche d'information sur le Web

Cédric Pruski

Université du Luxembourg

Université Paris-Sud

Une approche adaptative pour la recherche d'information sur le Web

THÈSE

présentée et soutenue publiquement le 29 avril 2009

pour l'obtention du

Doctorat de l'université du Luxembourg et de l'université Paris-Sud

(Cotutelle internationale spécialité informatique)

par

Cédric Pruski

Composition du jury

Rapporteurs :	Pr. Marie-Christine Rousset	Université de Grenoble
	Pr. Stefano Spaccapietra	Ecole Polytechnique Fédérale de Lausanne
Examinateurs :	Pr. Pascal Bouvry	Université du Luxembourg
	Pr. Nicolas Spyratos	Université Paris-Sud
Directeurs :	Pr. Nicolas Guelfi	Université du Luxembourg
	Pr. Chantal Reynaud	Université Paris-Sud

Remerciements

Je tiens à remercier tout particulièrement mes directeurs de thèse les professeurs Nicolas Guelfi et Chantal Reynaud pour leur professionnalisme et leurs qualités humaine. L'accomplissement d'un tel travail ne peut se faire sans un encadrement de grande qualité et un environnement favorable qu'ils ont su m'apporter tout au long de cette aventure.

Mes remerciements vont également vers les membres du jury qui m'ont fait l'honneur d'évaluer mes travaux. En ce sens, je tiens à exprimer ma plus profonde gratitude aux rapporteurs, les professeurs Marie-Christine Rousset et Stefano Spaccapietra, pour la lecture approfondie dont a bénéficié ce manuscrit. Je remercie également le professeur Pascal Bouvry pour m'avoir fait l'honneur de présider le jury de ma thèse ainsi que le professeur Nicolas Spyratos pour avoir accepté de faire partie du jury.

Je voudrais ensuite témoigner ma reconnaissance à l'ensemble des personnes qui m'ont fait profiter de leurs conseils et de leurs expériences. Je pense à tous les membres du LASSY de l'université du Luxembourg et de l'équipe IASI-GEMO de l'université Paris-Sud et de l'INRIA Saclay - Île-de-France. Je souhaite remercier plus particulièrement mes collègues de bureau, Benoît Ries, Gilles Perrouin et François Calvier pour avoir rendu les longues journées de travail beaucoup plus agréables et leur souhaite également toute la réussite qu'ils méritent.

Ma sympathie et mes meilleurs sentiments d'amitié vont vers tous les gens qui ont contribués de près ou de loin à l'accomplissement de cette thèse de doctorat. Je pense en particulier à mon frère Franck à Michaël Obara, Paul Sterges, Marcos Da Silveira, Jacques Klein et à tous les autres. J'adresse également mes remerciements les plus chaleureux à toute la famille Sevestre pour leur accueil lors de mes différents séjours à Orsay.

Enfin, ces remerciements ne seraient pas complets sans l'expression de mon très profond amour pour mon père et ma mère, pour l'ensemble de ce que je leur dois, et leur affection. Merci infiniment.

Pour conclure, selon une formule consacrée, je remercie celles et ceux qui n'auraient pas été nommés à travers ces quelques mots par oubli de ma part, et les prie de m'en excuser.

A mes parents

Résumé

Depuis son avènement au début des années 1990, le Web a profondément bouleversé la société contemporaine et ce à plusieurs niveaux. Ce nouvel outil est rapidement devenu incontournable et s'est affirmé comme la plus grande base de données du monde. La popularité sans cesse croissante du Web a généré une dynamique très importante principalement au niveau des données qu'il renferme. En effet, en vertu de l'évolution des connaissances du monde réel, de nouvelles informations sont rajoutées, d'autres retirées et certaines sont modifiées sans cesse sur le Web posant ainsi des problèmes pour retrouver l'information pertinente. Les moteurs de recherche existants ne sont pas capables d'une part de prendre en compte l'évolution des connaissances du Web lorsqu'un utilisateur pose une requête et d'autre part, de comprendre les besoins en information de l'utilisateur pour lui retourner les pages Web répondant à ces besoins. L'apparition du paradigme du Web Sémantique, visant à donner un sens aux données du Web pour les rendre compréhensibles par les machines grâce à l'utilisation d'ontologies, contribue à l'amélioration de la recherche documentaire sur le Web. Cependant, les problèmes posés par l'évolution restent peu pris en compte.

Dans ces travaux, nous nous sommes intéressés à la prise en compte de l'évolution des données du Web dans le but d'améliorer, en terme de pertinence des résultats, la recherche documentaire sur le Web. La solution que nous proposons est basée sur les ontologies, fondement du Web Sémantique, pour représenter les connaissances du domaine de recherche visé par des requêtes ainsi que les vues des utilisateurs sur ce domaine. Dans la solution que nous préconisons, les ontologies sont vues comme des connaissances qui évoluent au cours du temps. Cette solution nous a obligé à considérer l'évolution des ontologies sous deux aspects différents : de manière générale par rapport au domaine visé par la requête et de manière plus particulière par rapport aux points de vue des utilisateurs.

En premier lieu, nous proposons un modèle d'ontologies adaptatives ainsi qu'un processus d'adaptation permettant aux ontologies de s'adapter aux évolutions des connaissances d'un domaine. Le modèle ainsi défini s'appuie sur des idées émises dans les domaines de la psychologie et des sciences naturelles.

Ensuite, nous proposons une exploitation de ce type d'ontologie pour améliorer la recherche documentaire sur le Web. Nous introduisons tout d'abord, des structures de données (les WP-Graphs et W^3Graphs) pour la représentation des données du Web, puis le langage de requête ASK adapté à ces structures pour l'extraction des données pertinentes. Nous proposons également un ensemble de règles d'enrichissement des requêtes ASK basé sur les relations ontologiques et les éléments propres aux ontologies adaptatives des ontologies représentant le domaine visé par la requête et celle représentant les vues des utilisateurs sur le domaine.

Pour finir nous proposons un outil pour la gestion des ontologies adaptatives et la recherche d'information sur le Web ainsi qu'une validation expérimentale des concepts introduits. Cette dernière est basée sur un cas d'étude réaliste pour la recherche d'articles scientifiques publiés à la conférence internationale World Wide Web.

Mots-clés: évolution d'ontologies, Web Sémantique, recherche d'information sur le Web

Abstract

The advent of the Web in the early 90s has deeply upset our society. This new media has rapidly become the greatest database in the world. Moreover, the ever increasing popularity of the Web engendered a huge dynamics with respect to Web data. Actually, by virtue of knowledge evolution, data is permanently added, deleted or updated from the Web which raises important issues regarding Web information retrieval. Existing Web search engines are neither able to take knowledge evolution into account when users submit their queries nor able to understand users' needs in order to return the most relevant information to users. The Semantic Web, proposed in 2001 and which aims at giving a sense to Web data in order to make it machine understandable, helps to improve Web search but knowledge evolution is still problematic.

In this work, we address the problem of taking knowledge evolution for improving Web search in the sense of relevance of the returned results. The advocated solution is based on the use of ontologies, cornerstone of the Semantic Web, for representing both the domain targeted by the query and the profil of the user who submit the query. Ontologies are considered as knowledge that is evolving over time. In consequence, the ontology evolution problem has to be tackled as regards the evolution of the targeted domain but also with respect to the evolution of users' profile.

First of all, we introduce un new paradigm : adaptive ontology as well as a process for making adaptive ontologies smoothly follow evolution of a domain. The so-defined model rely on the adaptation of ideas developed in the field of psychology and biology to the knowledge engineering field.

Then, we propose an approach exploiting adaptive ontologies for improving Web information retrieval. To this end, we first introduce data structures, WPGraphs and W^3Graphs, for representing Web data. We then introduce the ASK query language tailored for the extraction of relevant information from these structures. We also propose a set of query enrichment rules based on the exploitation of ontological elements as well as adaptive ontologies characteristics of the ontology representing the domain targeted by the query and the one representing the view of the user on the domain.

Lastly, we introduce a tool for managing adaptive ontologies and for searching relevant information on the Web as well as an experimental validation of the introduced concepts. We based our validation on the definition of a realistic case study devoted to the retrieval of scientific articles published at the International World Wide Web series of conference.

Keywords: ontology evolution, Semantic Web, adaptive Web IR

Table des matières

Table des figures

Liste des tableaux

Introduction générale

Depuis son avènement au début des années 1990, le Web [Berners-Lee et al., 1992] a révolutionné le monde de l'information principalement en offrant un accès universel à la connaissance. Cette popularité a rapidement fait de cet outil la plus vaste base de données existante de part la quantité et la diversité des documents qu'elle contient. Cependant, le nombre toujours croissant de documents formant «la toile» ainsi que leur perpétuelle évolution soulèvent d'importants problèmes pour les utilisateurs notamment pour trouver l'information pertinente qu'elle renferme. L'interrogation des principaux moteurs de recherche comme Google [1], MSN [2] ou encore Yahoo ! [3], donne très souvent lieu à des résultats relativement pauvres du point de vue de la pertinence car ces applications n'ont pas la faculté de comprendre les besoins des utilisateurs et en plus ne peuvent pas s'adapter ni à l'évolution des données du Web, ni aux caractéristiques des utilisateurs. En conséquence, les utilisateurs se retrouvent dans l'obligation de filtrer manuellement les résultats retournés par ces applications ou encore dans le cas extrême de recommencer le processus d'interrogation en retravaillant de manière souvent conséquente la requête initiale jusqu'à trouver l'information répondant au mieux à leurs besoins initiaux.

Dans ce cadre, nous nous sommes intéressés au développement de techniques destinées à l'amélioration de la recherche de documents pertinents sur le Web. Nous proposons une solution basée sur l'exploitation d'ontologies [Gruber, 1993], fondements du Web Sémantique, pour représenter les connaissances du domaine de recherche visé par des requêtes ainsi que les vues des utilisateurs sur ce domaine. Dans la solution que nous préconisons, les ontologies sont vues comme des connaissances qui évoluent au cours du temps. Cette solution nous a obligé à considérer l'évolution des ontologies sous deux aspects différents : de manière générale par rapport au domaine visé par la requête et de manière plus particulière par rapport aux points de vue des utilisateurs.

1 Contexte de travail

Avant de présenter les différentes problématiques auxquelles nous nous sommes intéressés dans nos travaux, nous allons définir le contexte dans lequel nous nous plaçons. Nous allons d'une part présenter brièvement le Web et ses évolutions et d'autre part introduire la recherche d'information sur le Web.

1. http ://www.google.com
2. http ://www.msn.com
3. http ://www.yahoo.com

1.1 Du Web au Web Sémantique

Depuis sa création en 1989 par Tim berners-Lee [Berners-Lee et al., 1992], le Web est devenu très rapidement à la fois le service le plus populaire d'Internet et la plus grande base de données existante. Son contenu a très rapidement évolué et ce à plusieurs niveaux :

- Tout d'abord, la quantité d'information qu'il contient a littéralement explosé. Le nombre de pages Web constituant la toile est à ce jour estimé à quelques 30 milliards. De plus, l'information contenue dans ces pages Web est en perpétuelle évolution. La nature évolutive des connaissances fait qu'à chaque instant des informations du Web sont modifiées, enlevées ou de nouvelles y sont rajoutées.
- Ensuite, du fait de l'accroissement du nombre d'utilisateurs (avoisinant les 1.1 milliard fin 2006 soit 17% de la population mondiale), le contenu s'est diversifié. Chaque utilisateur ayant des centres d'intérêts différents, le Web fut obligé de suivre ces évolutions en offrant des documents appartenant à une large gamme de domaines allant de la recherche scientifique aux loisirs en passant par le tourisme ou l'actualité.
- Enfin, la dernière évolution notable du contenu du Web réside dans sa nature et sa structure. L'apparition de nouveaux standards tels que XML [1] (eXtended Markup Language) ont permis une structuration plus rigoureuse des données du Web. En plus, la diversification des domaines et les évolutions technologiques ont donné la possibilité aux utilisateurs du Web de diffuser les informations sous forme de vidéo, de son ou d'image renforçant ainsi l'attrait du Web.

Victime de son succès, le Web est rapidement devenu beaucoup trop volumineux pour que les applications ou les utilisateurs puissent exploiter de manière efficace toutes ses données. Le besoin d'une version plus aboutie du Web permettant la gestion de ses données s'est donc rapidement fait ressentir.

Introduit en 2001 par le créateur du Web Tim Berners-Lee, le paradigme du Web Sémantique [Berners-Lee et al., 2001] fait d'abord référence à une vision ambitieuse du Web. Dans cette nouvelle version du Web, ce dernier serait vu comme un vaste espace d'échange de ressources entre individus et machines permettant une exploitation, qualitativement supérieure, de grands volumes d'informations et de services divers et variés. A la différence du Web tel que nous le connaissons, le Web Sémantique devrait voir les utilisateurs déchargés d'une grande partie de leurs tâches fastidieuses de recherche, de construction et de combinaison des résultats, grâce aux capacités accrues des machines à accéder aux contenus des ressources et à effectuer des raisonnements sur ceux-ci.

Le Web dans sa version initiale s'appuyait essentiellement sur la syntaxe des documents ou plus généralement des ressources. Nous entendons par là que seule la structure des documents est bien définie, et que leur contenu, textuel pour la plupart, reste quasi inexploitable par les ordinateurs du réseau. Ce contenu ne peut être interprété que par les humains. Le Web Sémantique a pour ambition de lever cette difficulté. Les ressources du Web seront plus aisément accessibles aussi bien par l'homme que par la machine, grâce à la représentation sémantique formelle de leurs contenus.

Cette représentation sémantique se fait en s'appuyant sur des modèles de représentation des connaissances plus communément appelés *ontologies* [Gruber, 1993]. Une ontologie est l'ensemble structuré des termes et concepts fondant le sens d'un champ d'informations. L'ontologie

1. http ://www.w3.org/XML/

constitue en soi un modèle de données représentant un ensemble de concepts dans un domaine, ainsi que les relations entre ces concepts. Elle peut être employée pour raisonner sur les objets du domaine représenté.

1.2 La recherche d'information sur le Web

La recherche d'information pertinente a toujours été parmi les plus grands challenges depuis la création du Web. En effet, au commencement aucun outil de recherche n'existait. Pour accéder à l'information pertinente, l'utilisateur devait connaître l'adresse URL la localisant. Ainsi, par exemple, pour consulter le contenu de la page Web personnelle de *Cédric Pruski* l'utilisateur devait saisir directement l'adresse URL `http://se2c.uni.lu/users/CP` dans son logiciel de navigation.

L'explosion du nombre de pages Web aidant, il était alors évident que le succès du Web allait dépendre du développement d'outils adaptés à la recherche d'information. C'est alors que les premiers moteurs de recherche sont apparus vers 1995. Des applications comme Yahoo ! et Altavista ont été conçues dans l'intention de faciliter l'accès à l'information du Web en offrant la possibilité à l'utilisateur de saisir un ensemble de mots clés caractérisant l'information recherchée. Dès lors, le moteur de recherche est devenu indispensable à l'utilisateur dans sa quête d'information sur la toile. Le fonctionnement d'une telle application repose sur la construction dynamique d'un index de pages Web à partir de clés. Lorsque l'utilisateur soumet une requête au moteur de recherche, celui-ci l'évalue suivant différentes techniques. Ces méthodes d'évaluation sont souvent basées sur le calcul d'un coefficient de similarité entre la requête soumise et les pages de l'index. Les pages présentant le coefficient les plus importants sont considérées comme les plus pertinentes et sont alors retournées à l'utilisateur. Le calcul du coefficient est donc capital pour le moteur de recherche car il représente un facteur de pertinence. C'est sur cette observation que Larry Page et Sergei Brin développèrent le moteur de recherche Google [Page and Brin, 1998] en 1998 qui allait par la suite révolutionner la recherche de documents sur le Web. Aujourd'hui encore, Google reste la référence en matière de moteur de recherche sur le Web car près de 70% de requêtes effectuées sur le Web lui sont directement soumises.

2 Problématiques

Dans nos travaux nous nous sommes intéressés au problème de la recherche d'information sur le Web et plus particulièrement aux techniques qui permettraient d'améliorer la pertinence des résultats retournés. Le contexte décrit précédemment montre que l'utilisation des moteurs de recherche existants reste le seul moyen efficace de retrouver un document sur le Web. De plus, l'interrogation de ce type d'application se fait par le moyen d'une requête composée, grâce à un langage dédié, de plusieurs mots clés sensés spécifier les besoins en information des utilisateurs. Par conséquent, les deux principaux problèmes qui font obstacle à l'obtention de résultats pertinents sur le Web sont d'une part la façon dont on interroge les moteurs de recherche et d'autre part la manière dont ces applications traitent les requêtes.

Le premier problème fait directement référence à la qualité des requêtes construites par les utilisateurs. L'étude menée par Silverstein et al. [Silverstein et al., 1999] et confirmée par la suite par Jansen et al. [Jansen et al., 2000], montre que les requêtes soumises aux principaux moteurs de recherche sont généralement courtes, car composées de un ou deux mots clés. Ce nombre très limité de mots clés ne permet pas de caractériser de manière précise le domaine de recherche

auquel s'intéresse l'utilisateur. Toutefois, hormis la requête, l'utilisateur n'a aucun autre moyen de spécifier ce domaine de recherche.

Les mots clés utilisés sont souvent ambigus et par conséquent peuvent faire référence simultanément à plusieurs domaines. Prenons par exemple la requête *"publications sur les arbres"* pour illustrer ce problème. Il est très difficile de décider si le terme *arbre* fait ici référence au domaine de la théorie des graphes ou à celui de la botanique ou encore à la généalogie sans faire explicitement référence à un de ces domaines pour lever toutes ambiguïtés.

En plus d'être peu nombreux et la plupart du temps ambigus, les mots clés d'une requête sont souvent simplement mis bout à bout par les utilisateurs au moment de la construction de la requête. Ceci démontre la mauvaise exploitation de l'expressivité des langages de requête et principalement des opérateurs offerts par les moteurs de recherche pour la construction de bonnes requêtes c'est-à-dire des requêtes spécifiant correctement les besoins en information des utilisateurs.

Un problème essentiel dans le choix des mots clés composant la requête est relatif à l'évolution des connaissances et principalement celles du domaine visé par la requête. Cette évolution est très peu prise en compte dans ce choix ce qui par conséquent réduit la qualité des résultats de la recherche. Les difficultés rencontrées par l'utilisateur pour comprendre cette évolution en est la principale raison. Comme elles sont dues principalement à des éléments extérieurs au domaine dont l'utilisateur n'a pas conscience, il est très difficile pour lui de la caractériser précisément. Ensuite, la représentation de cette évolution n'est pas toujours aisée, les modèles de représentation des connaissances existants et principalement les ontologies sont d'une part inadaptés pour représenter les données liées à l'évolution et surtout ils ne bénéficient d'aucun mécanisme d'adaptation qui leur permettrait de suivre de manière rigoureuse les évolutions d'un domaine. Les travaux récents sur l'évolution d'ontologies [Noy and Klein, 2004, Stojanovic, 2004] montrent l'intérêt des communauté du Web Sémantique et de l'Ingénierie des Connaissances pour trouver des solutions adaptées aux problèmes qui en découlent comme la spécification des changements [Plessers et al., 2007], la vérification de la cohérence de l'ontologie évoluée [Haase and Stojanovic, 2005] et la propagation des changements dans le cadre d'ontologies distribuées [Stuckenschmidt and Klein, 2003]. L'exploitation des données relatives à l'évolution d'un domaine doit permettre une amélioration significative de la recherche documentaire sur le Web surtout du point de vue de la pertinence des résultats car ces derniers seront davantage en relation avec le domaine visé.

Le second problème majeur concerne l'évaluation de la requête par les moteurs de recherche. Cette évaluation est faite en tenant compte de différents facteurs.

Le premier d'entre eux est relatif aux caractéristiques structurelles du Web et de la sémantique de son contenu. Dans les approches existantes pour la recherche d'information sur le Web [Page and Brin, 1998, Kleinberg, 1999], seule la structure de graphe du Web, reposant sur les hyperliens reliant les pages, est exploitée. L'algorithme du PageRank implémenté par Google en est la parfaite illustration. La sémantique du contenu des pages Web qui est plus représentative des données du Web que cette simple structure de graphe, n'est pas exploitée par les principales applications pour la recherche sur le Web. Il serait logique que ces dernières au moment d'interpréter les requêtes utilisent la sémantique du contenu pour décider si un document est pertinent ou non. Une exploitation de cette sémantique à travers l'utilisation des concepts définis dans le cadre du Web Sémantique permettrait d'améliorer la recherche documentaire sur le Web.

Le second facteur est relatif aux caractéristiques des utilisateurs. L'application ne peut pas deviner le profil de l'utilisateur en traitant simplement la requête que ce dernier lui aura soumise. Prenons, par exemple pour illustrer ce problème, le domaine de la recherche scientifique.

Un utilisateur ayant un profil de chercheur fondamental n'aura pas les mêmes attentes lors de la recherche d'articles scientifiques qu'une personne ayant plutôt le profil de chercheur appliqué même si ces deux personnes saisissent la même requête. La première personne se verra certainement plus intéressée par des articles contenant des preuves formelles de théorèmes alors que la seconde s'intéressera plus probablement à des papiers présentant, par exemple, des résultats expérimentaux. L'acquisition des informations relatives aux utilisateurs [Teevan et al., 2005], la construction de leurs profils [Sieg et al., 2007] et l'exploitation de ces profils [Chirita et al., 2007] sont donc des problèmes cruciaux. De plus, les connaissances de ces utilisateurs évoluent à travers le temps en fonction, par exemple, de leurs expériences et des nouvelles informations portées à leur connaissance. La manière dont les problèmes posés par l'évolution des connaissances des utilisateurs, comme sa caractérisation et la répercussion des changements qui en découlent au niveau du profil les représentants seront traités aura un impact significatif sur la qualité d'une recherche documentaire.

Enfin, le troisième et dernier facteur concerne l'évaluation des requêtes par les moteurs de recherche. La façon dont ces applications interprètent la requête est souvent mal comprise par les utilisateurs. Ces derniers se voient retourner un ensemble de pages Web auquel ils ne s'attendaient pas du tout au moment où ils ont posé leur requête. Ceci est le fruit d'une interprétation différente de la requête par les utilisateurs et l'application. Ce facteur est déterminant dans la popularité et l'utilisabilité de l'application. L'interprétation de la requête est également liée au niveau d'expressivité du langage de requête. Plus celui-ci offre des possibilités pour spécifier des besoins en information précis et détaillés, plus les requêtes construites suivant ce langage seront sujettes à une mauvaise interprétation par les utilisateurs au moment de construire la requête puis par l'application pour décider de la pertinence des pages Web.

Les deux problèmes majeurs ainsi définis (la construction et l'évaluation des requêtes) doivent être traités avec pour objectif prioritaire l'obtention de résultats pertinents lors d'une recherche documentaire sur le Web. Nous entendons par là, la sélection des documents du Web les plus à jour en rapport avec, à la fois le domaine de recherche visé par l'utilisateur, et les vues que ce dernier a de ce domaine.

3 Contributions

Dans le but d'améliorer la pertinence des résultats d'une recherche d'information sur le Web, nous avons été amené à répondre aux différents problèmes évoqués au cours de la section précédente. En conséquence, les contributions scientifiques apportées par nos travaux sont les suivantes :

1. La proposition d'un modèle d'ontologies adaptatives et d'un processus d'adaptation de ces ontologies en réponse aux problèmes posés par la non prise en compte de l'évolution d'un domaine dans le processus de recherche d'information et principalement la caractérisation et la représentation de l'évolution d'un domaine. Ces propositions sont le fruit d'une réflexion sur la notion d'évolution des connaissances et de l'adaptation de certains concepts développés dans le domaine de la psychologie et des sciences naturelles.
2. L'exploitation des ontologies adaptatives pour la recherche d'information sur la toile. Dans ce sens, nous utilisons les ontologies adaptatives pour représenter le domaine de recherche auquel fait référence une requête ainsi que le profil de l'utilisateur ayant émis cette requête. Nous proposons également un mécanisme d'enrichissement de requêtes pour le Web basé sur certaines relations ontologiques ainsi que sur les caractéristiques particulières des

ontologies adaptatives représentant ces deux domaines (domaine de recherche et profil utilisateur). Enfin, nous avons adapté, d'une part, des structures de données (les WPGraphs et W^{3}Graphs) [Guelfi and Pruski, 2006] pour la représentation du Web et de son contenu, d'autre part, le langage de requête ASK aux propriétés des ontologies adaptatives. Ces propositions permettent de répondre aux problèmes liés à la qualité des requêtes posées par les utilisateurs ainsi qu'à leur interprétation.

3. L'intégration des technologies du Web classique et du Web Sémantique. Nous montrons, à travers l'utilisation combinée des ontologies adaptatives et des moteurs de recherche usuels, comment exploiter ces technologies. Du fait de la faible quantité de pages Web annotées grâce aux ontologies, nous montrons comment exploiter les concepts du Web Sémantique à d'autres fins afin d'accélérer leur utilisation.

4. Le développement d'un outil (TARGET) pour la recherche d'information sur le Web et l'évolution des ontologies. L'outil utilise Google comme interface avec le Web et le langage de requêtes ASK que nous avons développé. Une première requête non enrichie est soumise à Google afin d'extraire une première série de pages Web. L'ensemble des pages ainsi obtenu est converti en WPGraphs et W^{3}Graphs, des structures formelles du Web construites selon l'ontologie adaptative du domaine sélectionnée par l'utilisateur. Une deuxième requête, enrichie cette fois, par application de règles d'expansion de requêtes basées à la fois sur l'ontologie du domaine et celle représentant le profil de l'utilisateur, est vérifiée sur ces graphes afin de sélectionner les pages les plus pertinentes. Enfin, ces pages sont stockées dans un corpus de documents dont l'analyse fournira des informations pour faire évoluer l'ontologie du domaine. Le schéma fonctionnel de l'outil ainsi conçu est résumé sur la figure 1.

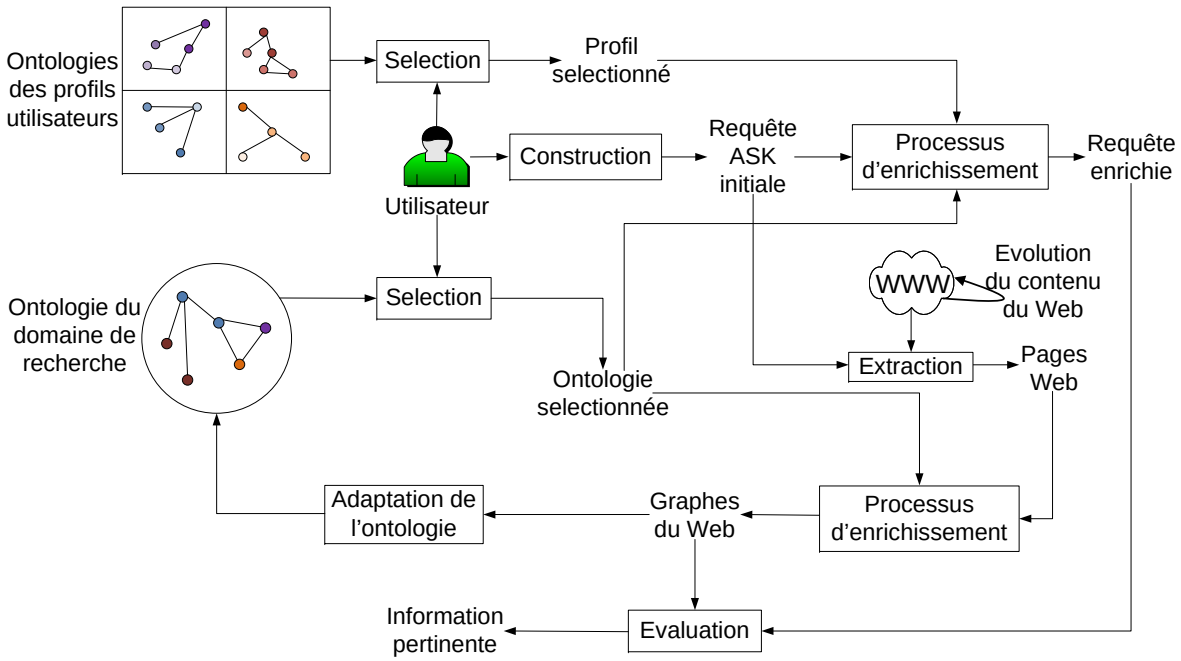


FIGURE 1 – Schéma fonctionnel de l'outil TARGET

5. L'évaluation expérimentale de l'ensemble des concepts développés dans le cadre de nos travaux. Ces expérimentations ont été réalisées grâce à l'outil TARGET et s'inscrivent

dans le cadre d'un cas d'étude consacré à la recherche de communications scientifiques publiées à la conférence internationale *World Wide Web*.

4 Plan du manuscrit

Ce manuscrit est composé de sept chapitres différents dont le contenu est brièvement présenté ci-après.

Chapitre 1 : Le premier chapitre de ce document présente un état de l'art sur la représentation des données du Web, leur évolution et leur exploitation dans le cadre de la recherche d'information. L'aspect représentation des données du Web inclut une présentation des différents types de données, des metadonnées et de la notion d'ontologie. Nous détaillons ensuite les approches existantes visant l'évolution des données du Web. Nous présentons les travaux existants traitant de l'exploitation des données du Web principalement en nous concentrant sur les techniques de recherche d'information pertinente sur le Web. Pour clore ce chapitre, nous positionnons nos travaux par rapport à ceux présentés dans ce chapitre.

Chapitre 2 : Le second chapitre est consacré à la présentation détaillée du modèle d'ontologie adaptative que nous proposons. Nous discutons les différents éléments qui constituent ce modèle. Cette étude comprend à la fois la conceptualisation, la modélisation ainsi que la représentation de ces éléments au niveau de l'ontologie.

Chapitre 3 : Le troisième chapitre traite du processus d'adaptation des ontologies adaptatives. Au cours de ce chapitre, nous présentons notre approche de l'adaptation à travers les différentes règles que nous avons définies et qui visent à automatiser le processus d'adaptation des ontologies aux évolutions du domaine qu'elles représentent.

Chapitre 4 : Le chapitre quatre présente une utilisation des ontologies adaptatives pour la représentation des données du Web dans le but d'optimiser la recherche de documents pertinents sur Internet. Nous présentons d'abord l'utilisation de ce types d'ontologies pour représenter le domaine de recherche, les documents du domaine de recherche ainsi que les différentes vues que les utilisateurs peuvent avoir sur ce domaine. Nous poursuivons par la présentation des structures de données que sont le WPGraphs et W^3Graphs construits suivant une ontologie adaptative et servant à la représentation enrichie du Web et de ses données.

Chapitre 5 : Le cinquième chapitre discute du langage de requêtes ASK ainsi que de la méthode d'enrichissement de requêtes basée sur les ontologies que nous proposons. Nous débutons par la définition de la syntaxe et de la sémantique de notre langage, puis nous discutons de l'utilisation des ontologies et en particulier des relations ontologiques pour l'enrichissement de requêtes pour le Web. Dans un troisième temps, nous montrons l'apport des ontologies adaptatives pour l'enrichissement des requêtes.

Chapitre 6 : Le chapitre six présente notre approche adaptative pour la recherche d'information pertinente sur le Web. Ce chapitre vise à intégrer les différents concepts présentés dans ce manuscrit dans un objectif d'optimisation des résultats lors d'une recherche d'information. Nous présentons comment les résultats pertinents sont extraits de nos structures de données ainsi que la façon dont ces résultats sont classés afin de répondre au mieux aux attentes des

utilisateurs.

Chapitre 7 : Le dernier chapitre est entièrement consacré à la validation expérimentale de l'approche TARGET présentée dans ce document. Pour se faire, nous présentons d'abord l'étude de cas sur laquelle reposent nos expérimentations. Ensuite, nous détaillons l'outil que nous avons conçu pour l'adaptation d'ontologie et pour la recherche de documents pertinents sur le Web. Nous finissons par la présentation du protocole de validation et des résultats expérimentaux obtenus grâce à notre outil.

Enfin, le mémoire s'achève sur une conclusion dans laquelle, d'une part nous résumons les travaux et les contributions détaillés dans cette thèse et, d'autre part, nous évoquons l'ensemble des perspectives que nos travaux peuvent donner principalement en recherche d'information et dans le domaine de l'ingénierie des connaissances.

Chapitre 1

Etat de l'art et de la pratique

Sommaire

Les différentes problématiques présentées dans la section 2 de l'introduction de ce mémoire, nous obligent à étudier les travaux existants dans plusieurs domaines afin de bien faire ressortir nos contributions pour améliorer la recherche d'information sur le Web.

Le premier de ces domaines fait référence aux techniques de représentation des données du Web. Ce point est fondamental car il est à la base du processus de recherche d'information. Une représentation adéquate des données du Web peut améliorer à la fois l'efficacité du processus de recherche et la pertinence des pages Web retournées. Nous détaillons dans cette partie les approches existantes pour la représentation du Web et de son contenu. Une attention particulière est portée à la notion d'ontologie, à sa définition, aux méthodologies de construction, aux langages ontologiques, aux outils supports et à leur utilisation pour la représentation des métadonnées.

Nous nous intéressons ensuite à la gestion de l'évolution des données du Web ou plus généralement des ressources du Web. Dans cette partie, nous débattons également de l'évolution des métadonnées puis nous nous attardons plus longuement sur les méthodes, techniques et outils du Web Sémantique permettant la gestion de l'évolution des ontologies. D'une part, nous expliquons les besoins de changement, d'autre part les types d'évolution qui peuvent affecter les ontologies. Pour finir, nous expliquons comment le problème de l'évolution des données est traité dans des domaines connexes comme, par exemple, celui des bases de données.

Le troisième et dernier point que nous nous proposons d'étudier dans cet état de l'art se rapporte à l'exploitation des données du Web. Dans cette dernière partie de notre étude, nous analysons les approches existantes pour la recherche d'information sur le Web en nous concentrant surtout sur les travaux portant sur la recherche intelligente de documents sur la toile. Nous évoquons les méthodes de classement des résultats d'une recherche et introduisons les différents langages de requêtes et les techniques d'expansion de requêtes permettant d'améliorer la recherche d'information. Nous présentons les différentes familles d'outils pour la recherche d'information sur le Web et pour finir nous décrivons d'autres techniques existantes d'exploitation des données et métadonnées du Web.

1.1 Représentation des données du Web

La représentation des données du Web joue un rôle important dans le processus de recherche d'information en facilitant notamment leur accès. En outre, la réalisation de la vision du Web Sémantique nécessite l'utilisation de données particulières, les métadonnées, extraites d'ontologies pour décrire le contenu des pages Web toujours dans le but de faciliter la recherche d'information. Cette partie est donc consacrée à l'étude des différents modes de représentation des données et métadonnées du Web.

1.1.1 Données et métadonnées du Web

Les données

Comme évoqué dans l'introduction de ce mémoire, l'évolution du Web affecte principalement ses données. Elles se retrouvent sur le Web principalement sous forme de texte, de ressources multimédias (i.e. son, image, vidéo) ou enfin d'éléments logiciels comme les applets Java par exemple. La représentation de ces données est généralement faite de manière standardisée grâce à différents langages.

Le langage dédié à la représentation des données du Web le plus répandu est HTML [1] (HyperText Markup Language). HTML est un langage de description de document qui est basé sur l'utilisation de balises. Il permet de structurer le contenu des pages Web, d'implanter des liens hypertextes dans le contenu des pages permettant ainsi d'y accéder, et aussi d'inclure des formulaires de saisie, des ressources multimédias dont des images, et des éléments programmables tels que des applets. Les 91 balises proposées par la version 4 du langage permettent d'implémenter les fonctionnalités suivantes au niveau d'une page Web :

- La définition de la **structure générale d'un document** : Au plus haut niveau, un document HTML est séparé entre un en-tête et un corps. L'en-tête contient les informations sur le document, notamment son titre. Le corps contient ce qui est affiché à l'écran.
- La définition d'informations relatives à la **langue** dans laquelle est écrit le contenu du document.
- **Le marquage sémantique** qui permet de différencier des contenus spécifiques tels que les citations d'œuvres externes, les extraits de code informatique, les passages en emphase et les abréviations.
- La définition de **listes**. HTML différencie des listes non ordonnées et des listes ordonnées, selon que l'ordre formel du contenu dans le code est en soi ou non une information.

1. http ://www.w3.org/TR/html4

- La définition de **tables**. Cette fonctionnalité a été développée pour permettre la présentation de données tabulaires mais est également exploitée pour ses puissantes capacités de mise en page.
- La définition des **hyperliens**.
- **L'inclusion d'images, d'applets et d'objets divers**.
- La définition **d'éléments de regroupement**. Ne conférant pas de signification au contenu qu'ils balisent, ces éléments génériques permettent d'appliquer des styles de présentation, de réaliser des traitements via des scripts ou toute autre opération nécessitant d'isoler une partie du contenu.
- La définition du **style de la présentation**. Les styles sont définis dans le document ou proviennent de feuilles de style en cascade (CSS) externes.
- **Marquage de présentation du texte**. Développé avant la généralisation de CSS pour fournir rapidement des fonctionnalités aux graphistes
- La définition de **cadres**. Cette fonctionnalité permet l'affichage de plusieurs documents HTML dans une même fenêtre de navigateur.
- La définition de **formulaires** pour l'insertion interactive de données.
- La définition de **scripts** permettant d'associer des morceaux de programmes aux actions des utilisateurs sur le document.

De plus, le langage définit un ensemble d'attributs qui permettent de préciser les propriétés des éléments HTML. Ces attributs s'appliquent différemment selon les balises considérées. Certains, comme les attributs génériques, les attributs d'internationalisation ou les gestionnaires d'événements, s'appliquent à tous les éléments. D'autres, comme les attributs permettant de définir les dimensions d'un objet graphique, sont propres à un élément unique. La plupart d'entre eux sont facultatifs.

La structure d'un document HTML est rarement «bien formée» (au sens XML [1] du terme). Certaines balises, comme <body> ou <head>, peuvent être utilisées de manière peu rigoureuse ou même tout simplement être omises lors de la construction d'un document HTML. Ceci engendre des problèmes d'interprétation du langage par les outils du Web réduisant ainsi leur interopérabilité.

Pour pallier à ce problème, d'autres standards pour la représentation des données du Web ont récemment été proposés. Parmi eux nous retrouvons notamment XHTML [2] (eXtended Hypertext Markup Language). Ce langage correspond à une évolution de HTML avec pour base l'utilisation de XML. Pour l'instant XHTML n'introduit aucune nouvelle fonctionnalité par rapport à celles offertes par HTML. Toutefois, la syntaxe beaucoup plus rigoureuse de XHTML permet en partie de résoudre les problèmes de structure des documents écrits en HTML. L'utilisation de ce type de langages s'appuyant sur la syntaxe de XML a notamment permit le développement de recommandations comme le DOM [3] (Document Object Model). DOM est une interface de programmation d'applications et s'appuie sur une représentation sous forme d'arbre de la structure d'un document et de ses éléments pour permettre aux programmes informatiques de manipuler de façon plus efficace la structure, le contenu ou le style d'un document.

Cependant, les données du Web ne sont pas toutes «visibles». Nous entendons par là accessibles directement par une URL. Une grande partie se retrouve localisée dans des bases de

1. http ://www.w3.org/XML/
2. http ://www.w3.org/TR/xhtml1/
3. http ://www.w3.org/DOM/

données et est utilisée par des programmes informatique (scripts) pour la génération dynamique de pages Web. Ce type de procédé est très largement utilisé par des sites Web de vente en ligne comme Amazon. L'utilisation de bases de données relationnelles permet une gestion plus efficace des grandes quantités de données dont dispose ce type d'applications.

Si les langages présentés ci-dessus permettent d'une part, de représenter de manière structurée les données textuelles du Web, et d'autre part, d'insérer des ressources multimédias ou logicielles dans les documents Web, le contenu des vidéos, sons, images, etc. ainsi que la sémantique de l'information textuelle n'est pas directement prise en compte dans les fonctionnalités offertes par ces langages. La réalisation de la vision du Web Sémantique, rend nécessaire l'introduction de moyens pour décrire des ressources existantes ou enrichir en sémantique des données afin de les rendre compréhensibles par les machines.

Les métadonnées

La notion de métadonnée, comme son étymologie l'indique, fait référence à un type particulier de données pouvant décrire d'autres données. Elles ont fait leur apparition dans le cadre du Web dès 1994 lors de la création du World Wide Web Consortium. Ce type de donnée a été adopté dans le but d'assurer l'interopérabilité entre les ressources du Web [Lassila, 1998]. A titre d'exemple, le nom d'un fichier informatique, sa taille, son extension sont des métadonnées décrivant une ressource, en l'occurrence ici un fichier.

Les métadonnées sont en général regroupées en trois grandes catégories [NISO, 2004] :

- Les métadonnées dites *descriptives* pour la description de ressources à des fins d'identification.
- Les métadonnées dites *structurelles* qui sont principalement utilisées pour décrire la structure des données. Elles facilitent la navigation et la présentation des ressources électroniques.
- Les métadonnées *administratives* dont le rôle est d'assister les utilisateurs dans la gestion des données.

En outre, les métadonnées se distinguent par :

- Le contenu qu'elles décrivent. Les métadonnées peuvent décrire une ressource, en donnant par exemple le nom et la taille d'un fichier, ou le contenu de la ressource. Par exemple, pour une vidéo, ce qu'elle montre.
- La possibilité qu'elles ont d'évoluer [Guelfi et al., 2008]. Les métadonnées peuvent ou non changer selon qu'elles décrivent des données statiques ou dynamiques.
- Le niveau logique auquel elles se rapportent. On peut distinguer trois niveaux, le niveau le plus bas contenant la donnée brute, le second niveau contenant la description de la donnée brute et enfin le troisième niveau permettant d'utiliser les métadonnées pour des tâches de raisonnement.

Les métadonnées du Web peuvent être utilisées dans les langages de marquage comme HTML ou XHTML avec des balises dédiées (par exemple la balise <META> de HTML). Mais, dans bons nombres de cas, elles sont décrites en RDF [1] (Ressource Description Framework). Constituant le langage de base du Web Sémantique, RDF est un modèle de graphe destiné à décrire de façon formelle les ressources et les métadonnées du Web, afin de permettre le traitement automatique

1. http ://www.w3.org/RDF/

de telles descriptions. Un document structuré en RDF est représenté par un ensemble de triplets $\{Sujet, Predicat, Objet\}$. Le $Sujet$ représente la ressource à décrire, le $Predicat$ représente une propriété applicable à cette ressource et enfin l'$Objet$ est une donnée ou une autre ressource [1]. Le $Sujet$, et l'$Objet$ dans le cas où c'est une ressource, peuvent être identifiés par une adresse de type URI, un littéral ou être des nœuds anonymes. Le prédicat, quant à lui, est nécessairement identifié par une URI. Un document RDF ainsi formé correspond à un multi-graphe orienté étiqueté. Chaque triplet correspond à un arc orienté dont le label est le $Predicat$, le sommet source est le $Sujet$ et le sommet cible représente l'$Objet$.

La sémantique d'un document RDF peut être exprimée en théorie des ensembles et en théorie des modèles en se donnant des contraintes sur le monde qui peuvent être décrites en RDF. Le langage hérite alors de la généricité et de l'universalité de la notion d'ensemble. Cette sémantique peut être aussi traduite en formules de logique du premier ordre, positive, conjonctive et existentielle.

$$\{Sujet, Predicat, Objet\} \Longleftrightarrow Predicat(Sujet, Objet)$$

Ce qui est équivalent à :

$$\exists Objet, \exists Sujet, Predicat(Sujet, Objet)$$

RDF est utilisé dans de nombreuses applications. Les plus connues d'entre elles sont le Dublin Core pour le classement bibliographique, les flux de données RSS, le navigateur Mozilla pour le stockage des marques-pages et enfin l'encyclopédie en ligne Wikipédia dont une partie de son contenu est rendu disponible sous la forme de triplets RDF.

Exemple : Le graphe RDF de la figure 1.1 contient les assertions suivantes. La ressource `se2c.uni.lu/users/CP` a pour créateur un objet anonyme qui a pour nom le littéral Cédric Pruski et pour e-mail le littéral cedric.pruski@uni.lu

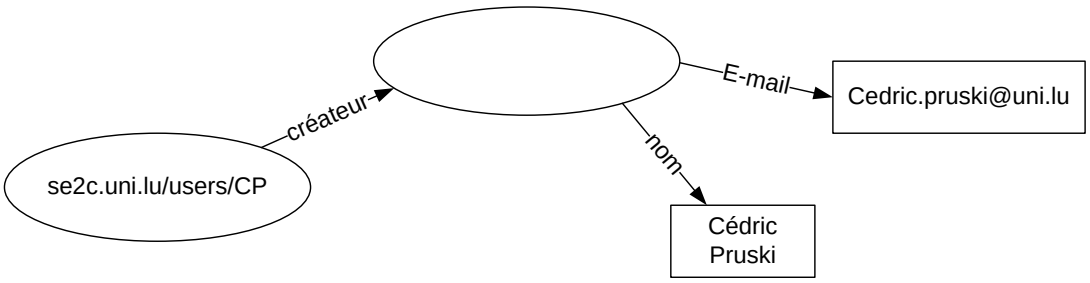


FIGURE 1.1 – Un exemple de graphe RDF

La représentation sous forme de graphe d'un document RDF peut être facilement traduite en XML. C'est pour cette raison que le W3C préconise l'utilisation de XML pour représenter les documents RDF. Ainsi, une représentation XML du graphe de la figure 1.1 est :

```
<description about="se2c.uni.lu/users/CP">
   <createur>
```

1. Une ressource du World Wide Web est un élément constitutif de base de l'architecture du World Wide Web. Le terme désigne le référent d'une URI

```
      <description>
         <Nom>Cédric Pruski</Nom>
         <Email>cedric.pruski@uni.lu</Email>
      </description>
   </createur>
</description>
```

Dans le cadre du Web Sémantique, les métadonnées organisées en hiérarchie sont appelées ontologies. Elles sont utilisées pour décrire les ressources du Web afin d'améliorer la recherche d'information en rendant l'information qu'elles décrivent compréhensible par les machines. C'est pourquoi ces métadonnées doivent avoir une sémantique bien définie. Si RDF permet de décrire les métadonnées, le langage n'est toutefois pas suffisamment expressif pour en définir toute la sémantique, c'est pour cette raison que le Web Sémantique a nécessité le développement de langages plus riches.

1.1.2 Les ontologies

Introduite dès l'antiquité par certains philosophes comme Aristote, *l'ontologie* est l'étude de l'être en tant qu'être ou plus précisément la branche de la métaphysique qui s'intéresse aux propriétés générales de ce qui existe. Faisant référence à la notion *d'existence*, c'est naturellement que certains domaines de l'informatique comme l'Intelligence Artificielle ont repris la notion d'ontologie en guise de modèle des connaissances d'un domaine.

Définition

Depuis l'apparition des systèmes à base de connaissance, plusieurs définitions ont été proposées pour définir la notion d'ontologie en informatique ([Gruber, 1993, Guarino, 1998]). Dans un soucis de cohérence, nous garderons celle donnée par Tom Gruber [Gruber, 1993] pour lequel,

> *«Une ontologie est la spécification explicite et formelle d'une conceptualisation faisant l'objet d'un consensus.»*

Cette définition d'une ontologie est la plus largement acceptée par les communautés de l'ingénierie des connaissances et du Web Sémantique. Selon elle, une ontologie est la *conceptualisation* d'un domaine, ce qui sous-entend un choix dans la façon de décrire ce domaine mais c'est aussi une *spécification formelle* de cette conceptualisation. Dans sa définition, Gruber explique que les connaissances contenues dans une ontologie peuvent être formalisées à l'aide de 5 éléments distincts :

- Les **concepts** sont les composants de base de l'ontologie. Un concept se définit comme l'ensemble des propriétés vérifiées par un objet indépendamment des variations qu'il peut subir selon les différents contextes où il se rencontre.
- Les **relations** permettent de définir la manière dont les concepts de l'ontologie s'articulent les uns par rapport aux autres.
- Les **fonctions** représentent un type de relation particulière. Une fonction relie un ensemble de concept à un seul concept.
- Les **axiomes** sont utilisés dans l'ontologie pour définir les faits du domaine qui sont toujours vrais. L'utilisation d'axiome dans l'ontologie permet de contraindre les informations de l'ontologie, de vérifier leur exactitude ou de déduire de nouvelles informations.
- Les **instances** de concept définissent les objet précis du monde réel.

Kalfoglou propose une définition plus formelle [Kalfoglou and Schorlemmer, 2003] dans laquelle une ontologie est vue comme un couple (S, A). S représente la signature (ou vocabulaire) de l'ontologie qui peut être modélisée à l'aide d'une structure mathématique comme un treillis de concept. A est un ensemble d'axiomes spécifiant l'interprétation du vocabulaire par rapport à un domaine. Pour être plus cohérent avec la définition de Gruber, la signature S de l'ontologie peut se décomposer en trois sous-ensembles (pas nécessairement disjoints). Le premier d'entre eux, noté C, représente l'ensemble des concepts. Le second, R, dénote l'ensemble des relations. Enfin, I est l'ensemble des instances. En conclusion, une ontologie est définie par le quadruplet (C, R, I, A).

Méthodologie pour la construction d'ontologies

La méthodologie suivant laquelle est construite une ontologie est fondamentale pour toutes les applications à base d'ontologies. Quelques méthodologies ont été proposées principalement par la communauté Ingénierie des connaissances. Toutes ces méthodes s'accordent sur un certains nombres de points :

- La définition du type d'ontologie que l'on veut construire. Plusieurs catégories d'ontologies ont été répertoriées. L'ontologie du domaine, l'ontologie générique qui représente les concepts les plus abstraits, l'ontologie d'une méthode de résolution de problème où le rôle joué par chaque concept dans le raisonnement est rendu explicite, l'ontologie d'application qui combine les spécificités d'une ontologie du domaine et d'une ontologie de méthode, et enfin l'ontologie de représentation qui repère et organise les primitives de la théorie logique permettant de représenter l'ontologie.
- L'identification des concepts du domaine et de leurs propriétés.
- La hiérarchisation de ces concepts à travers l'utilisation de la relation d'hypéronymie afin de définir la structure de l'ontologie.
- L'identification des autres relations du domaine permettant de relier les concepts entre eux et ainsi de compléter leur définition.

Les méthodes de construction d'ontologies se basent sur différentes approches. Certaines d'entre elles s'appuient sur un corpus de documents [Zweigenbaum et al., 1995], d'autres, comme celle proposée par Gruber [Gruber, 1993] sont un recueil de bonnes pratiques méthodologiques de construction d'ontologies.

L'utilisation de corpus de documents pour la construction d'ontologie a été proposée par Bouaud et al. [Bouaud et al., 1994, Charlet, 2002]. La méthodologie définie dans ces travaux repose sur les idées développées par Bachimont [Bachimont et al., 2002]. Les 4 étapes de la méthodologie sont :

- **La constitution du corpus et son analyse**. L'utilisation du Web permet la constitution d'un corpus de documents d'un domaine particulier de par la grande quantité de documents constituant la toile. Le corpus devient la source privilégiée qui permettra de caractériser les notions utiles à la modélisation ontologique et le contenu sémantique qui leur correspond. L'analyse du corpus, constitué de documents rédigés en langue naturelle, nécessite l'utilisation d'outils terminologiques qui reposent sur la recherche de formes syntaxiques particulières manifestant les notions recherchées, comme des syntagmes nominaux pour des candidats termes, des relations syntaxiques marqueurs de relations sémantiques, ou des proximités d'usage pour des regroupements de notions.
- **La normalisation sémantique**. Cette étape consiste en quelque sorte en la désambiguïsation des termes retenus lors de l'analyse du corpus de documents afin de leur attribuer un

sens bien défini ne permettant qu'une seule interprétation possible par rapport au domaine d'étude.
- **L'engagement ontologique** consiste à formaliser les notions sémantiques issues des étapes précédentes. L'organisation des concepts résultante aura la structure d'un treillis de concepts.
- **L'opérationalisation** consiste en la représentation de l'ontologie dans un langage de représentation des connaissances.

Le système TERMINAE développé par Aussenac-Gilles et al. [Aussenac-Gilles et al., 2000], s'inscrit dans un paradigme identique à l'approche présentée par Zweigenbaum et al. Aussenac-Gilles et al. mettent toutefois l'accent sur le dépouillement des corpus et l'étude linguistique, en focalisant notamment sur le repérage des relations. De plus, la question de l'opérationalisation dans une logique de description et son influence sur la modélisation a été tout spécifiquement étudiée.

L'approche proposée par Buitelaar et al. [Buitelaar et al., 2005] s'inspire d'une méthodologie similaire dans les grandes lignes à la méthode des ontologies différentielles proposée par Bachimont. Néanmoins, l'approche de Buitelaar et al. se différencie notamment dans le nombre d'étapes du processus de construction. Si l'analyse de corpus permet toujours de retrouver les termes candidats pour devenir des concepts du domaine, les auteurs s'appuient également sur le corpus pour découvrir un ensemble de termes synonymes. Ensuite, interviennent les phases de normalisation sémantique et d'engagement ontologique, puis la méthodologie propose d'utiliser le corpus de documents afin de trouver d'une part les différentes relations existantes entre les nouveaux concepts du domaine, comme le font Aussenac-Gilles et al. [Aussenac-Gilles et al., 2000], d'autre part un ensemble de propriétés s'appliquant aux concepts. Enfin, la dernière étape d'opérationalisation a lieu.

Le système OntoLearn développé par Velardi et al. [Velardi et al., 2006] combine l'utilisation de techniques à base de statistiques sur les composantes du corpus et des outils terminologiques tels que WordNet [Fellbaum, 1998] afin de désambiguïser les termes issus de l'analyse du corpus. L'utilisation de tels outils renforce l'automatisation du processus de construction d'ontologie.

D'autres méthodologies ont également été proposées pour la construction d'ontologies. Ces autres approches sont des recueils de règles décrivant de manière générale la construction d'ontologies.

La méthode de Uschold et King [Uschold and King, 1995] est la plus ancienne méthodologie connue. Cette méthode tient en 4 étapes. La première d'entre elles consiste en l'identification du domaine que doit représenter l'ontologie et aussi l'utilisation que l'on va faire de l'ontologie une fois construite. La deuxième phase concerne la construction de l'ontologie proprement dite suivant les principes évoqués précédemment. L'évaluation de l'ontologie et enfin sa documentation sont les troisième et quatrième étapes du processus. Les étapes de cette méthodologie ont servi de base à toutes celles proposées par la suite.

METHONTOLOGY a été développée par Fernandez et al. [Fernandez et al., 1997] dans laquelle les auteurs insistent sur la phase d'analyse des besoins pour la construction d'une ontologie c'est à dire dans quel but l'ontologie doit-elle être construite. La méthodologie prône également l'utilisation de plusieurs techniques dont l'analyse de corpus et des séances de brainstorming entre experts du domaine pour l'acquisition des connaissances à représenter dans l'ontologie. METHONTOLOGY introduit également une phase de maintenance dans le cycle de vie d'une ontologie. Durant cette phase, l'ontologie est amenée à évoluer en fonction des changements du domaine.

Le processus de création d'ontologie 101 développé à Stanford par Noy et McGuinness [Noy and McGuinness, 2001] se déroule suivant sept étapes. Comme pour les deux méthodes introduites précédemment, la première étape consiste en l'identification du domaine et des objectifs auxquels doit répondre l'ontologie. L'étape suivante nouvellement introduite se propose d'étudier la possibilité de réutiliser une ontologie existante. Puis viennent successivement les phases d'identification des concepts du domaine, de hiérarchisation de ces concepts, et de définition de leurs propriétés. Le processus s'achève avec la phase de création des instances de concepts.

Spyns et al. proposent la méthodologie DOGMA [Jarrar and Meersman, 2002] pour la construction d'ontologies formelles. DOGMA est directement inspirée de méthodologies développées dans les domaines des bases de données et du génie logiciel et de la méthode Rational Unified Process (RUP) en particulier. Les ontologies résultantes se veulent davantage utilisables et réutilisables par les applications. La méthodologie se décompose en trois étapes :

1. La définition de la base ontologique par un expert d'un domaine. Cette étape consiste à définir l'ensemble des faits sur un domaine et de les formaliser sous la forme de quintuplets appelés lexons.
2. La définition de la couche intermédiaire entre les applications et la base ontologique. Cette couche est composée d'un ensemble d'engagements ontologiques chacun d'entre eux est formé d'un ensemble de règle décrit dans une syntaxe particulière. Les règles contraignent les axiomes de la base ontologique à des faits du monde réel. La base ontologique et la couche intermédiaire forment l'ontologie.
3. La représentation des axiomes de la base ontologique et des règles de la couche intermédiaire dans un langage ontologique compréhensible par les applications pour lesquelles l'ontologie est construite.

Plus récemment, Kotis et Vouros ont développé la méthode HCOME [Kotis and Vouros, 2006]. Celle-ci se différencie des autres approches existantes principalement dans le fait que les ontologies sont construites puis gérées suivant des décisions prises en commun par un consortium d'experts et non plus par une seule personne comme c'est le cas dans les autres méthodologies présentées jusque là.

Comme le montre l'étude des méthodologies pour la création d'ontologie, les différentes étapes de construction d'une ontologie sont sensiblement les mêmes d'une méthode à l'autre. Les phases de maintenance d'une ontologie ou plus généralement d'évolution d'ontologie ne figurent que parmi les méthodologies les plus récentes. Même si les méthodologies proposées se veulent indépendantes de tous langages, ceux-ci tiennent une place importante, de part leur expressivité, dans certaines étapes du processus de construction notamment dans les phases de représentation des concepts, de leurs propriétés, des relations du domaine et des instances de concepts.

Langages du Web Sémantique pour la représentation d'ontologies

Les langages pour la représentation d'ontologies ou plus généralement pour la représentation des connaissances sont légions. Parmi les plus connus, on peut citer les langages de frames comme les graphes conceptuels [Sowa, 1984], ou les langages à base de logiques de description [Baader et al., 2003]. Une étude détaillée de ces langages peut être trouvée dans [Pruski, 2006]. Dans ce mémoire, nous nous focalisons sur ceux définis dans le cadre du Web Sémantique, en particulier RDF Schema et OWL.

Ressource Description Framework Schema (RDF/S)

Le langage RDF, présenté à la section 1.1.1, sert de fondation au Web Sémantique. On peut considérer les propriétés RDF comme des attributs des ressources et, en ce sens, elles peuvent correspondre aux couples attribut-valeur traditionnels. Les propriétés RDF représentent également des relations entre les ressources. Toutefois, RDF ne fournit aucun mécanisme pour décrire ces propriétés ni pour décrire les relations entre ces propriétés et d'autres ressources. La définition du langage RDF/S est un premier pas vers cet objectif. Il fournit un ensemble d'éléments de base pour la définition d'ontologies destinés à structurer des ressources décrites en RDF. RDF/S permet notamment de typer le *Sujet* et l'*Objet* des triplets RDF avec l'élément rdfs :class. Le langage permet de définir une hiérarchie entre les classes avec l'utilisation de l'élément rdfs :subClassOf. RDF/S permet de préciser la notion de propriété définie par RDF à travers les notions de domaine (rdfs :domain) et co-domaine (rdfs :range). D'autres constructeurs permettent de définir des conteneurs, des collections ou de documenter les éléments décrits.

RDF Schema a également été défini dans le but d'offrir des mécanismes de raisonnement sur les descriptions des ressources du Web. Le langage intègre la notion d'héritage, la réflexivité et la transitivité des propriétés. Il permet de décider du type d'un objet impliqué dans une propriété grâce aux notions de domaine et co-domaine.

Si les éléments offerts par RDF/S constituent la base d'un langage d'ontologie, ce dernier ne permet toutefois pas :

- de représenter le fait que deux classes sont disjointes.
- de représenter un ensemble de classes (ou d'individus) équivalentes.
- de définir une classe comme une intersection ou une union de plusieurs classes.
- de représenter des restrictions de cardinalités portant sur les propriétés.
- d'appliquer des restrictions sur le domaine d'une propriété (i.e. une propriété ne serrait valable que sur un ensemble restreint de classes).
- d'établir des caractéristiques telles que l'unicité, ou l'inverse d'une propriété.

Web Ontology Language (OWL)

OWL [McGuinness and van Harmelen, 2004] est sans aucun doute la famille de langages la plus aboutie développée par la communauté du Web Sémantique pour la représentation d'ontologies. OWL est construit sur les bases de RDF et RDF/S et offre en plus des solutions pour répondre aux manques de RDF/S évoqués au paragraphe précédent. OWL regroupe trois langages, chacun d'entre eux offre un niveau d'expressivité différent. La sémantique de ces langages est donnée en logique de description (DL). Ces trois langages sont :

- **OWL Lite**, le moins expressif de ces langages. Il offre les caractéristiques minimales pour construire une hiérarchisation simple de concepts et permet d'exprimer des contraintes de cardinalités mais seulement pour les valeurs 0 et 1.
- **OWL DL** qui a été conçu pour fournir le maximum d'expressivité tout en garantissant la complétude du raisonnement (i.e. tous les raisonnements se terminent) et la décidabilité (i.e. le raisonnement se termine en un temps fini). Il existe néanmoins certaines restrictions en OWL DL (par exemple, bien qu'une classe puisse être une sous-classe de plusieurs classes, elle ne peut pas être une instance d'une autre).
- **OWL Full** qui possède une sémantique différente de celle de OWL Lite et de OWL DL mais conserve une certaine compatibilité avec RDF/S. La principale distinction avec OWL DL réside dans le fait qu'une classe décrite en OWL Full peut être vue comme une collection d'individus ou comme un individu à part entière. OWL Full permet également d'étendre

le vocabulaire par défaut de RDF et OWL. Il est alors évident que ni la complétude ni la décidabilité du raisonnement effectué sur des ontologies décrites en OWL Full ne peuvent être garanties.

Il existe une dépendance de nature hiérarchique entre ces trois sous-langages : toute ontologie OWL Lite légale est également une ontologie OWL DL légale, et toute ontologie OWL DL légale est également une ontologie OWL Full légale mais la réciproque est fausse (i.e. une ontologie OWL Full légale n'est pas une ontologie OWL DL légale et une ontologie OWL DL légale n'est pas une ontologie OWL Lite légale). Les principales primitives du langage OWL, leur syntaxe abstraite, leur sémantique exprimée en logique de description et en logique du premier ordre sont décrites dans le tableau 1.1 ci-après. Dans ce tableau, C représente un concept, O une propriété servant à relier des individus entre eux, D est une propriété pouvant relier des individus à des types de données comme des chaînes de caractères ou des nombres entiers, I représente un individu, R dénote un ensemble de données, V représente une valeur et i,j, et n représentent des entiers naturels. Soit une *interprétation* $\mathcal{I} = (\Delta^{\mathcal{I}}, .^{\mathcal{I}})$ où :

- $\Delta^{\mathcal{I}}$ est le domaine d'interprétation (i.e. un ensemble non vide d'éléments),
- $.^{\mathcal{I}}$ est une fonction qui associe :
  - chaque concept à un sous-ensemble de $\Delta^{\mathcal{I}}$
  - chaque propriété à un sous-ensemble de $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$

nous avons, $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}, O^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}, D^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}, I^{\mathcal{I}} \in \Delta^{\mathcal{I}}, R^R \subseteq \Delta^{\mathcal{I}}_R$ et $V^{\mathcal{I}} = V^R$. Dans la plupart des applications, R,C,O,D et I sont des références URI alors que V est un littéral au sens RDF du terme. Les primitives que nous détaillons ci-après sont choisies afin que le lecteur puisse mieux comprendre certains à venir. Le détail des autres primitives constituant la famille de langage OWL se trouve en annexe.

Elément OWL	**Syntaxe abstraite**	**DL**	**Sémantique**
owl :Class	Class(A partial $C_1 \ldots C_n$)	$A \sqsubseteq C_1 \sqcap \ldots \sqcap C_n$	$A^{\mathcal{I}} \subseteq C_1^{\mathcal{I}} \cap \ldots \cap C_n^{\mathcal{I}}$
	Class(A complete $C_1 \ldots C_n$)	$A = C_1 \sqcap \ldots \sqcap C_n$	$A^{\mathcal{I}} = C_1^{\mathcal{I}} \cap \ldots \cap C_n^{\mathcal{I}}$
rdfs :subClassOf	SubClassOf(C_1 C_2)	$C_1 \sqsubseteq C_2$	$C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$
owl :equivalentClass	EquivalentClasses($C_1 \ldots C_n$)	$C_1 = \ldots = C_n$	$C_1^{\mathcal{I}} = \ldots = C_n^{\mathcal{I}}$
owl :ObjectProperty	ObjProp(O super(O_1)...super(O_n)	$O \sqsubseteq O_i$	$O^{\mathcal{I}} \subseteq O_i^{\mathcal{I}}$
	domain(C_1)...domain(C_m)	$\geq 1 O \sqsubseteq C_i$	$O^{\mathcal{I}} \subseteq C_i^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
	range(C_1)...range(C_l)	$\top \sqsubseteq \forall O.C_i$	$O^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times C_i^{\mathcal{I}}$
	[InverseOf(O_0)]	$O = (^-O_0)$	$O^{\mathcal{I}} = (O_0^{\mathcal{I}})^-$
	[Symmetric]	$O = (^-O)$	$O^{\mathcal{I}} = (O^{\mathcal{I}})^-$
	[Functional]	$\top \sqsubseteq \leq 1\ O$	$O^{\mathcal{I}}$ is functional
	[InverseFunctional]	$\top \sqsubseteq \leq 1\ O^-$	$(O^{\mathcal{I}})^-$ is functional
	[Transitive]	$Tr(O)$	$O^{\mathcal{I}} = (O^{\mathcal{I}})^+$
owl :DatatypeProperty	DatProp(D super(D_1)...super(D_n)	$D \sqsubseteq D_i$	$D^{\mathcal{I}} \subseteq D_i^{\mathcal{I}}$
	domain(C_1)...domain(C_m)	$\geq 1\ D \sqsubseteq C_i$	$D^{\mathcal{I}} \subseteq D_i^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
	range(C_1)...range(C_l)	$\top \sqsubseteq \forall D.C_i$	$D^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times C_i^{\mathcal{I}}$
	[Functional]	$\top \sqsubseteq \leq 1\ D$	$D^{\mathcal{I}}$ is functional

owl :minCardinality	$restriction(O\ minCardinality(n))$	$\geq n\ O$	$(\geq\ n\ O)^{\mathcal{I}} = \{x \mid \sharp(\{y. < x, y >\in O^{\mathcal{I}}\}) \geq n\}$
	$restriction(D\ minCardinality(n))$	$\geq n\ D$	$(\geq\ n\ D)^{\mathcal{I}} = \{x \mid \sharp(\{y. < x, y >\in D^{\mathcal{I}}\}) \geq n\}$
owl :maxCardinality	$restriction(O\ maxCardinality(n))$	$\leq n\ O$	$(\leq\ n\ O)^{\mathcal{I}} = \{x \mid \sharp(\{y. < x, y >\in O^{\mathcal{I}}\}) \leq n\}$
	$restriction(D\ maxCardinality(n))$	$\leq n\ D$	$(\leq\ n\ D)^{\mathcal{I}} = \{x \mid \sharp(\{y. < x, y >\in D^{\mathcal{I}}\}) \leq n\}$

TABLE 1.1: Syntaxe et sémantique des principales primitives de la famille de langage OWL

D'un point de vue conceptuel, les primitives contenues dans le tableau 1.1 se détaillent comme suit :

- owl :Class : Une classe définit un groupe d'individus mis ensemble parce qu'ils partagent certaines propriétés.
- rdfs :subClassOf : On peut créer des hiérarchies de classes au moyen d'une ou de plusieurs déclarations selon lesquelles une classe est une sous-classe d'une autre.
- owl :equivalentClass : On peut déclarer deux classes équivalentes. Les classes équivalentes ont les mêmes instances. On peut créer deux classes synonymes à l'aide d'une égalité. Par exemple, on peut déclarer «Voiture» comme étant une classe équivalente de «Automobile». Un moteur de raisonnement pourra en déduire que tout individu instance de «Voiture» est aussi une instance de «Automobile», et vice versa.
- rdf :Property : Des propriétés peuvent aussi servir à déclarer des relations entre individus, ou depuis des individus vers des valeurs de données. On prendra les propriétés suivantes pour exemple : «aPourEnfant», «estApparentéÀ», «estFrèreDe» et «estAgéDe». Les trois premières peuvent servir à relier l'instance d'une classe «Personne» (ce sont donc des occurrences de la propriété ObjectProperty), et la dernière («estAgéDe») à relier une instance de la classe «Personne» à une instance du type de données Integer (c'est donc une occurrence de la propriété DatatypeProperty).
- owl :minCardinality : La cardinalité se déclare sur une propriété par rapport à une classe particulière. Si on déclare une restriction minCardinality de valeur 1 sur une propriété par rapport à une classe, alors toute instance de cette classe sera reliée par cette propriété à au moins un individu. Cette restriction est une autre façon de dire que la propriété doit obligatoirement avoir une valeur pour toutes les instances de la classe. Par exemple, la classe «Personne» n'aurait aucune restriction de cardinalité minimum déclarée sur une propriété «aPourDescendant» dans la mesure où les personnes n'ont pas toutes une descendance. Inversement, la classe «Parent» aurait une cardinalité minimum de 1 sur la propriété «aPourDescendant». Dans OWL Lite, les seules cardinalités minimum admises sont 0 et 1.

– owl :maxCardinality : Si on déclare une restriction maxCardinality de valeur 1 sur une propriété par rapport à une classe, alors toute instance de cette classe sera reliée par cette propriété à un individu au plus.

La famille de langage OWL propose également un ensemble de constructeurs n'ayant aucune sémantique particulière. Ces primitives ont un rôle de documentation et sont utilisées pour compléter la description des éléments de l'ontologie. La plupart de ces primitives concerne l'évolution d'ontologies. Une description est donnée dans le tableau 1.2

Primitives OWL	**Utilisation**
owl :Ontology	Information contenue dans l'en-tête de l'ontologie. Elle permet de collecter des métadonnées à propos de l'ontologie
owl :Import	Element qui permet d'importer des éléments d'une autre ontologie
owl :OntologyProperty	Elément qui permet de mettre en relation plusieurs ontologies
owl :AnnotionProperty	Elément qui permet de rattacher des informations supplémentaires aux éléments de l'ontologie
owl :versionInfo	Element donnant la version de l'ontologie
owl :priorVersion	Element permettant de définir la version précédente de l'ontologie décrite
owl :backwardCompatibleWith	Element qui identifie l'ontologie indiquée comme une version précédente de l'ontologie courante, en indiquant en outre que cette dernière est rétrocompatible
owl :incompatibleWith	Element qui identifie l'ontologie courante comme une version ultérieure mais non compatible de celle indiquée
owl :DeprecatedClass owl :DeprecatedProperty	Element indiquant qu'une caractéristique particulière, quoique conservée pour des raisons de rétrocompatibilité, est susceptible de disparaître dans le futur

TABLE 1.2: Autres primitives du langage

Outils pour l'ingénierie d'ontologies

Les méthodologies présentées précédemment sont en règle générale supportées par des outils. L'essor du Web Sémantique a fait se multiplier de tels outils. Certains sont gratuits, d'autres sont payants, c'est pourquoi nous ne présentons dans cette section que certains d'entre eux en privilégiant ceux sur lesquels nous nous sommes appuyés dans nos travaux. Une étude plus exhaustive de ce type d'outils peut être trouvée dans [Denny, 2004].

Protégé [Gennari et al., 2002] est certainement l'éditeur d'ontologie le plus utilisé. Développé à Stanford, cet outil a été construit sur les bases de la méthode 101 de Noy et McGuinness [Noy and McGuinness, 2001]. Dans le modèle de connaissance de Protégé les ontologies consistent en une hiérarchie de classes qui ont des attributs (slots) qui peuvent eux-mêmes avoir certaines

propriétés (facets). Il permet d'exprimer les ontologies créées dans une grande variété de langage allant de RDF à OWL Full.

L'ensemble des fonctionnalités offert par l'outil se regroupe suivant cinq points. La première permet de créer, modifier ou supprimer une classe et de lui attacher des propriétés qui peuvent elles-mêmes être caractérisées. L'utilisateur peut définir des instances de classes et leur affecter des propriétés, conformément à la définition des classes et des propriétés. Un des atouts majeurs de Protégé est la flexibilité des formulaires de saisie pour les instances. Ce formulaire dynamique s'adapte en fonction des classes décrites. Enfin, une fonctionnalité intéressante de l'outil concerne la possibilité d'effectuer des requêtes sur l'ontologie en cours d'édition. L'interface graphique intuitive de l'outil dissocie les différentes fonctionnalités grâce à des onglets dédiés. Par ailleurs, la popularité de Protégé a fait se concentrer autour de cet outil toute une communauté qui a développé un ensemble de plugins facilitant le développement, la gestion ou encore la visualisation d'ontologies. L'utilisation de ces plugins permet en outre de représenter l'ontologie dans divers formats et de s'assurer de sa cohérence.

KAON (KArlsruhe ONtology and Semantic Web tool suite) [Oberle et al., 2004] est un environnement open source pour la gestion d'ontologies. L'application principale se décompose en deux parties que sont une interface graphique qui permet d'interagir avec les utilisateurs et un noyau qui traite de l'accès et de la gestion des ontologies locales ou distribuées. KAON offre un ensemble de composants optionnels comme un module pour raisonner sur les ontologies, un module pour faciliter l'intégration d'ontologies dans les applications du Web Sémantique ou encore un ensemble d'outils pour la construction d'ontologies à partir de corpus de documents. Enfin, KAON offre la possibilité de gérer l'évolution des ontologies. Nous reviendrons plus en détail sur cette fonctionnalité dans la suite de ce mémoire.

OilEd [Bechhofer et al., 2001] est développé à l'université de Manchester et consiste en un simple éditeur d'ontologie, contrairement à Protégé et à KAON qui sont des environnements plus complets. On peut créer des hiérarchies de classes, spécialiser les rôles, et manipuler à travers l'interface de l'outil les types d'axiomes les plus courants (OilEd utilise les expressions définies dans les logiques de description). OilEd offre également les services du raisonneur FaCT, qui permet de tester la satisfaisabilité des définitions de classes et de découvrir des subsumptions implicites dans l'ontologie. Il permet enfin d'exporter les ontologies construites dans différents langages de représentation d'ontologies.

OntoLingua [Farquhar et al., 1997] fut un des premiers outils dédiés à la construction d'ontologie à avoir été conçu. Cet outil est développé au Knowledge System Laboratory de l'université de Stanford. OntoLingua permet de spécifier les ontologies au niveau symbolique : une grande partie des définitions des objets se fait directement dans le langage de représentation de connaissances KIF auquel à la fois le créateur et l'utilisateur de l'ontologie doivent se plier. Ainsi, contrairement aux autres outils déjà présentés, OntoLingua n'offre pas une grande flexibilité d'utilisation, notamment dans le choix du langage de représentation de l'ontologie.

Les deux derniers outils pour la construction d'ontologie que nous évoquons dans ce mémoire sont DOE (Differential Ontology Editor) et WebODE. DOE [Troncy and Isaac, 2002] est développé à l'Institut National de l'Audiovisuel sur la base de la méthode des ontologies différentielles de Bachimont [Bachimont et al., 2002] alors que WebODE [Corcho et al., 2002] est développé à l'université de Madrid autour de la méthodologie METHONTOLOGY [Fernandez et al., 1997]. Ces deux outils ont pour objectif d'assister les utilisateurs pour la construction d'ontologies d'un

point de vue méthodologique.

1.2 Évolution des données du Web

La nature dynamique du Web et de son contenu nécessite une gestion efficace de leur évolution dans un souci de cohérence avec les informations du monde réel et dans le but de faciliter l'exploitation de toutes ces données. Dans cette section, nous décrivons les travaux existants relatifs à la gestion de l'évolution des ressources du Web. Nous mettons l'accent sur la gestion de l'évolution des métadonnées au travers de l'évolution d'ontologies. Nous présentons également la manière dont est traité le problème de l'évolution des connaissances dans des domaines connexes tels que les bases de données.

1.2.1 Techniques existantes pour l'évolution des ressources du Web

Comme le montre la première partie de cet état de l'art, les ressources du Web sont représentées de différentes façons suivant leur nature. Le Web étant un espace dynamique, les données qu'il renferme sont amenées à évoluer. Dans cette section, nous étudions les techniques existantes pour la gestion de l'évolution des données du Web. Nous présentons tout d'abord les hypermédia adaptatifs [Brusilovsky, 1996] puis nous présentons les travaux de la communauté du Web réactif [Alferes et al., 2004].

Les Hypermédias Adaptatifs

Les hypermédia adaptatifs couvrent à la fois le domaine des hypermédia (comme leur nom l'indique) mais aussi celui de la personnalisation des données. Les systèmes à base d'hypermédia adaptatifs se proposent de construire un modèle des objectifs, préférences et connaissances des utilisateurs et d'utiliser un tel modèle pour satisfaire l'utilisateur lors de l'interaction de ce dernier avec le système. Le paradigme des hypermédia adaptatifs repose sur l'observation qu'un système hypermédia classique retourne la même information à chaque utilisateur sans tenir compte de ses caractéristiques. Or, les utilisateurs n'ont pas tous la même expérience ni les mêmes besoins. Les hypermédia adaptatifs se proposent donc d'intégrer les besoins des utilisateurs afin de les satisfaire au mieux. Les hypermédia adaptatifs considèrent un aspect particulier de l'évolution : l'adaptation de la présentation des données à l'utilisateur en fonction du profil de ce dernier. De Bra a identifié trois types de système [De Bra, 1999] :

- Les systèmes **adaptables** dans lesquels les utilisateurs peuvent soumettre leurs caractéristiques en répondant à un questionnaire. Le système s'adapte ensuite aux informations collectées. Il réutilise ces données afin d'une part de déduire les informations pertinentes pour l'utilisateur et, d'autre part, d'adapter la présentation de ces informations aux caractéristiques de l'utilisateur. Ce type de système est fréquemment utilisé par les applications d'e-commerce sur le Web pour mettre en valeur les offres sensées intéresser les clients.
- Les systèmes **adaptatifs** sont capables de construire automatiquement le profil d'un utilisateur à travers leurs interactions réciproques. Le système s'appuie sur les informations consultées par les utilisateurs afin d'en déduire leurs centres d'intérêts, leurs connaissances, etc. Le système traite les informations collectées sur les utilisateurs pour retourner les informations qui leur correspondent le mieux. Le système peut toutefois également enrichir les modèles des utilisateurs à travers des questionnaires remplis par ces derniers.
- Les systèmes dits **dynamiques** dans lesquels un modèle de l'utilisateur est construit de la même façon que dans les systèmes adaptatifs, mais la présentation des informations à l'uti-

lisateur est générée dynamiquement à partir d'éléments d'informations dits «atomiques». Ces éléments sont, en général, des morceaux de texte définis à partir de techniques d'analyse de la langue naturelle.

Les différentes applications à base d'hypermédia adaptatifs comme AHAM [Wu, 2002], AHA ! [De Bra and Calvi, 1998], DEXTER [Halasz and Schwartz, 1994] ou HERA [Frasincar et al., 2002] sont toutes construites suivant le même modèle (voir figure 1.2). Il contient quatre composants de base :

- Le **modèle du domaine** représente les informations contenues dans l'application ou celles auxquelles l'application peut accéder. Ce modèle est, dans la plupart des cas, représenté sous la forme d'un graphe orienté acyclique. Un peu à la manière d'une ontologie, les sommets du graphe sont les concepts (les fragments d'information, une page Web ou encore des ensembles plus larges) et les arcs représentent les relations entre les concepts. Un tel graphe permet de représenter une hiérarchie allant de l'information la plus abstraite à la plus concrète.
- Le **modèle d'utilisateur** contient les mêmes concepts et relations que le modèle du domaine ainsi qu'un coefficient représentant l'intérêt de l'utilisateur pour chaque élément du modèle. Ces coefficients sont utilisés pour déterminer quelle information retourner à l'utilisateur. Ils sont modifiés par le système après une analyse du comportement de l'utilisateur.
- Le **modèle d'application** contient un ensemble de règles permettant de relier le modèle de l'utilisateur au modèle du domaine. Dans le système AHAM, les règles sont exprimées dans un langage de type ECA (Event-Condition-Action).
- Enfin, le **moteur** permet d'appliquer les règles du modèle d'application. Il extrait l'information pertinente de la base de connaissance et génère la présentation la plus appropriée possible de ces informations.

Les systèmes hypermédia adaptatifs sont très utilisés dans le domaine de l'e-learning dans lequel le système assiste un étudiant dans l'apprentissage d'un cours. Le cours représente la connaissance, l'étudiant est l'utilisateur et le système s'adapte en fonction des connaissances déjà assimilées par l'étudiant et des difficultés rencontrées par celui-ci afin d'optimiser le processus d'apprentissage. Si le fonctionnement de ce genre de système est efficace pour des environnements clos où les connaissances contenues dans la base sont peu évolutives et ne concernent qu'un domaine bien précis, ils le sont beaucoup moins dans un environnement ouvert comme le Web où la connaissance est dynamique et hétérogène. Cependant, l'idée de considérer un modèle de l'utilisateur et de le faire évoluer reste une des principales contributions de ces approches et se retrouve maintenant utilisée dans des applications de recherche d'information sur le Web et notamment les principaux moteurs de recherche (voir section 1.3).

Le Web réactif

Le Web réactif [Bry and Patrânjan, 2005] est un nouveau paradigme dont la mise à jour des données du Web, l'échange d'information sur les événements affectant ces mises à jour sont les thèmes principaux. Le Web réactif concerne les applications comme l'e-commerce, les applications adaptatives, le Web Sémantique ou les services Web. Dans ce contexte, l'évolution du Web est décrite comme étant la modification d'une ou plusieurs ressources du Web. Ces modifications sont la conséquence d'événements ayant lieu au cours du temps ou encore le fruit du comportement des utilisateurs.

Exemple : Considérons un site Web de vente en ligne où les prix des articles sont donnés avec et

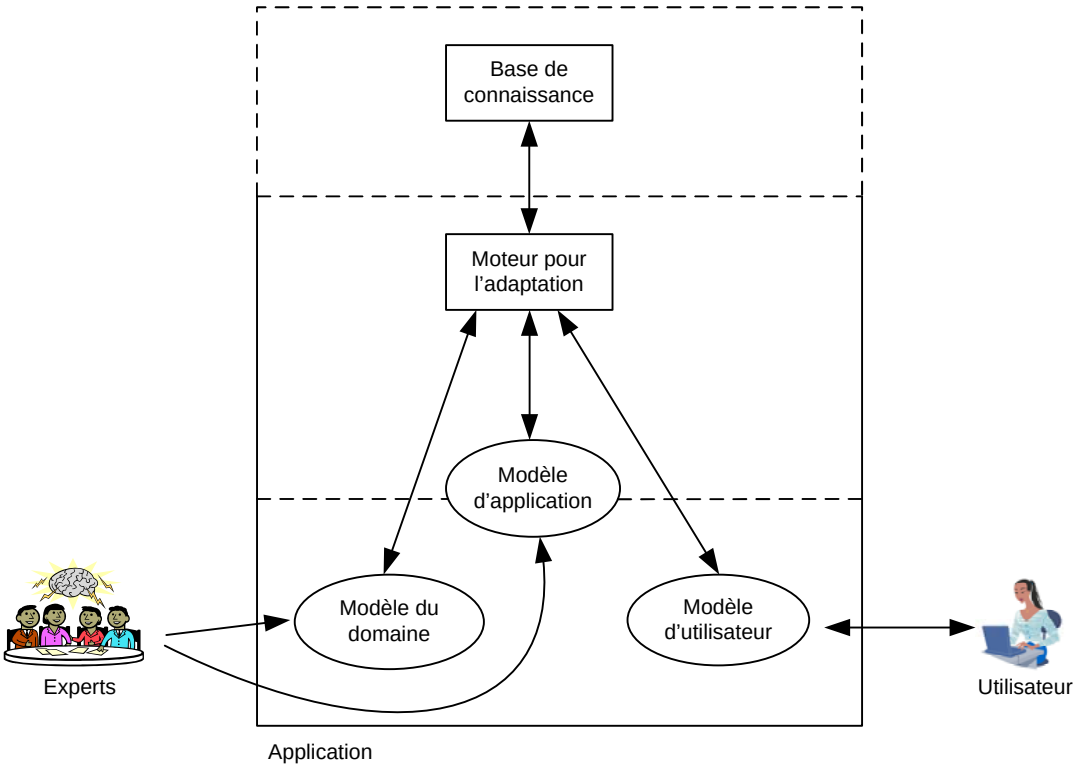


FIGURE 1.2 – Schéma fonctionnel d'un système hypermédia adaptatifs

sans la TVA. La modification de la valeur de la taxe entraînerait une mise à jour de tous les prix incluant la TVA. Une conception plus appropriée de l'application consisterait à ne stocker que le prix brut des produits et à appliquer le taux de la TVA pour connaître le prix TTC à afficher sur le site. Dans ce contexte, plusieurs cas de figure peuvent avoir lieu concernant la mise à jour des données :

- Le gestionnaire du site Web doit vérifier régulièrement la valeur du taux de la TVA et modifier manuellement toutes les pages Web du site en conséquence. Ce cas de figure représente le cas le moins «intelligent» des trois car tous les changements sont faits manuellement et sont donc source d'erreurs.
- Le gestionnaire souscrit à un service qui l'informe du changement de la valeur du taux de la TVA, ce qui déclenche une modification manuelle du prix des produits soumis à ce taux. Ceci représente le cas intermédiaire.
- Enfin, le gestionnaire du site Web peut également invoquer un service pour la mise à jour des données en cas de changement du taux de la TVA. Le service peut soit modifier la valeur de la TVA si cette dernière est stockée dans l'application ou modifier la valeur des prix en incluant la valeur de la TVA. Le service modifie également les données dépendantes du prix des articles venant d'être modifié quelque soit leur localisation sur le réseau. Ce dernier cas est le plus élaboré, et concerne la plupart des applications de commerce en ligne.

La vision du Web réactif s'intéresse au dernier cas énoncé dans cet exemple. La réalisation de ce cas nécessite la mise en œuvre d'un ensemble de concepts comme des gestionnaires d'évè-

nements complexes, des langages pour exprimer les événements et les actions à entreprendre en réponse aux événements ainsi que des mécanismes garantissant la cohérence des modifications effectuées.

Dans le Web réactif, les modifications, ou plus généralement l'évolution des ressources du Web (dans notre exemple *le prix des produits*), la définition des événements à prendre en compte (*le changement de la valeur de la TVA*) et les conditions à remplir pour que les modifications aient lieu (dans notre exemple une condition possible serait *l'existence d'un article*) se font à travers l'utilisation d'un langage de type ECA [Bry and Ecker, 2006]. Ce type de langage a une syntaxe intuitive et fonctionne de la façon suivante : **Lorsqu'un** événement survient, **si** une condition donnée est vérifiée **alors** exécuter une action définie. Ces langages proposent également des fonctionnalités pour pouvoir prendre en compte le comportement des utilisateurs ainsi que pour propager les modifications d'une ressource à une autre si celles-ci sont liées. Le langage Xchange proposé par Patrânjan [Patrânjan, 2005] est le langage de ce type le plus connus. La réactivité dans Xchange est exprimée au travers de règles (connues sous le nom de *trigger* dans le domaine des bases de données). Celles-ci peuvent tout d'abord exprimer des événements qui affectent le Web comme par exemple, la consultation d'une ressource par un utilisateur donné. Elles permettent également d'exprimer des conditions sur les ressources du Web comme par exemple, la valeur d'une ressource est-elle ou non supérieure à un certain seuil de référence (ces conditions sont exprimées dans le langage de requête Xcerpt [Schaffert, 2004]). Elles définissent les actions portant sur la mise à jour des ressources à effectuer. Enfin, le langage Xchange permet de définir un événement comme une combinaison d'événements et de propager les modifications à plusieurs ressources connectées à travers le réseau.

Les concepts du Web réactif ont été appliqués au Web Sémantique. Contrairement au Web où la plupart des sites ne fournissent que des informations uniquement consultables par des applications, le Web Sémantique bénéficie des concepts du Web réactif pour faire évoluer les ontologies utilisées pour décrire les ressources du Web. La modification des ressources du Web Sémantique (et des ontologies en particulier) et la propagation des changements aux ressources connectées doit se faire de manière cohérente. Dans ce contexte, un certain nombre de langages ECA ont été spécifiquement définis [Papamarkos et al., 2003] principalement pour propager les changements affectant les éléments d'une ontologie aux autres ontologies en relation. Le langage RDFTL développé par Papamarkos et al. [Papamarkos et al., 2004], par exemple, a pour objectif la mise à jour des documents RDF.

Le Web réactif a pour ambition de gérer l'évolution des données du Web de plusieurs manières différentes. Tout d'abord le type et la nature des données du Web sont considérés de manière abstraite. Les données sont vues de manière générale comme des ressources indépendamment de leur localisation ou de leur représentation. Ensuite l'évolution de ces ressources est une conséquence d'événements complexes dont les résultats sont propagés à toutes les ressources interdépendantes.

1.2.2 Évolution d'ontologies

L'émergence du Web Sémantique et l'utilisation massive d'ontologies comme métadonnées a rapidement donné lieu à de nouveaux problèmes de recherche. Parmi eux, celui de la gestion de l'évolution des ontologies. Les ontologies sont de plus en plus utilisées par des applications diverses et par conséquent plusieurs types d'évolution peuvent les affecter. La définition d'une

ontologie donnée par Gruber (voir section 1.1.2) laisse entrevoir plusieurs types d'évolution possible pour les ontologies.

Les ontologies représentant des descriptions de domaines du monde réel, des changements peuvent intervenir lorsque ces domaines évoluent [Fensel, 2001]. Ces changements sont le fruit des avancées technologiques ou plus généralement de la nature évolutive des connaissances. Les connaissances sur un domaine peuvent évoluer de plusieurs manières différentes. De nouvelles connaissances peuvent émerger ce qui conduit à définir de nouveaux concepts ou de nouvelles idées. Certaines connaissances peuvent sortir du domaine lorsqu'elles deviennent obsolètes, ce qui conduit à supprimer les concepts correspondant de l'ontologie. Enfin, la compréhension d'un domaine peut requérir une évolution de ses concepts. De nouveaux concepts peuvent être définis. Ils seront plus spécifiques que ceux existants si le domaine a besoin d'être plus précis, ou plus abstraits afin de simplifier le domaine et d'en faciliter sa compréhension. Tous ces changements à l'intérieur même d'un domaine doivent naturellement se refléter au niveau de l'ontologie le décrivant.

Des changements peuvent survenir dans la conceptualisation d'un domaine même si ce dernier n'a pas évolué. Ces changements sont souvent la conséquence d'une évolution des besoins pour lesquels l'ontologie a été construite. Considérons par exemple le domaine des espèces vivantes. Une ontologie représentant ce domaine pourrait contenir un concept *végétal* et un autre *animal*. Supposons que l'application utilisant cette ontologie voit ses besoins évoluer et nécessite une description plus précise du domaine. Des concepts tels que *mammifères* et *oiseaux* pourraient alors être ajoutés. Cet exemple illustre bien le fait que l'ontologie peut évoluer sans pour autant que le domaine qu'elle représente évolue car les concepts *mammifère* et *oiseau* faisaient évidemment parti du domaine des espèces vivantes bien avant la construction de l'ontologie.

Des changements de représentation des connaissances d'un domaine peuvent survenir. Pour répondre à des besoins particuliers, la représentation des éléments de l'ontologie peut nécessiter l'utilisation d'un autre langage. Ceci peut conduire à des problèmes de compatibilité entre les langages [Corcho and Pérez, 2000]. Par exemple le passage du langage OWL à RDF/S ne peut se faire sans perte de connaissances. Le changement de langage de représentation d'ontologie peut rétroactivement nécessiter de reconsidérer la conceptualisation du domaine. L'expressivité du langage peut, en effet, être un obstacle à la représentation de certains concepts et peut nécessiter leur complète ou partielle redéfinition.

Ces types de changements entraînent des modifications structurelles de l'ontologie, notamment dans la hiérarchie des concepts ou dans l'ensemble des relations qui relient les concepts entre eux. De plus, l'évolution d'ontologies peut se faire suivant différents modes comme décrits par Klein [Klein, 2004] :

- Le mode **tracé** dans lequel une trace de tous les changements intervenant dans l'ontologie (i.e. ajout ou suppression de concepts, ou de relations ...) est conservée. Il est parfois utile d'annuler certains changements notamment si les nouvelles connaissances décrites dans l'ontologie sont erronées ou entraînent des inconsistances, d'où l'intérêt de conserver une trace des changements.
- Le mode **non tracé** qui consiste à faire évoluer l'ontologie sans garder de traces. Certaines ontologies sont de grande taille (i.e. sont constituées de plusieurs milliers de concepts et de relations) et sont très dynamiques. Dans ce cas, il est difficile de conserver les informations relatives à tous les changements les affectant.

L'évolution d'ontologies est un problème complexe de part la diversité et la nature des changements qui peuvent survenir. Certaines approches comme celles pour la gestion des versions d'une ontologie s'inspirent de travaux réalisés dans le domaine des bases de données, d'autres proposent des méthodologies générales pour l'évolution d'ontologie. Certains langages ontologiques offrent des primitives pour modéliser l'information liée à l'évolution des connaissances. Dans la suite de cette section, nous allons présenter de manière plus détaillée l'ensemble de ces travaux.

Techniques pour la gestion des versions d'une ontologie

Une des principales techniques concernant l'évolution des ontologies est la gestion des versions d'une ontologie [Heflin and Hendler, 2000, De Leenheer, 2004] (voir figure 1.3). Cette technique directement inspirée des méthodes existantes pour la gestion des versions d'un schéma de base de données [Roddick, 1995] est définie par Klein et Fensel [Klein and Fensel, 2001] comme

> *«La possibilité de gérer l'évolution d'une ontologie en créant et gérant plusieurs variantes (ou versions) de cette ontologie.»*

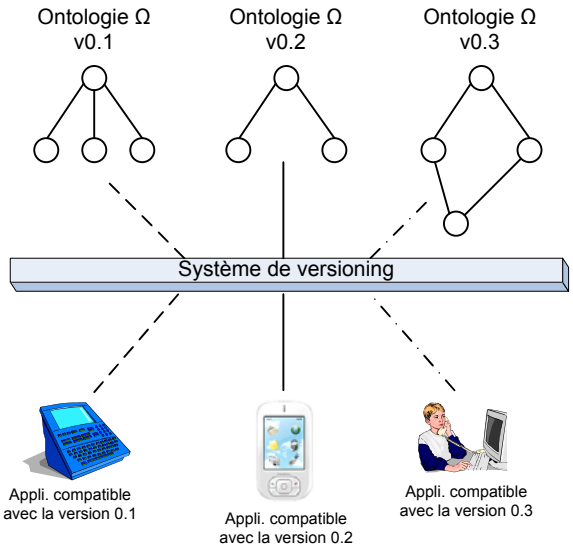


FIGURE 1.3 – Gestion des versions d'une ontologie

L'utilisation d'ontologies se généralisant de plus en plus, l'expert ayant construit une ontologie ne peut pas connaître toutes les applications qui utilisent son ontologie ni quelles parties de l'ontologie sont utilisées. De plus, l'ontologie construite par l'expert en question peut reposer sur d'autres ontologies dont les évolutions ne sont pas nécessairement connues par ce dernier. Pour toutes ces raisons, il est parfois nécessaire de considérer plusieurs versions d'une même ontologie. La gestion des versions d'une ontologie implique non seulement une gestion de l'ontologie qui évolue mais aussi une gestion des différentes versions de cette ontologie principalement au niveau de leur cohérence. Par conséquent, plusieurs problèmes, comme le stockage des différentes versions, leur identification, la caractérisation de la relation qui relie les différentes versions ainsi que leur compatibilité, doivent être traités.

Le principal problème concerne la caractérisation des changements devant donner lieu à une

nouvelle version de l'ontologie. Par exemple, la modification d'une classe d'une ontologie OWL par une autre classe sémantiquement équivalente doit-elle donner systématiquement lieu à une nouvelle version de l'ontologie ? Ce genre de problèmes est débattu dans [Heflin et al., 1999a] et [Klein et al., 2002]. L'accès à la bonne version est également problématique. En effet, le système doit permettre un accès transparent à la version demandée et pouvoir la mettre en relation avec des applications, des données ou plus généralement des éléments qui en dépendent.

Par ailleurs, la nature de la relation qui relie les différentes versions d'une ontologie est importante [Klein and Fensel, 2001]. Cette relation est appelée un changement de spécification. L'objectif est de rendre explicite les relations entre plusieurs versions d'un élément de l'ontologie. Ceci facilite la compréhension des changements de l'ontologie au cours du temps en vue de garantir une forme de compatibilité entre les versions. Klein et al. utilisent des métadonnées pour enrichir la description de cette relation [Klein et al., 2002].

Les changements donnant lieu à de nouvelles versions de l'ontologie peuvent être tracés. Dans l'approche de Plessers et De Troyer [Plessers and De Troyer, 2005] et celle de Noy et al. [Noy et al., 2006], les changements sont stockées dans un fichier de log sous la forme d'une ontologie. Les données stockées servent ensuite de base de connaissance sur lesquelles des requêtes, construites dans un langage adapté, peuvent être posées pour comprendre l'évolution de l'ontologie.

Les outils développés pour la gestion de versions d'ontologie assistent l'expert lors de la spécification des relations entre les versions. Plusieurs outils ont été développés pour identifier les différences syntaxiques entre les différentes versions d'une ontologie. Certains outils [Klein and Noy, 2003] n'apportent des informations que sur les changements dits élémentaires (i.e. opération consistant en la modification d'un seul élément du modèle OWL comme la cardinalité par exemple). PROMPTDIFF, développé par Noy et Musen [Noy and Musen, 2002] utilise plusieurs heuristiques pour comparer deux versions d'une ontologie et pour mettre en valeur leurs différences. Les heuristiques s'appuient sur la structure des ontologies.

De la même façon, OntoView [Klein et al., 2002] s'appuie sur la structure des ontologies pour les comparer. L'outil de Klein et al. génère en sortie une liste de changements considérés comme étant nécessaires pour passer d'une version à l'autre. Cette liste fournit en quelque sorte des mappings entre les différentes versions d'une ontologie.

Un nombre conséquent d'outils ont été développé pour la gestion des versions de documents RDF [Völkel and Groza, 2006, Auer and Herre, 2006]. Le plus aboutit d'entre eux s'appuie sur un algorithme proposé par Zeginis et al. [Zeginis et al., 2007]. L'algorithme utilisant quatre fonctions différentes, appelées «fonctions delta», pour comparer des documents RDF. Ces fonctions tiennent compte de la sémantique des documents RDF et exploitent les mécanismes d'inférences de RDF. Les auteurs comparent les fonctions delta selon plusieurs dimensions comme leur taille, leur exactitude, etc.

Le problème de la gestion des versions a donné lieu à d'autres travaux dans d'autres domaines [Menzies, 1999]. Dans le domaine des systèmes à base de connaissances, le problème principal concerne la caractérisation et la représentation des relations qui existent entre plusieurs versions des éléments de la base. Plusieurs approches sont proposées, comme les approches procédurales, logiques ou s'appuyant sur des réseaux [Menzies and Debenham, 2000]. Le domaine du génie logiciel est également confronté au problème de gestion des versions d'où le développement du système CVS [1] pour la gestion de configuration entre des fichiers informatiques.

1. http ://www.nongnu.org/cvs/

Méthodologies pour l'évolution d'ontologies

L'évolution d'ontologies (voir figure 1.4) est un problème supposé moins complexe que celui de la gestion des versions d'une ontologie du fait que la compatibilité entre les versions n'est pas à prendre en compte. Par contre, si les ontologies permettent de représenter les connaissances

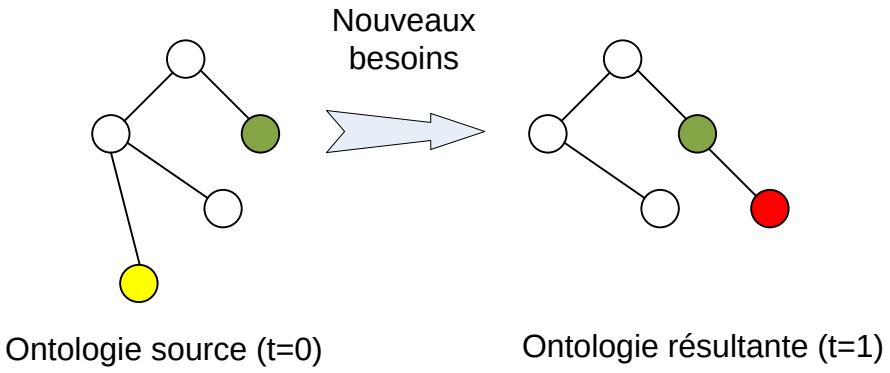


FIGURE 1.4 – L'évolution d'ontologie

d'un domaine, une de leurs principales caractéristiques est d'offrir la possibilité de raisonner sur ces connaissances. L'évolution d'une ontologie doit donc se faire de manière à ne pas entraver cette propriété fondamentale. Ainsi, pour un grand nombre d'approches, l'évolution d'ontologies se ramène à un problème de gestion de cohérence [1] garantissant la possibilité de raisonner sur l'ontologie après son évolution.

Pour Stojanovic [Stojanovic, 2004],

> *«L'évolution d'ontologie est l'adaptation à travers le temps d'une ontologie à des changements et la propagation cohérente de ces changements aux artefacts qui en dépendent»*

La première méthodologie appliquée au problème d'évolution d'ontologies fut proposée par Pons et Keller [Pons and Keller, 1997]. La méthode se déroule suivant trois étapes : la spécification de la demande de changement, l'identification des changements à apporter à l'ontologie et enfin la mise en œuvre de ces changements. Le principal problème de cette approche est la non considération des aspects validation et cohérence de l'évolution.

Klein et Noy [Klein and Noy, 2003] ont également proposé un framework général pour l'évolution d'ontologies. Leur approche s'appuie sur les différentes façons pour représenter les modifications dans une ontologie. Le framework couvre la transformation des données, la mise à jour de l'ontologie, la cohérence du raisonnement, la vérification des changements et l'accès aux données (ce dernier point est spécifique au problème de la gestion des versions d'une ontologie évoqué précédemment).

La méthode référence pour l'évolution d'ontologie est celle développée par Stojanovic et al. [Stojanovic et al., 2002]. Nous allons décrire les six étapes de la méthode et confronter les travaux existants sur l'évolution d'ontologie par rapport à ces différentes étapes.

La première étape est la **découverte des changements**. Cette phase consiste en l'analyse des données pour découvrir les changements. Selon Stojanovic et al. [Stojanovic et al., 2002],

1. Une ontologie est cohérente si elle satisfait à tous les invariants du modèle d'ontologie.

la détection des changements peut être dirigée par différents facteurs. L'étude de *l'utilisation de l'ontologie* permet de définir quelles parties de l'ontologie sont les plus utilisées par les applications afin de concentrer les efforts de détection de changement sur ces parties. Leea et al. s'inspirent de cette approche dans leurs travaux pour la découverte de services sur la Grille [Leea et al., 2007]. Une autre solution consiste en *l'étude des données* relatives au domaine représenté par une ontologie. Cette tâche peut être effectuée de manière manuelle par des experts du domaine, ou automatiquement via l'étude de corpus de documents (voir la section 1.1.2) [Weichselbraun et al., 2007, Hahn and Marko, 2002]. Dans l'approche proposée par Castano et al. [Castano et al., 2006], des techniques d'appariements d'ontologies sont mises en œuvre pour détecter les changements et en particulier les nouveaux concepts à ajouter à l'ontologie. Dans certaines approches, ce sont les métadonnées du Web qui sont étudiées pour découvrir quels sont les changements susceptibles d'affecter l'ontologie. Maynard et al. présentent dans [Maynard et al., 2007] une idée originale pour l'évolution d'ontologie dans laquelle l'évolution de l'ontologie est dirigée par les modifications sur des métadonnées et inversement, les métadonnées suivent les possibles modifications de l'ontologie qui leur est liée. L'étude de la sémantique des tags communément employés dans les folksonomies devient également une piste de recherche de plus en plus explorée pour l'évolution d'ontologie. Les techniques existantes [Specia and Motta, 2007, Gendarmi and Lanubile, 2006] utilisent les ontologies existantes afin de définir la sémantique des tags puis si des ressources sont annotées avec des tags dont certains sont inconnus, ces tags pourront être proposés comme label de nouveaux concepts à ajouter à l'ontologie.

Au cours de la seconde étape, la **représentation des changements** est définie. Avant d'être appliqués aux éléments d'une ontologie, les changements doivent être définis de manière cohérente avec le modèle d'ontologie utilisé [Sindt, 2003]. De manière plus spécifique, certaines approches se concentrent sur la définition et la formalisation d'un ensemble de changements possibles sur une ontologie. Ces changements sont regroupés en deux catégories, les changements dits *élémentaires* et les changements *composites*. La première catégorie représente des changements simples comme l'ajout ou la suppression d'un concept alors que la seconde concerne les changements se décomposant en une série de changements élémentaires. Stojanovic et al. ont identifié 12 changements composites [Stojanovic et al., 2002], Noy et Klein en ont identifiés 22 dans [Noy and Klein, 2004] et Stuckenschmidt et Klein, 120 dans [Stuckenschmidt and Klein, 2003]. Le nombre de changements est en fait dépendant du modèle de l'ontologie utilisé. Plus le modèle est élaboré, plus le nombre de changements possibles est important.

La troisième étape correspond à la définition de la **sémantique des changements**. Elle consiste à déterminer les effets des changements détectés à l'étape 1 sur l'ontologie. Cette phase est certainement la plus importante puisque cette sémantique sera réutilisée dans la dernière étape du processus lors de la validation de l'évolution. Une approche originale pour capturer la sémantique des changements est proposée par Flouris et Plexousakis [Flouris and Plexousakis, 2005]. Elle consiste à appliquer des techniques développées dans le domaine de la révision des croyances (belief change) [Gärdenfors and Rott, 1995]. Une croyance est définie comme une information qui peut être remise en question. Le processus de remise en question se fait suite à l'apprentissage d'une nouvelle information. On distingue généralement deux types de changements : la mise à jour et la révision. Au cours de la mise à jour, la nouvelle information concerne le présent alors que toutes les croyances concernent le passé. La mise à jour consiste en la modification des croyances passées pour prendre en compte la nouvelle information. La révision est différente car la nouvelle information et l'ensemble des croyances font référence à la même situation (passée ou présente). Une inconsistance doit donc être levée et le principe de révision consiste en l'acceptation de la nouvelle information en tant que croyance sans générer d'inconsistance. La révision

des croyances s'appuie sur des logiques classiques et non pas sur des logiques de description. L'application de ces techniques contribue à l'automatisation de la détermination des effets indésirables des modifications. Un certain nombre d'arguments en faveur de ce type d'approche ainsi que certains résultats basés sur l'utilisation des postulats AGM [Alchourron et al., 1985] sont donnés par Flouris [Flouris et al., 2006] et par Kang [Kang and Lau, 2004] dans leurs travaux. Les postulats AGM décrivent ce que doit être un bon processus de révision à savoir un nombre minimum de changements et un maximum de cohérence. Gutierrez et al. ont appliqué les techniques de la révision de croyance à la mise à jour de documents RDF [Gutierrez et al., 2006]. Leur formalisation permet de déterminer automatiquement le résultat d'une modification d'un document. Une approche similaire est appliquée par Lee et Meyer [Lee and Meyer, 2004] pour la mise à jour d'ontologie décrite dans la logique de description $\mathcal{ALU}$. Ribeiro et Weissermann [Ribeiro and Wassermann, 2007] appliquent le concept des opérateurs noyau, emprunté à Hansson [Hansson, 1994] et propre à la révision des croyances, aux formalismes pour la représentation des connaissances afin de faciliter l'évolution d'ontologies.

La quatrième étape est la **mise en œuvre des changements**. Durant cette phase plusieurs aspects sont traités. Tout d'abord, une liste de tous les changements à effectuer est dressée. Cette liste est destinée à l'ingénieur en charge de l'ontologie. C'est ce dernier qui prend la décision finale d'appliquer ou non les changements proposés. Les modifications sur les éléments de l'ontologie sont ensuite appliquées. Stojanovic et al. conseillent de conserver une trace des changements effectifs. Le problème de la mise en œuvre des changements dans le processus d'évolution d'ontologie est vu et traité par certains comme un problème de transformation de modèle. Dans ce sens, Jin et al. ont défini un framework pour l'évolution d'ontologie [Jin et al., 2005] dans lequel des techniques de transformation de modèle empruntées au domaine du génie logiciel sont appliquées. La représentation sous forme de graphe d'une ontologie a fait se développer une famille d'approches pour l'évolution d'ontologie basée sur la transformation de graphes comme dans [Leenheer and Mens, 2007]. Dans ces approches, les ontologies sont vues comme des graphes dont les sommets représentent les concepts du domaine et les arêtes sont les relations entre les concepts. Une suppression ou un ajout d'un concept ou d'une relation dans l'ontologie s'apparente à une suppression ou un ajout de sommets ou d'arêtes dans un graphe. Ces problèmes sont déjà traités dans la théorie des graphes. Ainsi, dans [Trinkunas and Vasilecas, 2007], les auteurs s'inspirent de transformation de graphes pour transformer une ontologie OWL DL en un modèle Entité-Relation. Souvent, l'application de techniques de transformations de graphe au problème d'évolution d'ontologie s'accompagne d'un changement de formalisme de l'ontologie.

La cinquième étape est la **propagation des changements**. Elle consiste à propager les changements d'une ontologie à toutes les autres ressources qui lui sont liées, de manière cohérente [Haase and Stojanovic, 2005]. Ceci concerne plus particulièrement les ontologies distribuées [Maedche et al., 2003] ou encore les ontologies «modulaires» [Stuckenschmidt and Klein, 2003]. L'idée d'utiliser les systèmes multi-agents pour propager les changements d'une ontologie à une autre a été proposée dans [Afsharchi and Far, 2006]. L'ontologie représente la connaissance du monde qu'a un agent et cette ontologie évolue grâce à la connaissance que l'agent en question acquiert à travers ses interactions avec d'autres entités (d'autres agents dans la majorité des cas). Ensuite, cette nouvelle connaissance peut être assimilée de manière automatique comme c'est le cas dans [Afsharchi and Far, 2006] ou de manière supervisée par des humains.

La sixième et dernière étape de la méthodologie est la **validation**. Elle se propose de valider le processus en prenant principalement en compte la sémantique des changements définie au cours de la troisième étape. Ce point se concentre sur la cohérence de l'ontologie obtenue après évolution.

L'évolution d'ontologies dans les langages de représentation d'ontologies

L'évolution d'ontologie est également pris en compte au niveau des langages de représentation d'ontologie et notamment dans certains standards du W3C.

Les spécifications du langage OWL [1] définissent un ensemble de primitives pour l'évolution d'ontologie et principalement la gestion des versions d'une ontologie. Les fonctionnalités qui ont été majoritairement inspirées par celles du langage SHOE [Heflin et al., 1999b], sont les suivantes :

- *owl :versionInfo* : L'objet d'une déclaration owl :versionInfo est une chaîne de caractères fournissant des renseignements à propos de la version de l'ontologie, par exemple, des mots-clés. Cette déclaration ne contribue pas à la signification logique de l'ontologie hormis celle fournie par le modèle théorique RDF/S. Bien que cette propriété sert habituellement à faire des déclarations à propos des ontologies, elle peut s'appliquer à n'importe quelle structure du langage. Par exemple, on pourrait associer une déclaration owl :versionInfo à une classe OWL. La propriété owl :versionInfo est une instance de la classe owl :AnnotationProperty.
- *owl :priorVersion* : Une déclaration owl :priorVersion contient une référence à une autre ontologie. Elle identifie l'ontologie indiquée comme une version précédente de l'ontologie courante. Cette déclaration n'a pas de signification dans la sémantique théorique du modèle hormis celle fournie par le modèle théorique RDF(S). Par contre, elle peut être exploitée par un programme pour organiser les ontologies par version. Le domaine et l'image de la propriété OWL intégrée owl :priorVersion sont de la classe owl :Ontology. La propriété owl :priorVersion est une instance de la classe owl :OntologyProperty
- *owl :backwardCompatibleWith* : Une déclaration owl :backwardCompatibleWith contient une référence à une autre version précédente de l'ontologie courante, en indiquant que cette dernière est rétrocompatible. Elle indique notamment que tous les identificateurs de la version précédente gardent la même interprétation dans la nouvelle version. C'est une indication de confiance fournie aux auteurs afin qu'ils privilégient la nouvelle version (en mettant simplement à jour les déclarations d'espaces de nommage et les déclarations owl :imports). Deux versions qui ne sont pas reliées par la propriété owl :backwardCompatibleWith ne sont pas compatibles. La propriété owl :backwardCompatibleWith n'a pas de signification dans la sémantique théorique du modèle, hormis celle fournie par le modèle théorique RDF/S. Le domaine et l'image de la propriété OWL owl :backwardCompatibleWith, sont de la classe owl :Ontology alors que la propriété owl :backwardCompatibleWith est une instance de la classe owl :OntologyProperty.
- *owl :incompatibleWith* : Une déclaration owl :incompatibleWith contient une référence à une autre ontologie non compatible avec l'ontologie courante. Cette déclaration s'adresse aux auteurs d'ontologies voulant signaler explicitement que les documents ne peuvent pas être mis à jour conformément à la nouvelle version sans vérifier si des changements sont nécessaires. La propriété owl :incompatibleWith n'a pas de signification dans la sémantique théorique du modèle hormis celle fournie par le modèle théorique RDF/S. Le domaine et l'image de la propriété OWL intégrée owl :incompatibleWith sont de la classe owl :Ontology. La propriété owl :backwardCompatibleWith est une instance de la classe owl :OntologyProperty.
- *owl :DeprecatedClass* et *owl :DeprecatedProperty* : La contre-indication est un mécanisme couramment employé dans la gestion de configuration dans le domaine du génie logiciel (cf. par exemple, le langage de programmation Java) pour indiquer qu'une caractéristique particulière, quoique conservée pour des raisons de rétrocompatibilité, est suscep-

1. http ://www.w3.org/TR/owl-features/

tible de disparaître dans le futur. Auquel cas, on dira qu'un identificateur donné est du type owl :DeprecatedClass ou owl :DeprecatedProperty, où owl :DeprecatedClass est une sous-classe de rdfs :Class, et owl :DeprecatedProperty une sous-classe de rdf :Property. La contre-indication d'un terme signifie qu'il ne devrait plus être employé dans les nouveaux documents annotés par une ontologie. Cela permet à l'ontologie de rester compatible pendant le remplacement progressif de l'ancien vocabulaire (utiliser la contre-indication n'est donc logique qu'en combinaison avec la rétro-compatibilité). Les données et les applications anciennes peuvent migrer plus facilement tout en facilitant l'adoption de la nouvelle version. Cela n'a pas de signification dans la sémantique théorique du modèle hormis celle fournie par le modèle théorique RDF/S. Toutefois, les outils de création peuvent s'en servir pour avertir les utilisateurs lors de la vérification d'un balisage OWL.

Avery et Yearwood [Avery and Yearwood, 2003] ont identifié un ensemble de constructeurs manquant au niveau de OWL en matière de gestion des versions d'une ontologie. Ces constructeurs concernent la gestion du renommage d'une classe ou d'une propriété, la gestion de la suppression d'une classe ou d'une propriété, la possibilité de restreindre une classe, la redéfinition du domaine et co-domaine d'une propriété ainsi que ses caractéristiques (symétrie, transitivité ...), la coalescence de plusieurs classes ou propriété et enfin l'éclatement d'un classe ou d'une propriété en plusieurs. En conséquence, les auteurs proposent une extension de OWL via le langage dOWL. Les nouveaux constructeurs apportés par dOWL sont :

- *dowl :removeClass* et *dowl :removeClassRestrictions* : Ces éléments contiennent un élément rdf :about pointant sur la classe ou ses restrictions à supprimer.
- *dowl :removeProperty*, *dowl :removeRange* et *dowl :removeDomain* : de la même façon, ces éléments contiennent un élément rdf :about pointant sur la propriété, son domaine ou co-domaine à supprimer de l'ontologie.
- *dowl :renameClass* : contient un élément rdf :ID contenant le nouveau nom de la classe et un élément rdf :about pointant sur la classe à renommer.
- *dowl :renameProperty* : contient un élément rdf :ID contenant le nouveau nom de la propriété et un élément rdf :about pointant sur la propriété à renommer.
- *dowl :defineAsTransitive*, *dowl :defineAsSymmetric*, *dowl :defineAsFunctional*, *dowl :defineAsInverseFunctional* et *dowl :defineAsNormalProperty* : tous ces éléments contiennent un rdf :about pointant sur la propriété dont une des caractéristiques (transitivité, symétrie, ...) doit être modifiée.
- *dowl :coalesceClass* : contient un élément rdf :ID dénotant le nom de la nouvelle classe et un élément owl :unionOf rassemblant les classes à fusionner.
- *dowl :coalesceProperty* contient un élément rdf :ID dénotant le nom de la nouvelle propriété et un élément owl :unionOf rassemblant les propriétés à fusionner.
- *dowl :divideClass* contient le nom de la classe à définir dans l'élément rdf :about et une énumération de classes avec l'élément owl :unionOf.
- *dowl :divideProperty* contient le nom de la propriété à définir dans l'élément rdf :about et une énumération de propriétés avec l'élément owl :unionOf.

L'ensemble des constructeurs définis dans dOWL permet de modifier une ontologie. Cependant, les constructeurs du langage et la sémantique qui leur est associée ne sont pas suffisant pour garantir la cohérence de l'ontologie obtenue après évolution.

L'évolution des documents RDF est la raison pour laquelle le langage RUL a été proposé [Magiridou et al., 2005]. S'inspirant directement des langages de requête et de visualisation RQL et RVL, RUL permet de modifier les sommets et les arcs d'un graphe RDF sans violer ni la

sémantique de RDF ni celle de RDF/S. De plus, RUL prend en compte la modification des instances de classes et de propriétés.

D'autres travaux sur la prise en compte de l'évolution d'ontologies dans les langages ontologiques ont été menés, notamment sur les logiques de descriptions. Ces langages ont été proposés à des fins différentes.

Chen et Matthews [Chen and Matthews, 2007] ont étendu la logique de description $\mathcal{SHIQ}$ avec des propriétés temporelles pour caractériser les changements d'une ontologie. Plessers et al. [Plessers et al., 2007] ont défini le langage CDL (Change Detection Language) s'appuyant sur les logiques temporelles pour exprimer les changements en termes de différences entre les versions d'une ontologie.

D'autres langages ont été développés pour la gestion des versions d'une ontologie. Ainsi, Huang et Stuckenschmidt ont proposé une logique temporelle basée sur un ensemble d'opérateurs pour raisonner sur plusieurs versions d'une ontologie. Le système MORE a été développé dans le but de charger plusieurs versions de l'ontologie et de les interroger. Le langage OWL-MeT [Keberle et al., 2007] permet également de manipuler les différentes versions d'une ontologie. L'originalité du langage réside dans l'introduction d'une métrique permettant de revenir un nombre de versions dans le passé. Le temps dans cette approche est considéré comme linéaire et discret.

Comme le montre cette étude des différents langages pour la prise en compte de l'évolution d'ontologies, la majorité d'entre eux contribue à la gestion des différentes versions d'une ontologie. La notion de temps, pourtant prépondérante dans les phénomènes évolutifs, est souvent passée sous silence. A part dans les travaux de Keberle et al. [Keberle et al., 2007] et ceux basés sur des logiques temporelles où la notion de temps est clairement caractérisée, les autres approches ne s'appuient pas (ou peu) sur le temps pour définir leur langage. Dans la plupart des langages ontologiques que nous avons présentés, et principalement ceux basés sur les logiques de description, l'accent est mis sur la consistance de l'ontologie obtenue après l'application du processus d'évolution.

Outils pour l'évolution d'ontologies

L'ensemble des approches et langages pour l'évolution d'ontologies sont en règle générale supportés par des outils (voir section 1.1.2). Nous allons présenter certains d'entre eux dans cette section à travers une comparaison de leurs fonctionnalités dédiées à la prise en compte de l'évolution d'ontologie. Cette comparaison s'inspire et complète celle proposée par Rogozan [Rogozan, 2005].

Fonctionnalité	**Protégé**	**KAON**	**OntoStudio**	**OilEd**	**WebOde**
Méthodologie générale supportée	101	6 phases	On-To-Knowledge	×	Methontology
Librairie des versions d'ontologies	∼	∼	∼	∼	∼
Support pour l'analyse de corpus de documents	×	∼	×	×	×

Fonctionnalités pour l'édition des changements élémentaires	✓	✓	✓	✓	✓
Fonctionnalités pour l'édition des changements complexes	∼	∼	×	×	×
Moyens pour spécifier les modifications sur les éléments de l'ontologie	×	∼	×	×	×
Annotation des changements par un ensemble des métadonnées	∼	∼	∼	∼	∼
Vérification de la consistance de l'ontologie après évolution	✓	✓	✓	✓	✓
Aide à la résolution des inconsistances introduites par les changements	✓	×	×	×	×
Support au travail collectif avec gestion des conflits	∼	∼	×	×	∼
Conservation de la trace des changements	∼	∼	∼	×	×
Identification des changements entre les versions	∼	×	×	×	×
Caractérisation de la relation sémantique entre deux versions consécutives d'une ontologie	×	×	×	×	×
Analyse a priori des effets des changements sur la cohérence de l'ontologie	×	×	×	∼	×
Analyse des effets des changements sur la compatibilité des versions	×	×	×	×	×
Propagation des changements dans les objets référencés par l'ontologie	×	×	×	×	×
Propagation des changements dans les ontologies dépendantes	∼	×	×	×	×
Mise en évidence des différences entre deux versions d'une ontologie	✓	×	✓	×	∼
Possibilité d'annuler les changements sur une ontologie	∼	✓	∼	×	∼

Gestion de l'accès aux objets référencés des versions multiples	×	×	×	×	×

TABLE 1.3: Comparaison des différentes fonctionnalités liées à l'évolution dans les outils pour la gestion d'ontologies

Les plates-formes comparées dans le tableau [1] 1.3 sont généralement étendues par d'autres outils. Par exemple, SemVersion [Völkel and Groza, 2006] qui est un outil pour la gestion de versions de document RDF peut s'intégrer comme un plugin dans Protégé.

Par ailleurs, le système ReTAX+ [Lam et al., 2004] a été développé spécifiquement pour la gestion de l'évolution de taxonomies. Le système assiste les experts lors de l'ajout, la suppression, la fusion et l'éclatement de concepts ou encore la modification de leurs attributs. Le système garantit la cohérence de la taxonomie résultante.

L'étude de la prise en compte de l'évolution d'ontologie dans les outils a mis en évidence un certain nombre de manques (voir le tableau 1.3). Aucun support n'est fourni pour l'analyse de corpus de documents alors que certaines approches, et notamment celle de Stojanovic et al. [Stojanovic et al., 2002], préconise l'utilisation de ce type d'élément afin de détecter les évolutions du domaine. Un tel manque constitue une entrave à l'automatisation du processus d'évolution. De plus, aucune stratégie d'évolution ne peut être définie et aucune information propre au phénomène d'évolution n'est conservée. La gestion des versions d'une ontologie est quant à lui davantage pris en compte et l'accent est mis sur la cohérence de l'ontologie et la propagation des changements aux artefacts qui en dépendent.

L'évolution des schémas de bases de données

L'évolution des schémas de base de données [Banerjee et al., 1987], domaine de recherche plus ancien étudié dans le domaine des bases de données orienté-objet, a souvent été comparée au problème d'évolution d'ontologie. Ce problème est défini comme la gestion à travers le temps des changements sur le schéma sans perte de données alors que le système continue de fonctionner. L'évolution des schémas de bases de données a été étudié en détail par Ram et Shankaranarayanan [Shankaranarayanan and Ram, 2003] ainsi que Rahm et Bernstein [Rahm and Bernstein, 2006]. Les principaux problèmes sur l'évolution des schémas émergeant de ces études concernent les changements en cascade (i.e. changement induisant d'autres changements), la garantie de la cohérence du schéma et la propagation des changements du schéma aux données de la base. Ces deux derniers points font également partie des problèmes auxquels se heurte la communauté Web Sémantique pour l'évolution d'ontologie. Cependant les deux approches comportent quelques différences comme le montre l'étude de Noy et Klein [Noy and Klein, 2004]. Les principales différences identifiées sont les suivantes :

- Contrairement aux schémas de base de données, l'ontologie peut être considérée comme une donnée à part entière dans les applications.
- Les langages de représentation d'ontologie sont en général plus expressifs. Les changements à ce niveau sont donc plus complexes à gérer. Du fait de cette expressivité, les ontologies sont plus riches du point de vue de la sémantique mais l'utilisation de raisonneur peut faciliter la vérification de la cohérence de l'ontologie.

1. Légende : × signifie aucun support, ∼ un support partiel et ✓un support total

- En général, les schémas de base de données ne sont pas faits pour être étendus contrairement aux ontologies.
- La construction d'ontologie est le fruit d'un travail collaboratif contrairement à celle des schémas de base de données. Les problèmes de synchronisation entre les différents collaborateurs rendent la construction d'ontologie encore plus difficile.

En dépit de ces différences, les travaux sur l'évolution des schémas de base de données ont inspiré et influencé ceux de la communauté Web Sémantique sur l'évolution d'ontologies et ceci principalement pour la gestion des versions. Mostowfi et Fotouhi [Mostowfi and Fotouhi, 2006] appliquent des techniques empruntées à l'évolution de schémas de base de données pour la transformation d'ontologies.

1.3 Exploitation des données du Web

La troisième et dernière partie de notre état de l'art concerne l'étude de l'exploitation des données et métadonnées du Web. Dans cette étude, nous nous focalisons sur la recherche d'information sur le Web et principalement sur les approches et outils dédiés à la recherche adaptative. Dans ce contexte, nous détaillons les langages de requêtes et les techniques pour l'expansion de requêtes.

1.3.1 Recherche d'information

Dans cette section, nous présentons les différents modèles existants pour la recherche d'information. Cette présentation est nécessaire pour comprendre le fonctionnement des approches existantes que nous présenterons plus tard dans ce mémoire.

Comme évoqué au cours de l'introduction, la recherche d'information sur le Web se fait quasi exclusivement à l'aide des moteurs de recherche et l'utilisation de requêtes constituées d'un ensemble de mots clés suivant le modèle de van Rijsbergen (voir figure 1.5). La plupart de ces moteurs de recherche proposent depuis peu un ensemble de répertoires thématiques regroupant les ressources Web par domaine, ce qui constitue un premier filtre pour la recherche.

Techniques existantes pour la recherche d'information sur le Web

Les différentes techniques existantes pour la recherche d'information sur le Web se répartissent en quatre grandes catégories : l'approche booléenne, vectorielle, cognitive et probabiliste. Dans cette section, nous allons détailler ces différentes approches.

L'approche booléenne

Le modèle booléen est le modèle le plus ancien et aussi le plus utilisé. Ce modèle est construit sur la logique booléenne et la théorie des ensembles. Dans ce modèle, un document D est représenté par la conjonction des termes constituant le document ($D = t_1 \wedge t_2 \wedge \ldots \wedge t_n$) et est considéré comme un ensemble de termes. Une requête Q est définie comme une expression booléenne construite avec les opérateurs de conjonction ($\wedge$), de disjonction ($\vee$) et de négation ($\neg$). Un document D vérifie la requête Q si et seulement si $D \supset Q$.

Les principales caractéristiques de ce modèle sont :

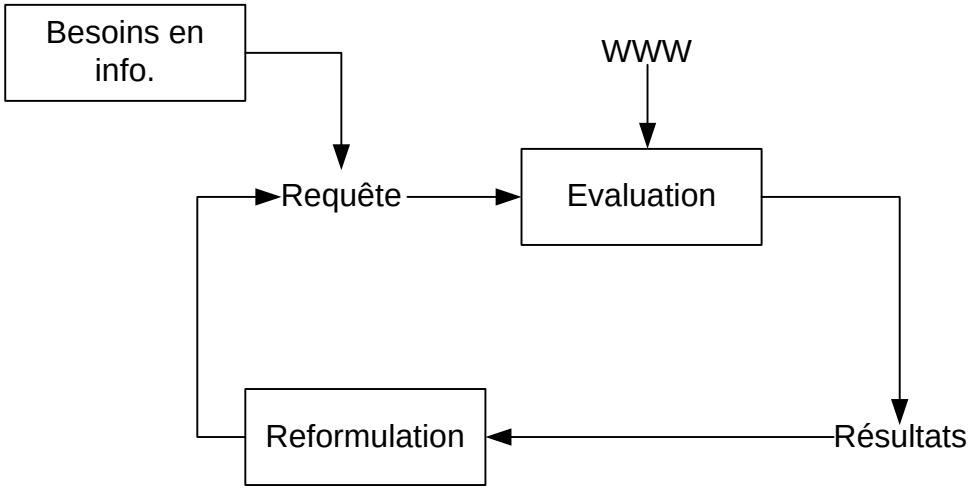


FIGURE 1.5 – Le modèle de van Rijsbergen pour la recherche d'information

- La correspondance entre le document et la requête est stricte. La notion de proximité n'existe pas dans ce modèle.
- Aucune distinction n'est faite entre les documents pertinents. Le modèle n'offre pas la possibilité de déterminer si un document est plus pertinent qu'un autre.
- Les opérateurs booléens permettent de construire des requêtes complexes.

Ce modèle peut être étendu par l'ajout de coefficients de pondération sur les termes des documents dans le but de pouvoir distinguer les documents pertinents entre eux.

L'approche vectorielle

Le modèle vectoriel répond au problème de la représentation mathématique du contexte d'un document. Le modèle est construit à partir d'un vocabulaire $T = \{t_i\}, i \in \mathbb{N}, 1 \leq i \leq N$ (i.e. un ensemble de N termes). Tous les documents sont décrits suivant ce vocabulaire. Un document D est représenté par un vecteur $\overrightarrow{D}$ de dimension N, $\overrightarrow{D} = (w_1, w_2, \ldots, w_N)$ où chaque élément w_i consiste en un poids associé au terme d'indice i de T et au document D. De la même façon, une requête Q est représentée par un vecteur $\overrightarrow{Q} = (q_1, q_2, \ldots, q_N)$ où les éléments q_i sont des coefficients. Les coefficients sont soit attribués de manière arbitraire soit définis en considérant, par exemple, le nombre d'occurrences d'un terme de T dans le document.

Étant donnée une représentation vectorielle d'un corps de documents, on peut introduire une notion d'espace vectoriel sur l'espace des documents en langage naturel et définir la notion de proximité entre documents. En introduisant des mesures de similarité adaptées, on peut quantifier la proximité sémantique entre différents documents. On peut déterminer de cette manière les documents pertinents. Les mesures de similarité sont choisies en fonction de l'application. Une mesure habituelle est la mesure de l'angle entre D et Q en utilisant la définition du produit scalaire de D et Q.

Le modèle vectoriel possède les caractéristiques suivantes :

- La correspondance entre la requête et les documents n'est pas stricte. Ce modèle introduit la notion de similarité entre les documents et la requête ou les documents entre eux.
- La mesure de similarité utilisée introduit une relation d'ordre sur les documents pertinents.

Le modèle vectoriel permet également de construire des ensembles de documents partageant une même thématique.

L'approche probabiliste

La théorie des probabilités est utilisée comme un moyen de modéliser le processus de recherche d'information. Dans les systèmes de recherche d'informations traditionnels, les documents sont extraits en réponse à une requête quand l'ensemble des termes clés d'un document s'apparente dans une certaine mesure aux termes d'une requête. Dans de tels cas, les documents sont dits pertinents par rapport à cette requête.

Soit R et NR représentant respectivement les événements «un document est pertinent» et «un document n'est pas pertinent». Soit D un document, l'idée de base dans un modèle probabiliste est de tenter de déterminer les probabilités $P(R|D)$ et $P(NR|D)$ pour une requête donnée. Ces deux probabilités signifient respectivement : si on retrouve le document D, quelle est la probabilité d'obtenir l'information pertinente et non pertinente.

Plus récemment, des modèles probabilistes plus élaborés ont été proposés dans le cadre de la recherche d'information. Ces modèles font appel à des réseaux bayesiens ou des réseaux d'inférence.

Les trois modèles présentés jusqu'ici (i.e. booléen, vectoriel et probabiliste) ne tiennent pas compte de la sémantique des termes constituant le vocabulaire, c'est pourquoi un nouveau type de modèle, les modèles cognitifs, a été défini.

L'approche cognitive

Ces modèles ne se limitent pas à faire une correspondance simple des représentations des requêtes avec celles des documents, mais ils permettent aussi l'évaluation sémantique de cette correspondance. Cette évaluation se base sur les comparaisons sémantiques entre le document et la requête. Elle se traduit par des déductions permettant de définir des fonctions de correspondance beaucoup plus perfectionnées, et ce à partir d'un ensemble de relations sémantiques entre les termes. Les techniques utilisées pour établir ces déductions sont issues de l'intelligence artificielle.

Ce type de modèle a donné lieu à de nouvelles approches pour la recherche d'information sur le Web. La prise en compte du contexte que nous allons détailler dans les sections à venir est l'une d'entre elles.

Le contexte dans la recherche d'information sur le Web

Dans la littérature, le contexte est défini de plusieurs manières différentes [Lawrence, 2000], [Freund and Toms, 2005] et est pris en compte soit de manière explicite, directement au niveau de la requête à travers son expansion ou par une restriction du domaine de recherche de l'information, soit par une déduction à partir des mots clés de la requête [Coyle and Smyth, 2007]. Le contexte en recherche d'information peut tout d'abord être défini par un ensemble de termes liés sémantiquement. Ce type de contexte est utilisé principalement dans les approches d'indexation sémantique latente. Cette approche est mise en œuvre par le moteur de recherche Yahoo !. Ce dernier offre la possibilité aux utilisateurs de restreindre la recherche à certaines parties des pages Web et le moteur réutilise ensuite l'information de ces parties pour extraire des termes proches sémantiquement de ceux de la requête.

Les systèmes dédiés à un domaine ou une discipline précise comme Google scholar [1] ou Citeseer [2] pour les publications scientifiques s'appuient sur l'homogénéité du contexte pour lever les ambiguïtés que pourraient comporter les requêtes. Ces approches se basent sur une vue statique du contexte et ne tiennent pas compte des évolutions de celui-ci.

Les approches consistant à définir le contexte par l'ensemble des caractéristiques liées à l'utilisateur sont les plus étudiées. Les données relatives aux utilisateurs sont souvent collectées de manière transparente et sont le résultat de l'étude de leur comportement sur le Web. Un exemple caractéristique est celui de l'application de commerce en ligne Amazon pour la proposition de produits relatifs à ceux que le client a déjà achetés ou est sur le point d'acheter.

Le type de document recherché peut également servir à définir le contexte. Les approches exploitant cette observation se basent sur l'hypothèse que les utilisateurs cherchent souvent le même type de document. Celui-ci est repérable par sa nature (vidéo, son, texte, ...) ou le domaine de son contenu (publication scientifique, des pages personnelles, ...). Ceci donne lieu à une catégorisation des documents comme c'est le cas dans le moteur Guru-Net.

Un type particulier de contexte, orthogonal aux autres, concerne l'environnement matériel dans lequel est posée la requête. Dans ce cas le contexte est défini par des caractéristiques liées à l'environnement (comme le bruit ou la luminosité ambiante) et les caractéristiques du terminal (taille de l'écran, quantité de mémoire disponible, ...). Dans ce domaine, la communauté scientifique s'intéresse plus particulièrement à la faisabilité de la recherche d'information plutôt qu'à la pertinence des résultats des recherches.

Dans les deux sections à venir, nous détaillons les approches de recherche d'information existantes concernant la prise en compte du domaine de recherche et la personnalisation des recherches.

Recherche d'information s'appuyant sur la représentation du domaine de recherche

Les approches existantes concernant la prise en compte du domaine de recherche ont donné lieu au développement de moteurs de recherche spécifiques comme Inquirus [Glover et al., 1999] qui proposent explicitement à l'utilisateur de choisir entre plusieurs domaines. Ces outils logiciels fonctionnent, en général, sur des techniques d'apprentissage comme dans [Chakrabarti et al., 1999]. Le principe repose sur la construction d'un index de documents appartenant à un même domaine. Les documents de l'index sont sélectionnés sur le Web par des crawlers entraînés, suivant ces techniques d'apprentissage sur des corpus de documents d'un domaine donné. L'apprentissage supervisé combiné à des modèles probabilistes est également appliqué pour former des clusters de documents. Oyama et al. [Oyama et al., 2004] proposent une approche basée sur des techniques d'apprentissage pour construire un ensemble de mots clés relatifs à un domaine. Les mots clés sont déterminés grâce à l'étude des documents du domaine. Ils sont ensuite exploités par les applications pour faire le rapprochement avec les termes de la requête pour déterminer le domaine visé par la requête.

Une autre technique pour la prise en compte du domaine de recherche consiste à utiliser des outils linguistiques, comme des ontologies ou des thésaurus, qui représentent les connaissances d'un domaine. L'exploitation la plus courante de ces outils se fait à travers des techniques d'expansion de requêtes (voir section 1.3.2) afin de faire explicitement référence au domaine de recherche dans la requête. D'autres approches exploitent les ontologies représentant le domaine pour extraire les informations pertinentes de pages annotées ou pour annoter des pages. Dans

1. http ://scholar.google.com/
2. http ://citeseer.ist.psu.edu/

l'approche proposée par Alani et al. [Alani et al., 2003], les auteurs utilisent une ontologie pour représenter le domaine de recherche de la peinture et l'exploitent afin de construire une base de connaissance sur ce domaine. Des parties précises des pages Web, supposées plus riches en information, sont visées en priorité pour l'extraction des données constituant la base de connaissance. L'ontologie représentant le domaine peut être proposée à l'utilisateur au moment où il construit sa requête pour qu'il choisisse les mots clés les plus pertinents. Dans l'approche de Sizikov et Soshnikov [Sizikov and Soshnikov, 2002], la recherche d'information est effectuée sur les annotations embarquées dans les pages Web. L'ontologie permet de définir quelles metadonnées sont relatives au domaine représenté par l'ontologie. Enfin, Vallet et al. [Vallet et al., 2007] proposent une méthode basée sur les ontologies combinant le domaine et les préférences de l'utilisateur. L'ontologie représentant le domaine est dynamique et sert à définir quelle partie du profil utilisateur utiliser pour la recherche d'information.

L'utilisation de patterns spécifiques pour faire explicitement référence au domaine de recherche a également été étudiée. Cette idée a notamment été exploitée par Brin dans ses travaux [Brin, 1998] effectués dans le domaine de la recherche bibliographique. Il définit un ensemble de pattern (ici des couples (titre, auteur)) et collecte des pages Web mentionnant ces ouvrages. Puis, il analyse les documents récupérés pour déduire d'autres patterns, plus riches, pour compléter sa recherche bibliographique sur ces ouvrages. Toujours à base de patterns, l'approche proposée par Kudelka et al. [Kudelka et al., 2007] fonctionne en deux temps. Dans un premier temps, un ensemble de patterns est défini. Ceux-ci sont considérés comme étant caractéristiques des documents du domaine. Ces patterns sont ensuite réutilisés par un crawler pour identifier les documents du domaine les contenant.

Personnalisation de la recherche

Une approche intelligente de la recherche d'information consiste à tenir compte des caractéristiques des utilisateurs afin de personnaliser la recherche. Dans ce sens, plusieurs aspects sont à considérer parmi lesquels :

- La construction d'un modèle de l'utilisateur pour représenter ses caractéristiques, ses intérêts, ses connaissances un peu à la manière des modèles utilisateur des hypermédia adaptatifs (voir section 1.2.1). Ceci sous-entend à la fois la détermination du contenu de ce modèle et sa représentation.
- L'utilisation des informations contenues dans le modèle d'utilisateur afin de retourner les documents attendus par l'utilisateur. A ce sujet, deux types d'approches ont été identifiées dans [Pitkow et al., 2002] : l'expansion de requêtes ou le reclassement (en anglais re-ranking) des pages retournées.

Les approches utilisant la personnalisation de la recherche d'information se distinguent les unes des autres sur ces deux points et s'appuient, en règle générale, sur des modèles cognitifs pour représenter la sémantique des informations représentées dans le modèle d'utilisateur.

Les modèles utilisateurs (ou profils des utilisateurs) les moins élaborés sont généralement construits manuellement et représentés par un ensemble $\{(T, V)\}$ ou T et V représentent respectivement un terme et un coefficient mesurant l'intérêt de l'utilisateur pour le concept T. Dans [Teevan et al., 2005], les auteurs en proposent une variante. Le profil est construit automatiquement à partir de l'analyse d'un corpus de documents manipulés par l'utilisateur (e-mail, pages Web, ...). Dans leur approche, le profil est un vecteur R dont chaque composante r_i représente le nombre d'occurrences du terme i dans le corpus. Le vecteur R est utilisé pour classer les résultats de la recherche.

Qiu et Cho proposent dans leurs travaux [Qiu and Cho, 2006] une approche probabiliste de la personnalisation des recherches. Les auteurs introduisent la notion d'intérêt implicite dans le profil d'un utilisateur. Cette notion est déduite du comportement des utilisateurs, en particulier des documents visités par ceux-ci. A partir des informations collectées, le système associe une probabilité à chaque document. Cette mesure de l'incertitude représente l'intérêt potentiel qu'a un utilisateur pour le document. Les documents ayant la probabilité la plus forte sont ainsi retournés.

L'approche décrite dans [Tanudjaja and Mui, 2002] varie des autres car celle-ci intègre au niveau du profil des utilisateurs la notion de désintérêt d'un utilisateur pour un concept. Le profil est représenté sous la forme d'un graphe coloré où la couleur des sommets du graphe représente le degré d'intérêt ou de désintérêt de l'utilisateur déterminé de manière semi-automatique après intervention de l'utilisateur. Les arcs du graphe représentent à l'instar d'une ontologie les relations entre les termes des sommets. Le profil est ensuite réutilisé à des fins de reclassement des résultats.

Les modèles de représentation des connaissances que sont les ontologies sont également utilisées dans certaines approches pour la construction du profil des utilisateurs [Gauch et al., 2003, Sieg et al., 2007]. Dans cette approche, le profil d'un utilisateur est une ontologie dont les concepts sont pondérés automatiquement grâce à l'observation de la navigation de celui-ci. Les poids sont ensuite réutilisés pour reclasser les résultats. Dans [Yong and Guan-yu, 2007], les auteurs ont construit un moteur de recherche personnalisé basé sur les ontologies.

Plus récemment, des approches utilisant les folksonomies pour personnaliser la recherche ont été développées [Noll and Meinel, 2007, Xu et al., 2008, Vallet et al., 2007]. Ces approches reposent sur l'idée que les folksonomies contiennent un ensemble d'information relatives aux différents utilisateurs, notamment concernant leurs centres d'intérêts et l'exploitation de ces informations permet d'améliorer la pertinence des résultats. Noll et Meinel [Noll and Meinel, 2007] utilisent une approche vectorielle pour classer les résultats d'une recherche. Les auteurs calculent une similarité entre les documents et le profil des utilisateurs qui est, en fait, constituer des données contenues dans les folksonomies. Dans [Nauman and Khan, 2007] les folksonomies sont utilisées pour filtrer davantage les résultats afin d'éliminer les pages considérées comme non pertinentes.

La construction des profils est très importante. Toutefois, leur évolution l'est également pour ne pas compromettre leur exploitation par les applications. L'évolution des profils désigne leur adaptation à la variation des centres d'intérêt des utilisateurs qu'ils décrivent, et par conséquent, de leurs besoins en information au cours du temps. Peu de travaux ont exploré le problème de l'évolution du profil de l'utilisateur sous l'angle de la dimension temporelle. L'évolution est davantage abordée comme un problème de représentation de la diversité des domaines d'intérêt de l'utilisateur.

Un premier type d'approche utilise des techniques de classification pour représenter les domaines d'intérêt. Mizzaro et Tasso [Mizzaro and Tasso, 2002] utilisent ces techniques pour classifier les centres d'intérêt des utilisateurs. Dans leur approche les auteurs distinguent les intérêts à court terme et à long terme.

Dans les travaux de Chen et Sycara [Chen and Sycara, 1998], l'évolution des profils des utilisateurs est basée sur des heuristiques liées la notion de cycle de vie artificielle d'un centre d'intérêt. Les heuristiques fonctionnent à partir des informations fournies par l'utilisateur après coup. Le système demande à l'utilisateur d'évaluer la pertinence des pages renvoyés.

Dans [Tamine et al., 2007], le procédé d'évolution du profil repose sur l'interaction entre ses dimensions, sans utilisation d'autres ressources telles que des ontologies ou des classifieurs de concepts. Une méthode statistique est déployée pour cela afin d'évaluer, au cours du temps, la

corrélation entre les contextes associés à différentes sessions de recherche.

Comme le montre notre étude, les approches pour la personnalisation de la recherche d'information favorisent les modèles cognitifs pour représenter le profil des utilisateurs. La principale difficulté réside dans la construction et la gestion de l'évolution des profils. Les méthodes utilisées dans cet objectif sont soit automatiques soit semi-automatiques. Les approches automatiques consistent en une analyse statistique des documents consultés par un utilisateur alors que les approches semi-automatiques reposent sur l'interrogation des utilisateurs pour définir si un concept est intéressant pour lui ou non. Ce type de procédé soulève des problèmes éthiques car les utilisateurs ne sont pas toujours au courant des informations collectées par les applications les concernant.

Méthodes pour le classement des résultats d'une recherche

La pertinence des résultats d'une recherche s'observe lors du classement des pages retournées par les moteurs de recherche. Les techniques de classement ont été largement étudiées dans la littérature. Les approches existantes se répartissent en deux grandes catégories : celle focalisant sur la structure du Web en particulier les hyperliens reliant les pages entre elles et celle utilisant le contenu des pages jugées pertinentes.

L'algorithme du PageRank proposé par Page et Brin [Page and Brin, 1998] mis en œuvre dans Google est incontestablement le plus connu. Cet algorithme fonctionne sur le principe de popularité d'un document, en conséquence plus une page Web est pointée par d'autres pages, plus son score est élevé et la page apparaîtra parmi les plus pertinentes. Cependant, le PageRank ne tient pas compte de la sémantique du contenu des pages Web. L'étude menée par Dhyani et al. [Dhyani et al., 2002] distingue deux grandes catégories d'approches de classement des résultats pour définir la pertinence et la qualité des pages, à savoir celle exploitant les relations (sémantiques ou physiques) existantes entre les pages et celle considérant les pages indépendamment les unes des autres.

Dans beaucoup d'approches, le Web est vu comme un graphe dont les sommets sont les pages Web et les arêtes les hyperliens pointant d'une page à l'autre [Kleinberg et al., 1999]. L'exploitation de cette structure a donné lieu à de nombreuses métriques pour le classement des résultats d'une recherche parmi lesquelles le PageRank et toute une variété d'approches s'inspirant de cet algorithme. En outre, dans [Yuwono and Lee, 1996], les auteurs utilisent le modèle booléen pour déterminer l'ensemble des pages pertinentes. Ils exploitent la structure de graphe du Web pour classer les résultats en fonction de la longueur des chemins (nombre minimal d'hyperliens reliant les pages) entre les pages représentant les sommets du graphe. Dans [Marchiori, 1997], Marchiori émet l'hypothèse que si deux pages p_1 et p_2 pointent vers la même page q, alors p_1 et p_2 partagent des sujets en commun. Il exploite cette idée pour classer les pages pertinentes. L'approche proposée par Bollen et al. [Bollen et al., 2005] s'appuie sur une idée défendue dans le domaine de la bibliométrie pour mesurer l'impact d'un journal scientifique. Dans leur approche, les auteurs considèrent un site Web comme un journal et exploite le nombre de citations (le nombre de fois où le site est référencé par d'autres) pour classer les pages.

L'autre famille d'approches considère les pages indépendamment les unes des autres. Une première technique a été développée par Lee et al. [Lee et al., 1997] et s'appuie sur un modèle vectoriel. Un poids est attribué à chaque page en fonction de la mesure de l'angle entre le vecteur construit à partir de la requête et celui construit sur la page Web. Les pages Web sont ensuite classées par rapport à ce poids. Le poids le plus important correspond à la page la plus perti-

nente. Zhuang et Cucerzan [Zhuang and Cucerzan, 2006] utilisent une mesure de la pertinence des pages Web basée sur l'étude des fichiers de logs contenant des informations sur l'historique de navigation de l'utilisateur pour classer les pages.

Le sens des données d'une page Web est rarement pris en compte dans les différentes métriques. Les approches existantes dédiées à la mesure de la pertinence des pages Web s'appuient le plus souvent sur une analyse syntaxique des documents ou sur les relations liant les documents entre eux. C'est à partir de cette observation que la communauté scientifique a défini une nouvelle famille de méthodes. Une approche originale a été proposée par Massa et Hayes [Massa and Hayes, 2005]. Les auteurs prennent en compte la sémantique des hyperliens dans le calcul de la mesure de pertinence d'une page Web. Cette mesure permet de gommer les imperfections du PageRank de Google et notamment les créateurs de pages Web introduisant énormément de liens pointant vers leur page afin d'augmenter leur score dans Google. La mesure ainsi introduite tient donc compte de l'intention des utilisateurs. Cependant, la popularité des algorithmes de classement employés dans les moteurs de recherche usuels et notamment le PageRank focalise les efforts de la communauté scientifique autour des approches s'appuyant sur la structure du Web. Les approches s'appuyant sur la sémantique du contenu sont quelque peu délaissées.

Outils existants pour la recherche d'information sur le Web

Les outils existants pour la recherche d'information sur le Web sont avant tout les moteurs de recherche. Ceux-ci se différencient sur plusieurs aspects parmi lesquels on retrouve la quantité de pages Web indexées, l'algorithme de classement des résultats utilisé, l'expressivité et la nature du langage de requête offert. Les études comparatives des principaux moteurs de recherche sur le Web sont légions. Une étude significative se trouve à l'URL `http://www.infopeople.org/search/chart.html`.

Un type de moteur particulier est le métamoteur qui fonctionne au dessus des moteurs de recherche classiques. Les métamoteurs se présentent à l'utilisateur comme des moteurs usuels mais leur fonctionnement est différent dans la mesure où la requête saisie par l'utilisateur est propagée à un ensemble de moteurs usuels de recherche classique. Les résultats sont ensuite fusionnés et reclassés [Liu et al., 2008] par le métamoteur.

Il existe d'autres types d'outils pour rechercher des informations sur la toile. Chaque outil a ses propres spécificités. Les études de Huang [Huang, 2000] et de Haav [Haav and Lubi, 2001] proposent une comparaison de ces outils.

1.3.2 Langages de requêtes pour le Web et techniques d'expansion de requêtes

La requête est certainement l'élément principal dont dépend le succès d'une recherche d'information sur le Web. Les langages de requête et les techniques permettant d'améliorer la qualité des requêtes sont donc déterminants dans cette entreprise. Les requêtes du Web sont classées en trois grandes catégories. Les *requêtes d'information* posées avec l'intention d'obtenir des informations sur un sujet précis. Les *requêtes pour la navigation* construites dans le but d'obtenir l'URL d'un site Web. Les *requêtes transactionnelles* destinées bien souvent à l'achat de produits sur le Web. Dans cette section, nous allons présenter les différents langages de requêtes existants pour le Web. Nous présentons, dans un second temps, les techniques existantes d'expansion de requêtes dont le but est d'améliorer les résultats d'une recherche sur le Web.

Les langages de requêtes

Les langages de requêtes pour le Web sont inévitablement associés à un outil de recherche. Pour cette raison, les langages des moteurs de recherche sont incontournables et forment la catégorie de langages la plus utilisée pour la recherche d'information sur le Web. Après avoir détaillé cette famille de langages, nous présentons les langages de recherche pour le Web proposés dans un autre cadre que celui des moteurs de recherche.

Langages des moteurs de recherche

Les langages de requêtes des principaux moteurs de recherche du Web fonctionnent tous suivant un modèle booléen (i.e. la requête peut s'apparenter à une expression booléenne) et offrent une syntaxe intuitive afin de toucher une grande variété de catégories d'utilisateurs. De plus, les moteurs de recherche offrent différentes fonctionnalités aux utilisateurs pour construire leurs requêtes au niveau du langage ou au niveau de leur interface.

Les opérateurs booléens utilisables dans les requêtes ne sont pas les mêmes d'un moteur à un autre. Si tous les moteurs supportent la conjonction (avec l'opérateur + dans la plupart des cas), la disjonction et la négation (l'opérateur -) entre les termes d'une requête, certains comme Yahoo ! ou Gigablast [1] offrent l'opérateur () pour regrouper des termes et appliquer une opération booléenne sur un ensemble de termes.

Certains langages permettent également de spécifier la distance devant séparer les mots clés de la requête dans le document. Google permet l'utilisation de l'opérateur "" pour construire des expressions devant se retrouver à l'identique dans les pages Web. Exalead [2] supporte l'opérateur binaire NEAR/X où X désigne le nombre de mots dans la page séparant les deux termes de l'expression (par exemple la requête *publication NEAR/5 graphe* concerne les pages Web contenant les termes publication et graphe séparés de cinq termes).

Certains moteurs de recherche pratiquent également la lemmatisation ou la troncature des termes de la requête. Le langage de requêtes de Google offre la possibilité d'utiliser l'opérateur * pour signaler que seules les pages contenant au moins une occurrence de la chaîne de caractères concaténée à * sont recherchées. Par exemple, la requête *publi** fait référence à toutes les pages dans lesquelles figurent des mots avec le suffixe *publi* comme publier, publication, ... Une autre caractéristique de ces moteurs est de prendre en compte certains pluriels des termes de la requête.

Restreindre la recherche à certaines parties des pages Web est également une possibilité offerte par certains langages de requête. Ce mode de recherche indique dans quels éléments de la page Web les termes de la requête doivent apparaître. Les parties du document sont en fait spécifiées par l'utilisation des balises du langage HTML. Les parties visées sont le plus souvent le titre de la page, le corps, un nom de domaine, une adresse IP, etc.

Il peut être également possible de préciser les pages souhaitées en indiquant à l'aide de constructeurs spécifiques certaines de leurs caractéristiques. Certains langages de requêtes comme ceux de Yahoo !, Google ou Exalead permettent, par exemple, de définir la date de publication ou la langue dans laquelle est écrit le contenu des pages Web souhaitées. L'application va interpréter l'information de la requête référencée par ces constructeurs sur des métadonnées annotant les pages Web. Cette indication peut aussi concerner le type des documents souhaités comme des fichiers Microsoft office, des fichiers MP3 ou des fichiers vidéo encodés dans un certain format (avi, mpg, mov, etc).

Enfin, une des dernières spécificités concerne l'élimination des *mots vides* c'est à dire des termes de la requête n'apportant aucune information sur les documents recherchés. Ces types

1. http ://gigablast.com/
2. http ://www.exalead.com/search

de mots sont en règle générale les articles comme *le, la, les* ou *the* en anglais. L'utilisateur peut toutefois forcer le moteur à tenir compte de ces mots en utilisant l'opérateur booléen de conjonction (e.g. + pour Google) ou celui permettant de chercher une chaîne de caractères exacte dans les documents (e.g. l'opérateur "" de Google).

Le principal avantage offert par ce type de langage est la simplicité d'utilisation. Leur syntaxe est très intuitive et nécessite peu d'expérience pour construire des requêtes donnant des résultats exploitables. De plus, ils offrent d'autres fonctionnalités pour les utilisateurs confirmés pour construire des requêtes plus complexes afin d'améliorer la recherche. Par contre, les différences entre les langages des moteurs de recherche rendent l'utilisateur dépendant d'un moteur.

Autres langages pour le Web

Bien que les langages de requêtes des moteurs de recherche focalisent l'attention de tous les internautes, il existe néanmoins d'autres langages pour la recherche d'information sur le Web. Ces langages se répartissent en trois grandes catégories, ceux basés sur le langage SQL, ceux s'inspirant de Xquery et ceux basés sur les langages booléens des moteurs de recherche. L'utilisation d'une famille de langages donnée se fera en fonction de la nature des documents recherchés.

Les langages de type SQL sont utilisés par des applications considérant le Web comme une base de données relationnelle. C'est le cas du langage Sqeal [Spertus and Stein, 2000]. Ce langage s'appuie sur un schéma, décrivant la structure des documents du Web, du même type qu'un schéma de base de données classique et les requêtes respectent la syntaxe du langage SQL. Le langage WebDB [Li et al., 1998] s'inspire aussi du domaine des bases de données. Il offre la possibilité d'intégrer la structure du Web et de son contenu au niveau des requêtes. Il permet par ailleurs d'exprimer que les documents recherchés doivent contenir des objets particuliers comme des tableaux de valeurs, des images ou des formulaires. Le langage WebSQL [Arocena et al., 1997] fonctionne également sur le même principe mais le système qui utilise ce langage met l'accent sur l'efficacité de l'interprétation de la requête et de l'extraction des informations du Web.

Depuis l'avènement du Web Sémantique et de l'utilisation de plus en plus fréquente de XML par les utilisateurs du Web pour écrire leurs pages, une nouvelle famille de langage de requête a été proposée. Ces langages, dont la syntaxe et la sémantique sont proches de celles de SQL, permettent l'extraction des données semi-structurées. On retrouve dans la littérature les langages comme RDQL [1] et SPARQL [2] pour l'extraction de données décrites en RDF, OWL-QL [3] pour les données décrites en OWL. Plus récemment, les langages versatiles [Bry et al., 2005] permettant d'interroger aussi bien le Web que le Web Sémantique ont été développés.

D'autres langages de requête pour le Web s'inspirent de ceux proposés par les moteurs de recherche. La syntaxe intuitive de ces langages permet de construire des requêtes ayant la forme d'expressions booléennes. WeQueL [Mezaour, 2003] est un exemple de ce type de langages.

Comme le montre notre étude, les langages de requêtes pour le Web sont soit des langages permettant de construire des requêtes sous la forme d'expressions booléennes, soit des langages inspirés des bases de données, de SQL en particulier, soit enfin des langages à la XQuery pour l'extraction de données semi-structurées. La syntaxe très peu intuitive pour les usagers du Web des langages à la SQL constitue un réel obstacle à leur généralisation dans le cadre du Web. En

1. www.w3.org/Submission/2004/SUBM-RDQL-20040109/
2. www.w3.org/TR/rdf-sparql-query/
3. ksl.stanford.edu/projects/owl-ql/

outre, la difficulté pour les utilisateurs de comprendre toute la puissance des opérateurs booléens pour la construction de bonnes requêtes a une conséquence non négligeable sur la qualité des résultats d'une recherche.

Techniques d'expansion de requêtes

L'expansion ou l'enrichissement de requêtes consiste à retravailler la requête initiale bien souvent en y ajoutant des informations permettant d'obtenir de meilleurs résultats lors de la recherche. Ce procédé permet en quelque sorte de corriger certains problèmes liés aux difficultés rencontrées par les utilisateurs pour exprimer les requêtes. Les termes à ajouter aux requêtes sont déterminés de plusieurs manières différentes que nous allons détailler dans cette section. On distingue trois grandes catégories d'approches pour l'expansion de requête dans la littérature : les approches basées sur les informations obtenues par interactions avec les utilisateurs, celles utilisant l'analyse d'un corpus de documents et enfin, les techniques mettant en œuvre des outils terminologiques.

Interaction avec l'utilisateur

Ce type de technique fonctionne de manière itérative [Kelly et al., 2005]. L'utilisateur pose une requête initiale qui est évaluée par le système. Parmi les pages retournées, l'utilisateur désigne celles qu'il considère comme étant pertinentes. Le système analyse ces données, et décide quels mots clés, caractérisant les besoins de l'utilisateur, ajoutés à la requête initiale et ainsi de suite. Une alternative à ce procédé consiste à considérer les n premiers résultats renvoyés comme pertinents sans la confirmation de l'utilisateur et d'analyser ces n documents.

La difficulté dans ces approches réside principalement dans l'analyse des documents pour extraire les mots clés discriminants. L'étude de Tombros et Sanderson [Tombros and Sanderson, 1998] porte sur la qualité des pages soumises à l'utilisateur avant leur analyse. Elle démontre que l'utilisateur est plus à même de prodiguer un bon jugement sur des documents contenant tous les mots clés de la requête initiale surtout si ceux-ci sont proches les uns des autres. Les travaux de Ruthven et Lalmas [Ruthven and Lalmas, 2003] montrent que les termes extraits de documents d'un même domaine sont de meilleure qualité car ceux-ci sont analysés de manière plus fiable par les utilisateurs. Leroy et al. [Leroy et al., 2003] proposent une méthode à base d'algorithme génétique pour la sélection des termes à rajouter sur l'ensemble des documents désignés comme pertinents par l'utilisateur.

L'efficacité de ces approches varie en fonction de la qualité et de la quantité des documents retournés par le système mais aussi par rapport aux possibilités offertes aux utilisateurs pour analyser la pertinence des documents . L'utilisateur peut interagir à plusieurs niveaux. Il peut décider si tout un document est pertinent ou seulement une partie, il peut même désigner directement quels mots clés lui semblant pertinents. Le système doit ensuite lever l'ambiguïté du jugement et décider si les informations fournies par l'utilisateur sont pertinentes pour tout un domaine ou seulement pour l'utilisateur.

Sélection des mots clés à partir de corpus

Une grande famille d'approches pour l'expansion de requête s'appuie sur l'étude d'un corpus de documents pour le choix de mots clés à rajouter à la requête initiale. L'analyse du corpus est faite en utilisant des outils statistiques comme la fréquence de termes, la co-occurrence ou la distribution des termes dans le corpus.

L'approche de Kim et Choi [Kim and Choi, 1999] fonctionne sur l'hypothèse que deux termes apparaissant fréquemment sur les mêmes documents ont un lien sémantique. Leur approche exploite cette observation pour le choix des mots clés. Ces derniers sont sélectionnés suivant leur co-occurrence dans le corpus de documents.

Dans [Collins-Thompson and Callan, 2005], les auteurs appliquent des modèles de Markov sur plusieurs sources de documents afin d'en extraire les mots clés les plus pertinents à rajouter à la requête.

Fattahi et al. [Fattahia et al., 2008] proposent une méthode pour l'expansion de requête dont les termes supplémentaires sont sélectionnés dans les 800 premiers documents renvoyés par Google. Les requêtes étendues visent uniquement certaines parties des documents comme le titre ou l'adresse URL de la page.

La popularité des réseaux sociaux a inspiré Bender et al. [Bender et al., 2008] pour la proposition d'une méthode pour l'expansion de requête où le choix des mots clés est déterminé suivant l'étude des tags et des signets définis par l'utilisateur ayant soumis une requête initiale. Un graphe est construit à partir de ces informations et un algorithme en déduit les mots clés pertinents.

Même si ces approches ont montré des résultats intéressants en terme de précision et de rappel des résultats lors de l'interprétation de la requête étendue, leur efficacité est dépendante de la qualité du corpus de documents à analyser. Un des problèmes majeurs de ce type d'approche basée sur l'exploitation d'un corpus de documents réside, par ailleurs, dans le fait que la sémantique des termes de la requête n'est pas pris en compte.

Utilisation de ressources terminologiques

Une façon de corriger le problème de la sémantique des relations entre les mots clés posé par les approches présentées précédemment est d'utiliser des ressources terminologiques comme les ontologies ou les thésaurus.

L'utilisation de WordNet [Fellbaum, 1998] pour l'expansion de requête a donné lieu à de nombreux travaux. Voorhees [Voorhees, 1994] utilise WordNet pour montrer que les termes sélectionnés pour l'expansion donnent de meilleurs résultats lorsque ceux-ci sont liés par une relation lexicale à ceux de la requête initiale. De plus, son approche montre que l'expansion de requête améliore peu les résultats lorsque la requête est longue car celle-ci contient suffisamment d'information pour être discriminante.

Navigli et Velardi [Navigli and Velardi, 2003] prétendent que la sélection des termes à ajouter à la requête initiale basée sur les relations ontologiques que sont la synonymie et l'hypéronymie n'améliore pas de manière significative les résultats de la recherche. Les auteurs utilisent en plus des informations lexicographiques fournies par WordNet comme la description des concepts. L'ensemble des termes extraits est ensuite affiné par des techniques statistiques utilisant un corpus de documents.

WordNet est utilisé dans [Song et al., 2006] en complément de l'analyse statistique d'un ensemble de documents extrait à partir de la requête initiale. WordNet permet d'extraire des termes synonymes à ceux issues de l'analyse statistique de la première salve de documents retournée.

Le principal problème dans l'utilisation de ressources générales comme WordNet est que la description du domaine visé par les requêtes n'est pas suffisamment précise. Les mots clés qui en sont extraits ne sont pas suffisamment discriminants. Des approches utilisant des ontologies plus spécifiques ont alors été proposées.

Fu et al. [Fu et al., 2005] proposent une méthode s'appuyant sur une ontologie du domaine et une ontologie géographique pour la recherche d'information sur le domaine du tourisme. Dans leur approche, les requêtes sont étendues par rapport aux indices géographiques qu'elles contiennent. Les noms de lieu sont contenues dans l'ontologie géographique et les autres concepts comme «la proximité» sont contenues dans l'ontologie du domaine.

Une façon différente d'utiliser les ontologies dans le processus d'enrichissement des requêtes a été proposée par Stojanovic et al. [Stojanovic et al., 2004]. L'approche proposée se sert des ontologies pour mesurer une distance sémantique entre la requête posée et les besoins en information de l'utilisateur déterminés de manière automatique par le système à travers l'analyse du comportement de celui-ci. Un ensemble de requêtes étendues par des termes de l'ontologie sélectionnés en fonction des informations collectées sur l'utilisateur lui est proposé.

Chirita et al. [Chirita et al., 2007] combinent l'utilisation d'ontologies pour désambiguïser les mots de la requête et les informations sur l'utilisateur pour la sélection des mots clés adéquats.

Les approches utilisant les ontologies n'utilisent pas toutes la palette des relations ontologiques offerte par ces ressources, seules les relations linguistiques que sont la synonymie et l'hypéronymie sont exploitées. Par ailleurs, la pertinence des résultats retournés suivant une requête étendue par le vocabulaire d'une ontologie dépend de plusieurs facteurs. Le premier d'entre eux est la qualité de l'ontologie. La représentation du domaine par l'ontologie doit être cohérente, à jour, précise et non-ambiguë. Dans le cas contraire, les mots clés extraits et rajoutés à la requête ne donneront pas les résultats escomptés. Ensuite, l'utilisateur doit être familier avec l'ontologie utilisée. Enfin, la taille de l'ontologie peut jouer un rôle important dans l'efficacité et la précision des algorithmes d'extraction car les termes de la requête doivent être mis en relation avec les concepts de l'ontologie.

De manière générale, l'utilisation de techniques d'expansion de requête est destinée à augmenter la précision des résultats retournés au détriment du rappel. En favorisant la précision, des résultats pertinents qui répondent aux besoins des utilisateurs ne lui sont pas retournés. Il est nécessaire pour les applications de trouver un compromis entre précision et rappel pour optimiser la qualité des résultats.

1.4 Synthèse

Dans cet état de l'art nous avons présenté un ensemble de méthodes représentatives des différents types d'approches des problèmes abordés dans cette thèse, comme la représentation, l'évolution et l'exploitation des données du Web.

Notre étude a montré que les données du Web étaient représentées sous différentes formes à l'aide de plusieurs standards. Si HTML reste le principal langage pour la représentation du Web, l'utilisation de XML est de plus en plus fréquente ce qui force les données à être mieux structurées. L'apparition du paradigme du Web Sémantique a accéléré l'utilisation d'un nouveau type de données, les métadonnées, pour annoter le contenu du Web et en faciliter l'exploitation en rendant les ressources du Web compréhensibles par les machines. La sémantique de ces métadonnées est représentée à l'aide de modèles de représentation des connaissances, les ontologies. Ces modèles de données ont donné lieu à de très nombreux travaux traitant des problèmes de leur construction à travers des méthodologies supportées par des outils, et de leur représentation avec la définition de plusieurs standards comme RDFS ou OWL. Néanmoins, si la plupart des

éléments sont réunis pour la réalisation de la vision du Web Sémantique, l'annotation des pages Web en métadonnées issues des ontologies est très lente et vu la quantité de pages Web existante il est difficile de croire que la totalité des pages sera un jour annotée. L'exploitation des technologies du Web Sémantique doit donc se faire d'une manière différente.

L'aspect hautement dynamique du Web oblige la communauté scientifique à traiter les problèmes liés à l'évolution de ses données. Comme le montre notre étude, l'évolution des données du Web est traitée à différents niveaux. Tout d'abord des approches comme celles proposées dans le cadre du Web réactif agissent directement au niveau des données du Web pour garantir la cohérence du contenu des pages Web. En outre, l'évolution des métadonnées est traitée principalement à travers l'évolution des ontologies. Dans ce sens, des méthodologies pour la gestion de l'évolution des ontologies ainsi que des approches pour la gestion du versioning des ontologies ont été proposées. L'étude de ces approches pour la gestion de l'évolution des ontologies a souligné plusieurs lacunes. La première d'entre elles est reliée directement à la caractérisation de l'évolution. Aucune des approches étudiées n'essaye de comprendre le phénomène de l'évolution des connaissances à travers le temps pour en extraire des informations qui faciliteraient la gestion de l'évolution des ontologies. De plus, aucune caractéristique propre au phénomène d'évolution n'apparaît au niveau de la représentation des ontologies. Ce manque de réflexion autour de la nature de l'évolution empêche également la définition de stratégies pour l'évolution d'ontologies qui irait dans le sens d'une gestion automatique du processus d'évolution et permettrait le développement d'un outil support pour l'évolution d'ontologies.

Enfin, l'exploitation des données du Web se fait principalement au travers de la recherche d'information. Des avancées significatives ont été proposées dans ce domaine notamment par la prise en compte du contexte de la recherche c'est-à-dire les caractéristiques de l'utilisateur ou le domaine de recherche visé par la requête. Notre étude montre que le contexte est exploité principalement dans deux catégories d'approches pour améliorer la pertinence des résultats. La première consiste en l'utilisation du contexte pour classer les résultats d'une recherche par ordre de pertinence. La seconde s'intéresse à l'utilisation du contexte pour l'expansion de requêtes, les termes ajoutés à la requête initiale ayant un effet filtrant sur les documents hors contexte. Dans cette deuxième catégorie d'approches, les technologies du Web Sémantique comme les ontologies sont utilisées pour représenter le contexte. Un gros manque dans la recherche d'information basée sur le contexte réside dans la non prise en compte de l'évolution de ce contexte au niveau de la recherche, aussi bien dans le reclassement des résultats pertinents que dans l'expansion des requêtes.

Nos travaux visent à répondre aux différentes lacunes précédemment identifiées. Tout d'abord, nous proposons un modèle d'ontologies, *les ontologies adaptatives*, capable de s'adapter aux évolutions des connaissances du domaine. Ce modèle qui est le fruit de réflexions sur le phénomène d'évolution des connaissances, prend en compte les caractéristiques propres à l'évolution. Nous montrons comment représenter les ontologies adaptatives avec les langages ontologiques présentés dans ce chapitre. Nous proposons également un processus général pour l'adaptation des ontologies construites selon notre modèle et un outil pour l'assistance à l'évolution d'ontologies.

Dans un deuxième temps, nous exploitons les caractéristiques des ontologies adaptatives afin de pallier aux manques mis en évidence par notre étude concernant l'exploitation des données du Web dans la recherche d'information. Nous proposons tout d'abord une technique d'enrichissement du contenu des pages Web basée sur les graphes et les ontologies adaptatives. Le contenu ainsi enrichi nous permet de définir un nouvel algorithme pour le classement des résultats perti-

nents exploitant la sémantique du contenu des pages Web. Nous proposons d'exploiter l'aspect dynamique des ontologies adaptatives pour représenter le domaine de recherche visé par une requête et le profil des catégories d'utilisateurs potentiellement intéressés par les informations du domaine. Les données contenues dans ces ontologies sont ensuite réutilisées pour étendre les requêtes. Contrairement aux approches existantes évoquées précédemment, notre technique d'expansion des requêtes est basée sur une grande variété de relations ontologiques en plus des éléments des ontologies adaptatives. Les termes sélectionnés dans les ontologies adaptatives sont les plus représentatifs et surtout les plus à jour des domaines ce qui améliore la pertinence des résultats des recherches.

Enfin, nous proposons un outil pour la recherche d'information sur le Web exploitant les technologies existantes du Web classique et du Web Sémantique. Contrairement aux outils existants, l'outil que nous proposons bénéficie des caractéristiques des ontologies adaptatives, et notamment de leur faculté à s'adapter aux évolutions du domaine pour personnaliser les recherches. Nous montrons la validité de nos concepts à travers une série d'expérimentations portant sur un cas d'étude dédié à la recherche de publications scientifiques.

Chapitre 2

Un modèle d'ontologies adaptatives

Sommaire

Les diverses définitions d'une ontologie ont toutes en commun la notion de représentation d'un ensemble de connaissances d'un domaine du monde réel. L'évolution des ontologies peut donc s'apparenter à un problème d'évolution des connaissances, un sujet largement abordé dans les domaines de la philosophie, de la psychologie (ou plus généralement des sciences cognitives) ou encore des sciences naturelles et repris récemment dans les sciences de l'information et de l'intelligence artificielle.

La théorie de la connaissance ou *l'épistémologie* est une des branches les plus anciennes et des plus débattues de la philosophie. De nombreux illustres penseurs comme Aristote ou Kant ont proposé des réflexions sur ce sujet. Jusqu'au dix-neuvième siècle l'essentiel des efforts des philosophes avait porté sur la connaissance en tant qu'achevée. Ce n'est que plus récemment qu'un nouveau mouvement philosophique, *le constructivisme*, fondé sur la vision Kantienne de la connaissance, selon lesquelles la connaissance des phénomènes résulte d'une construction, a vu le jour. Cette nouvelle façon d'envisager la connaissance amène intuitivement à la notion d'évolution des connaissances.

C'est sur ces bases que le psychologue Jean Piaget développa dans les années 1920 la notion *d'épistémologie génétique* dans laquelle la connaissance est envisagée non plus comme figée mais évolutive et ayant la capacité de s'adapter à travers un processus de construction permettant à l'individu d'acquérir la connaissance mais également de la faire évoluer afin d'atteindre «la connaissance supérieure». Pour élaborer les théories fondatrice de l'épistémologie génétique et

principalement celle mettant en avant l'acquisition et l'adaptation des connaissances au cours du temps, Piaget s'appuya principalement sur l'observation du comportement de jeunes individus dans diverses situations.

Le développement récent des sciences de l'information a également nécessité l'utilisation de connaissances à des fins diverses et variées. Les systèmes à bases de connaissances, comme les systèmes experts par exemple, sont des outils d'aide à la décision s'appuyant sur un ensemble de faits représentant la connaissance du système et des règles d'inférence. Cependant, les problèmes liés à l'évolution des connaissances ont très peu été étudiés dans ces domaines de plus les idées évoquées précédemment et développées en psychologie ou en philosophie n'ont pas été approfondies suffisamment pour être appliquées à l'évolution des connaissances dans les sciences de l'information.

Les idées développées dans les domaines de la psychologie et de la philosophie sont toutes basées sur l'observation du phénomène de l'évolution et des conséquences de celui-ci sur les connaissances du domaine auquel elles se rapportent. Cependant, l'absence de preuves formelles démontrant de manière irréfutable ces phénomènes illustre toutefois les difficultés de la communauté scientifique pour caractériser la notion d'évolution des connaissances, ce qui par conséquent laisse la porte ouverte à la discussion et à la remise en question des idées proposées dans les domaines évoqués précédemment. Néanmoins, toutes ces théories, largement acceptées par la communauté scientifique, restent encore inappliquées au problème de l'évolution des ontologies. Elles méritent d'être approfondies pour être adaptées à celui-ci, ce qui permettra de combler certaines des lacunes mises en évidence dans le chapitre précédent.

2.1 Motivations

Les manques mis en évidence tout au long du premier chapitre concernant l'absence de caractérisation de l'évolution des connaissances, constituent la première motivation pour la proposition d'un modèle d'ontologie ayant la capacité de s'adapter aux évolutions d'un domaine. La construction d'un tel modèle repose d'une part sur la compréhension des mécanismes de l'évolution et d'autre part sur l'adaptation des idées développées dans les domaines évoqués au cours de l'introduction de ce chapitre au problème d'évolution d'ontologies.

Le modèle d'ontologies adaptatives que nous nous proposons de construire doit servir de fondation à la création d'un processus automatique (ou semi-automatique) d'adaptation des ontologies aux évolutions successives de leur domaine respectif. Ceci permettra d'ouvrir la voie à la définition de stratégies pour l'évolution d'ontologies. Par ailleurs, ce processus d'adaptation va permettre le développement d'un outil d'aide à la gestion des ontologies.

L'objectif à plus long terme qui motive la proposition d'un modèle d'ontologies adaptatives reste l'amélioration de la recherche d'information sur le Web. Les possibilités offertes par notre modèle pour représenter les connaissances d'un domaine particulier et pour s'adapter aux évolutions de ce dernier vont être exploitées pour la recherche d'information. Les ontologies adaptatives pourront représenter le contexte de la recherche (caractéristiques du domaine de recherche et des utilisateurs), les informations relatives à l'évolution contenues dans les ontologies adaptatives seront utilisées pour enrichir les requêtes posées par les utilisateurs.

2.2 La notion de temps dans l'évolution

Avant de présenter les différents éléments constituant le modèle des ontologies adaptatives, il semble important de discuter des caractéristiques de la notion de temps dans le processus d'évolution car non seulement temps et évolution sont deux notions fortement liées, mais les caractéristiques du temps vont conditionner la modélisation des éléments propres à l'évolution. Selon le dictionnaire de référence le Petit Robert :

> *«Le temps est un milieu indéfini où paraissent se dérouler irréversiblement les existences dans leur changement, les événements et les phénomènes dans leur succession.»*

Il est également une entité abstraite représentative du changement continuel de l'Univers.

La notion de temps peut tout d'abord être considérée dans sa *durée* (chronométrie). La durée permet de considérer le temps dans sa globalité, mais aussi de le mesurer, ce qui par conséquent autorise la définition de périodes limitées comme la seconde, la minute, le siècle ... menant à la notion d'espace (ou d'intervalle) de temps. La mesure du temps a permis notamment de l'envisager comme une notion discrète où une période représente le pas de discrétisation. Plus spécifiquement par rapport au phénomène de l'évolution, la durée permet surtout de mesurer le rythme de l'évolution (i.e. la vitesse et la fréquence des changements). La fréquence et la vitesse des changements dépendent du domaine concerné. La loi de Moore, selon laquelle la puissance des microprocesseurs double tous les dix huit mois, et l'évolution géologique de la Terre extrêmement lente, prenant des millions d'années, en sont les parfaites illustrations. La loi de Moore fait, par ailleurs, référence à la notion de répétition ou de cycle dans la durée.

Ensuite le temps peut être considéré dans une *succession* (chronologie) d'espaces de temps par rapport à une origine (comme l'an zéro dans l'Histoire de l'humanité par exemple). Ce point de référence a donné lieu aux notions philosophiques que sont le passé, le présent et le futur. Cette origine est primordiale dans le phénomène d'évolution car c'est à partir d'un moment précis dans le temps (caractérisé par une *date*) que débute l'observation du domaine pour en capturer les changements. C'est à travers la succession que l'on peut ordonner l'ensemble des changements par rapport à une référence chronologique. Cette chronologie des événements permet ainsi d'éviter toutes inconsistances notamment au niveau de la représentation (abstraite ou concrète) et de la compréhension des connaissances du domaine par rapport aux différents changements ayant eu lieu.

Notre vision du temps dans le phénomène d'évolution des connaissances nous amène à considérer les aspects mentionnés ci-dessus.

Tout d'abord, dans notre vision des choses, le caractère absolu ou relatif du temps (i.e. la mesure du temps est faite par rapport à une origine unique bien définie ou selon les besoins par rapport à un événement quelconque) importe peu. La chose importante est que la chronologie soit respectée afin de distinguer l'ordre dans lequel surviennent les évolutions du domaine. Même si les domaines sont indépendants les uns des autres, les observations sur ces domaines peuvent se faire, soit par rapport à une date absolue déterminée de manière arbitraire, soit par rapport à un point de départ attribué à chaque domaine lors de leur création. Pour notre part, nous considérons une date absolue (l'année zéro de l'Histoire), ce qui permet le maintien d'une certaine cohérence avec l'évolution des connaissances des domaines du monde réel dont les changements sont référencés par convention par rapport à la date de l'an zéro. Ce choix est également influencé par des contraintes techniques, sur lesquelles nous reviendrons plus en détail dans la suite de ce mémoire, notamment dans la section 4.1 consacrée à la représentation de l'évolution au niveau de l'ontologie. Par contre, la façon dont les individus perçoivent le temps peut influencer fortement

la compréhension de l'évolution des connaissances d'un domaine. Ainsi la notion d'intervalle de temps est importante et doit être bien définie afin que chaque individu ait la même perception du temps ce qui limitera les problèmes de compréhension de l'évolution.

Ensuite, l'aspect discret ou continu du temps nous semble d'une bien plus grande importance. Dans notre approche de l'évolution des connaissances, nous considérons un temps discret. Ce choix se justifie d'un point de vue conceptuel, par les arguments suivants :

- L'évolution sera plus facile à quantifier et les changements seront davantage mis en évidence si l'observateur regarde le domaine à intervalles de temps réguliers. Ceci implique une définition rigoureuse du pas de discrétisation en fonction du domaine, comme discuté précédemment.
- L'impact de l'évolution sur les connaissances du domaine sera plus facile à mesurer. Les différentes métriques proposées à cet effet seront donc plus faciles à définir et plus faciles à comprendre sans pour autant perdre en pertinence.

De plus, considérer un temps discret offre également des avantages d'un point de vue technique parmi lesquels :

- Le caractère discret des ordinateurs facilite grandement la mesure du temps si celui-ci est discret. De plus, le stockage de la valeur associée au temps ne pourra se faire sans perte d'information seulement si cette valeur est rationnelle, ce qui sous entend une discrétisation du temps.
- Les problèmes liés à la synchronisation des données plaident également en faveur de la considération d'un temps discret. Les changements intervenant dans le monde réel lors de la mise à jour des données ne seront ainsi pas pris en compte, d'où l'intérêt d'observer le domaine à intervalles de temps réguliers et de traiter les changements survenus entre deux intervalles successifs.

Comme le temps est une notion fondamentale dans l'évolution, il en est de même pour l'évolution d'ontologie. Les caractéristiques du temps dans le processus d'évolution d'ontologie doivent favoriser plusieurs aspects parmi lesquels le suivi de l'évolution est le principal. Un temps bien défini doit permettre de mettre en valeur les nouveaux concepts apparaissant dans l'ontologie, ceux qui y persistent et la connaissance obsolète (i.e. les éléments ontologiques n'étant plus en phase avec les connaissances du monde réel).

Maintenant que la notion du temps dans notre approche a été davantage discutée, nous pouvons nous concentrer sur la définition des éléments constituant notre modèle d'ontologies adaptatives. Nous présentons tout d'abord les différents concepts.

2.3 De l'évolution des connaissances à l'évolution d'ontologies

Comme évoqué dans l'introduction de ce chapitre, le modèle d'ontologies adaptatives que nous proposons est inspiré des idées développées dans les sciences cognitives et les sciences naturelles. Les éléments de ce modèle [Guelfi et al., 2007b] sont proposés sur la base de nos observations de l'évolution des connaissances du monde réel, comme l'ont fait Piaget et les biologistes.

L'évolution d'ontologie est envisagée sous plusieurs formes dans la littérature (voir section 1.2.2). Dans notre approche, l'évolution d'ontologies est envisagée sous un aspect particulier. Nous nous proposons d'identifier les changements du monde réel puis d'adapter l'ontologie en fonction de ces changements. Dans cette partie, nous allons détailler les principales caractéristiques de l'évolution des connaissances que nous proposons de prendre en compte.

2.3.1 Emergence

Le premier phénomène à travers lequel se manifeste l'évolution des connaissances correspond à la création ou l'émergence de nouvelles connaissances. Le développement de la connaissance pour Piaget s'appuie sur des actions sensori-motrices qui sont ensuite intériorisées à travers l'accès à la fonction symbolique, c'est-à-dire à la capacité de représenter des actions ou des objets concrets par des symboles. Piaget [Piaget, 1946] montre notamment que le passage du concret à sa représentation symbolique se construit progressivement à travers différents stades caractérisés d'abord par la mise en œuvre d'opérations concrètes, puis par celle d'opérations abstraites faisant appel à des représentations formelles. Les études conduites par Piaget, à la base des théories de l'apprentissage, permettent de décomposer le processus de création des connaissances en plusieurs étapes :

1. La première phase consiste en *l'observation* d'un phénomène encore inconnu dans le domaine d'étude à un moment précis dans le temps.
2. *L'assimilation* correspond à l'incorporation d'un objet ou d'une situation à la structure d'accueil du sujet (structure d'assimilation) sans modifier cette structure, mais avec transformation progressive de l'objet ou de la situation à assimiler. Le sujet transforme les éléments provenant de son environnement pour pouvoir les incorporer à sa structure d'accueil. Par exemple, un enfant en bas âge sait comment saisir son hochet préféré avec les doigts d'une main et le lancer pour qu'il fasse du bruit. Quand il découvre un nouvel objet, comme la fragile montre de son père, il transfère sans problème ce schéma moteur connu au nouvel objet et l'envoie rebondir sur le plancher.
3. *L'accommodation* : lorsque l'objet ou la situation résistent, le mécanisme d'accommodation intervient en entraînant une modification de la structure d'accueil de l'individu, de manière à permettre l'incorporation des éléments qui font l'objet de l'apprentissage. Dans ce cas, le sujet est transformé par son environnement, ce qui correspond au processus inverse de l'assimilation. Par exemple, si le même enfant rencontre maintenant un ballon de plage, il va essayer de le saisir comme il le fait pour son hochet avec une seule main. Très vite, il va se rendre compte que ce procédé ne fonctionne pas et découvrira éventuellement comment tenir le ballon entre ses deux mains.

Remarque : Le passage constant entre les phases d'assimilation et d'accommodation correspond à la phase *d'adaptation*.

Le mouvement constructiviste, auquel Piaget est rattaché, définit donc le développement de la connaissance à travers l'observation et l'expérimentation. Par opposition, pour les béhavioristes, le modèle de base de l'apprentissage repose sur l'idée que : à une stimulation de l'environnement, le sujet réagit par des comportements considérés uniquement dans leur aspect observable en relation avec des conditions externes comme le rappellent les psychologues Bernadette Aumont et Pierre-Marie Mesnier. Sans nier la réalité que constitue l'individu et tout ce qui s'y passe, les béhavioristes (classiques) ne s'en occupent pas directement. Ce qui les intéresse, c'est de spécifier, sans référence aux variables internes non observables et hypothétiques, les conditions et les processus par lesquels l'environnement contrôle le comportement. Le schéma selon lequel ils travaillent met entre parenthèses l'individu qu'ils considèrent comme une «boîte noire». En particulier, ils laissent de côté toutes les questions relatives à la conscience. Cette théorie béhavioriste a été remise en cause à plusieurs reprises, notamment par Piaget, qui a démontré qu'on ne pouvait pas résumer l'intelligence à des phénomènes d'apprentissage et d'imitation sur le modèle de l'éthologie animale sans tenir compte de la manière dont la connaissance se construit chez un

sujet et un groupe. Comme la connaissance n'est pas un phénomène observable, le béhaviorisme ne s'est pas engagé dans la problématique de l'épistémologie.

Si les différents mouvements s'opposent sur certains aspects de la création de la connaissance, la phase d'observation des changements est commune à tous les modèles. L'ordre précis dans lequel sont survenus les changements est important à prendre en compte pour notre problème d'évolution d'ontologies puisqu'il va déterminer ensuite les phases d'assimilation et d'accommodation dans lesquelles la connaissance émergente sera caractérisée. Ceci nécessite de rattacher chaque connaissance émergente à un point précis du temps.

Le phénomène d'émergence des connaissances est également transposable au problème d'évolution d'ontologies. Afin de rester cohérent par rapport aux évolutions du monde réel, l'ontologie a besoin d'être complétée par l'ajout de nouveaux concepts d'où l'émergence de nouveaux éléments dans l'ontologie.

2.3.2 Suppression

Le phénomène de création des connaissances trouve son pendant dans le processus de suppression. Plusieurs approchent émanant de plusieurs domaines différents tentent d'expliquer ce phénomène.

Tout d'abord, la connaissance précédemment créée peut être erronée. En vertu du principe de révision des croyances mentionné au chapitre précédent (voir section 1.2.2), la connaissance peut être remise en question par des nouvelles informations. Les connaissances considérées comme établies jusqu'alors sont reconnues comme fausses et par conséquent sont soit supprimées définitivement, soit remplacées par de nouvelles. Par exemple le fait de considérer la Terre comme plate a été remis en question puis supprimé lorsque les scientifiques ont apporté la preuve du contraire.

En psychologie, la suppression des connaissances s'explique par le principe d'obsolescence. Une connaissance peut devenir obsolète parce que le domaine courant est totalement diffèrent du domaine initial dans lequel elle a été créée. Certaines connaissances ne proviennent que de l'expérience humaine (cf. béhaviorisme). Comme ils sont hautement situés, ils sont habituellement trop complexes pour être décrits en termes simples. Les représentations formelles sont rapidement limitées. Ces connaissances peuvent disparaître lorsque les gens qui les possèdent disparaissent. D'autres peuvent être formalisables mais si elles ne sont pas utilisées pendant une période de temps suffisamment longue, elles sont progressivement oubliées, et sont éventuellement remplacées par d'autres. Les principes d'assimilation et d'accommodation peuvent également conduire à une destruction d'une connaissance principalement si la connaissance assimilée est fausse. Cette connaissance sera supprimée à travers le processus d'accommodation.

En science de l'information, la suppression de connaissances peut être la conséquence d'une limitation de capacité. La taille de l'espace mémoire bien souvent limitée dans la plupart des systèmes d'information nécessite de faire un choix dans les données et les connaissances à conserver lorsque cette limite est atteinte ce qui implique la destruction des connaissances superflues. Le choix dans les données à conserver fait apparaître la notion d'importance puisque dans la plupart des cas, seules les données les plus importantes seront conservées. Nous reviendrons plus longuement sur ce point à la section 2.3.4.

Concernant l'évolution des ontologies, la suppression des connaissances se manifeste à travers la suppression d'un ou plusieurs éléments de base de l'ontologie comme des concepts, des

relations ou encore des instances de concept.

2.3.3 Abstraction et spécialisation

Les connaissances d'un domaine sont parfois remplacées par d'autres dans le but de faciliter la compréhension de celui-ci ou d'augmenter la précision de sa description. Ce type de phénomène consiste en l'abstraction, dans le cas où la connaissance initiale fait place à une connaissance plus générale, et en la spécialisation, dans le cas contraire où les connaissances deviennent plus précises. Dans ses travaux [Piaget, 1977] sur les phénomènes d'abstraction, Piaget en distingue deux types :

- *L'abstraction empirique* est un processus qui consiste à ne retenir qu'une seule des multiples propriétés ou régularités intrinsèques à un objet (le corps propre compris) ou à un événement actuellement considéré par le sujet (par exemple la couleur d'un objet, la relation de succession qui unit deux événements, la forme globale d'un objet ou encore la relation de grandeur entre deux objets). L'abstraction empirique est le fruit d'une interaction directe entre le sujet et son environnement. En observant ou en agissant, le sujet collecte des informations sur les caractéristiques du milieu dans lequel il est plongé. Ces informations sont acquises à partir de la manipulation ou de l'observation des objets. Contrairement aux conceptions empiristes de la connaissance, l'abstraction empirique ne suffit pas à elle seule à rendre compte de l'origine des connaissances empiriques, et encore moins de celle des connaissances logico-mathématiques.
- *L'abstraction réfléchissante* est l'un des mécanismes centraux par lesquelles les formes ou les structures logico-mathématiques de la connaissance sont construites progressivement par le sujet. Elle est composée de deux processus. Le premier consiste à réfléchir (au sens quasi optique du terme) sur un nouveau plan, une organisation d'action ou de pensée préalablement construite (le sujet constate ou décrit par exemple les coordinations logiques d'une action composée qu'il vient de réaliser). Ce processus s'apparente à ce qui se passe sur le plan de l'abstraction empirique (ou pseudo-empirique), sauf que l'ordre est alors extrait des actions ou des opérations que le sujet a précédemment acquises. Ce premier processus de réflexion dépend par ailleurs des formes ou des concepts logico-mathématiques que le sujet est en train de construire sur ce nouveau plan, construction qui fait alors intervenir une activité de réflexion intellectuelle composée de régulations et d'activités logiques diverses et variées.

Les processus d'abstraction décrits par Piaget montrent bien la volonté de simplifier les choses (le fait de ne considérer qu'une seule caractéristique d'un objet par exemple) afin de les rendre plus compréhensibles. Le phénomène inverse de l'abstraction des connaissances correspond à la spécialisation. Ce procédé consiste à raffiner une connaissance par une autre plus précise afin de compléter les connaissances d'un domaine. Le fait de raffiner une connaissance n'implique pas dans tous les cas le remplacement de cette connaissance. Mais dans le cas où la connaissance initiale est remplacée, on peut voir cette modification comme une composition de suppressions et d'émergences de nouvelles connaissances plus précises ou plus abstraites.

Le principe général d'abstraction (ou de spécialisation) ainsi décrit peut se traduire au niveau d'une ontologie soit par le remplacement (i.e. suppression puis émergence) d'un élément ontologique, soit par l'ajout de nouveaux éléments (de concepts principalement) et l'établissement de relations de subsumption entre ce nouveau concept et les concepts à abstraire ou à spécialiser. Ces modifications de l'ontologie ont pour effet de rendre le domaine plus précis dans le cas où les nouveaux éléments sont plus spécifiques ou de rendre sa description plus générale si les nouveaux

éléments sont plus généraux.

De manière plus spécifique, la description d'un concept d'une ontologie peut également être enrichie (ou au contraire appauvrie). Le phénomène d'abstraction empirique tel qu'il est décrit s'intéresse principalement aux caractéristiques ou aux propriétés des objets. Ainsi, les nouvelles connaissances obtenues suivant l'observation ou la manipulation d'un objet vont servir à enrichir la description de cet objet. En transposant ce phénomène au niveau d'une ontologie, cela nous amène à rajouter un certain nombres de propriétés ou d'attributs (suivant le modèle ontologique considéré) au concept rendant ainsi la description du domaine plus riche et donc plus précise. L'abstraction réfléchissante, quant à elle, permet principalement d'établir à partir d'une réflexion des relations entre les objets observés. Ce qui se traduit au niveau d'une ontologie par l'apparition de nouvelles relations entre les concepts ayant pour effet un enrichissement de la description du domaine.

2.3.4 Distance et poids sémantique

Pour Piaget, l'apprentissage c'est-à-dire le développement des schèmes [1] opératoires, est le résultat d'un processus dynamique de recherche d'équilibre entre le sujet et son environnement. En fait, la mise en œuvre du mécanisme d'accommodation implique :

1. qu'il y ait d'abord tentative d'assimilation de manière à ce que les structures d'accueil adéquates soient mobilisées et que les éléments qui font l'objet de l'apprentissage soient reliés à ce que le sujet connaît déjà. Ce phénomène peut entraîner un renforcement des structures existantes ;
2. que l'assimilation crée un déséquilibre qui conduise à un «conflit cognitif» ;
3. que le conflit soit «régulé» par une «rééquilibration majorante» c'est-à-dire que le déséquilibre soit réellement dépassé de sorte qu'il conduise à une nouvelle forme d'équilibre. Ce dernier correspond à un progrès réel en terme de développement cognitif se mesurant notamment par une progression au sein des stades (ou des sous-stades) de développement décrits par Piaget.

Les mécanismes régissant l'assimilation des connaissances décrivent un processus dynamique au cours duquel plusieurs notions importantes se dégagent. La première étape du processus (i.e. la tentative d'assimilation) met en évidence une «distance» existante entre les connaissances déjà assimilées et celles en cours d'assimilation. L'assimilation conduit également à un renforcement des structures existantes ce qui souligne l'importance que peut prendre certaines connaissances d'un individu. De plus, le phénomène de régulation, visant à créer un équilibre entre les structures cognitives développées chez l'individu, induit une variation de cette distance et de cette importance (i.e. poids sémantique). Dans les sous-sections suivantes, nous allons discuter plus largement de ces notions et de leur apport dans le processus d'évolution d'ontologies.

Distance sémantique

La notion de distance sémantique que nous introduisons dans cette section fait explicitement référence au lien mis en évidence dans le processus d'assimilation entre les structures cognitives existantes et la nouvelle connaissance à assimiler. Selon Piaget, la construction des savoirs dépend à la fois des structures cognitives préexistantes du sujet par l'assimilation au moi et des

1. Le schème est une structure ou organisation des actions telles qu'elles se transforment ou se généralisent lors de la répétition de cette action en des circonstances semblables ou analogues

objets perçus dans l'environnement par l'accommodation aux choses. Dans chaque situation rencontrée par le sujet, il se doit d'appliquer un programme général à cette situation particulière afin d'adapter les nouvelles connaissances à celles déjà assimilées.

L'autorégulation entre les structures cognitives existantes du sujet et la transformation de ces structures pour l'adaptation à des situations extérieures s'appelle équilibration. Il s'agit d'un état dynamique qui réunit assimilation et accommodation. Dans ce processus, la nouvelle connaissance (ou les nouveaux schèmes) sont reliés progressivement aux schèmes déjà assimilés, ce qui montre le rapprochement ou l'éloignement (dans le cas où la connaissance assimilée doit être modifiée) des connaissances entre elles, d'où la variation de la distance sémantique entre les connaissances. Si les nouvelles connaissances ne peuvent pas être assimilées parce qu'elles ne correspondent à aucune structure cognitive préexistante, il en résulte une situation de déséquilibre qu'il s'agit de rééquilibrer. Le sujet essaiera d'ajuster le nouveau objet en transformant ses structures cognitives afin de s'adapter au milieu. Le sujet s'ajuste en fonction des caractéristiques de l'objet et de la situation. L'accommodation correspond à une modification de l'organisme pour s'adapter aux conditions extérieures, le processus d'accommodation sert à enrichir ou élargir un schème d'action en le rendant plus flexible. L'accommodation est le processus inverse de l'assimilation, c'est-à-dire il faut changer sa structure cognitive pour intégrer un nouvel objet ou un nouveau phénomène. L'apprentissage se fait en mettant en jeu à la fois les structures mentales et l'expérience.

La distance sémantique mis en évidence à travers le processus d'assimilation peut également être considéré au niveau des ontologies d'une part dans la description d'un domaine et d'autre part dans le processus d'évolution de l'ontologie. Cette nouvelle notion, encore absente des modèles ontologiques existants, doit permettre de mesurer la distance existante entre les différents concepts d'une ontologie. De plus, cette distance doit pouvoir varier lors de l'évolution d'ontologie afin de représenter aussi fidèlement que possible l'état des connaissances du domaine. La variation de la distance fera l'objet d'une discussion plus approfondie au cours du chapitre suivant.

Poids sémantique

De la même façon, la notion de poids sémantique que nous proposons ici fait référence au renforcement des connaissances déjà assimilées au cours du processus d'équilibration. Ce renforcement intervient principalement lorsque le sujet est plongé en permanence dans le même environnement et que les informations qu'il perçoit de cet environnement ne concernent qu'un objet en particulier. L'équilibration va avoir pour effet de renforcer les connaissances du sujet sur cet objet au détriment des autres connaissances assimilées par le sujet au cours de ses expériences passées. Ceci démontre non seulement l'existence du poids sémantique assigné à chaque connaissance du sujet mais que celui-ci est amené à évoluer au cours du temps suivant l'assimilation d'autres connaissances.

Ce poids sémantique mérite d'être considéré dans le phénomène d'évolution d'ontologies, comme le suggèrent également Weichselbraun et al. dans leurs travaux [Weichselbraun et al., 2007]. Ce poids doit permettre de mesurer l'importance de chaque concept dans l'ontologie. Suivant le niveau d'abstraction de représentation des connaissances d'un domaine, le poids sémantique peut être considéré de manière particulière, c'est-à-dire individuellement pour chaque concept ou de manière plus générale, c'est-à-dire en considérant la somme des poids de plusieurs concepts directement reliés entre eux. La variation de ce poids est discutée plus largement au chapitre suivant.

2.3.5 Résistance aux changements et persistance

Les phénomènes de construction de la connaissance décrits par Piaget mettent en évidence deux autres aspects de l'évolution de la connaissance : la persistance et la résistance. Ces aspects et les relations qui les relient sont également discutés dans la théorie de l'évolution de Charles Darwin.

Résistance aux changements

En termes de représentations et d'idées, la résistance au changement est avant tout la conséquence d'un mécanisme désigné par les psychologues comme la dissonance cognitive. Le fondement théorique nous dit qu'il est difficile pour l'être humain d'accepter une chose et son contraire, de faire siennes deux idées qui s'opposent. Si tel est le cas, la personne se retrouve en situation de dissonance cognitive, état psychologique dont le sujet cherchera à s'échapper le plus rapidement possible. Il peut aussi apparaître lorsque le sujet a un comportement ou une attitude contraire à ses valeurs ou principes et va donc montrer une résistance par rapport aux nouvelles informations perçues. Par ailleurs, s'il reçoit une information ou une idée qui contrarie les connaissances déjà assimilées, le déséquilibre cognitif est créé. La manière la plus radicale et efficace de réduire cette dissonance est de rejeter purement et simplement cette nouvelle donnée, de ne pas la croire, de l'oublier. Une deuxième méthode consiste à créer de la cohérence là où il n'y en a pas, c'est-à-dire d'ajouter entre les deux éléments incompatibles (l'idée première et l'information apportée) d'une troisième donnée qui rend pertinentes les deux premières entre elles. Une troisième possibilité consiste à adapter la pensée initiale à l'information nouvelle. Ce processus d'accommodation va prendre plus ou moins de temps suivant la résistance du sujet.

Le principe de la sélection naturelle fait également intervenir le principe de résistance aux changements. La sélection naturelle consiste en l'élimination progressive des individus inadaptés à l'environnement dans lequel ils sont plongés. Ainsi, les individus ayant su s'adapter aux évolutions de leur environnement ou ceux dont les évolutions de l'environnement n'ont pas d'emprises sur eux (i.e. les individus deviennent résistants) vont subsister.

Concernant l'évolution des ontologies, la résistance aux changements doit être prise en compte au niveau des éléments de l'ontologie afin de bénéficier d'un moyen pour limiter et contrôler les effets du changement.

Persistance

La persistance des connaissances se manifeste par la durée pendant laquelle la connaissance est présente dans la mémoire d'un individu. Cette persistance peut tout d'abord être une conséquence directe des phénomènes de résistance aux changements et de poids sémantique. En effet, plus une connaissance assimilée va résister aux changements, plus cette connaissance va persister. De même, la persistance est la conséquence d'une connaissance solidement ancrée, donc d'un poids sémantique élevé. La durée de persistance est également fonction de la stimulation de la mémoire. Le fait que la mémoire soit stimulée régulièrement va entraîner une persistance plus longue des connaissances contenues dans la partie de la mémoire qui est stimulée. C'est d'ailleurs cette vision qui a donné lieu à la notion de persistance dans le domaine de la révision des croyances.

Par ailleurs, la persistance d'une connaissance peut dépendre de la durée de validité de cette connaissance. Certaines informations ne sont valides que pour une période de temps connue à l'avance. Les connaissances relatives à ces informations sont ensuite supprimées. Par exemple,

le fait : « La date limite de soumission d'articles scientifiques à la conférence WWW 2009 est fixé au 3 novembre 2008» ne sera valide que pendant la période délimitée par la date à laquelle cette information est parvenue au sujet jusqu'à la date du 3 novembre 2008. Ainsi, la persistance des connaissances sera d'autant plus facile à gérer que les périodes de validité seront bien définies.

La notion de persistance des connaissances telle que nous venons de la décrire permet de l'envisager sous deux aspects différents dans notre problème d'évolution d'ontologies. Nous avons tout d'abord la persistance globale qui correspond à la durée totale pendant laquelle la connaissance est présente au niveau de l'ontologie. Cette période peut être déterminée facilement grâce au temps global considéré dans le processus d'évolution et à l'émergence de la connaissance dans l'ontologie. La seconde caractéristique concerne les informations dont la validité est connue à l'avance. Dans ce cas précis, une période de validité est attribuée à la connaissance au bout de laquelle celle-ci sera supprimée de l'ontologie.

2.3.6 Synthèse

Notre étude de l'évolution de la connaissance et son adaptation au problème d'évolution des ontologies a mis en avant huit caractéristiques différentes représentées sur la figure 2.1. **L'émergence**, la **suppression**, **l'abstraction**, la **spécification**, la **persistance**, la **résistance**, la **distance sémantique** et le **poids sémantique**. Ces nouvelles notions font références aux principaux éléments moteurs décrits par Piaget pour caractériser l'évolution des connaissances. En conséquence, ces éléments, s'ils sont pris en compte, peuvent contribuer au problème d'évolution d'ontologies. Nous allons maintenant proposer une modélisation de ces huit caractéristiques pour l'évolution d'ontologies.

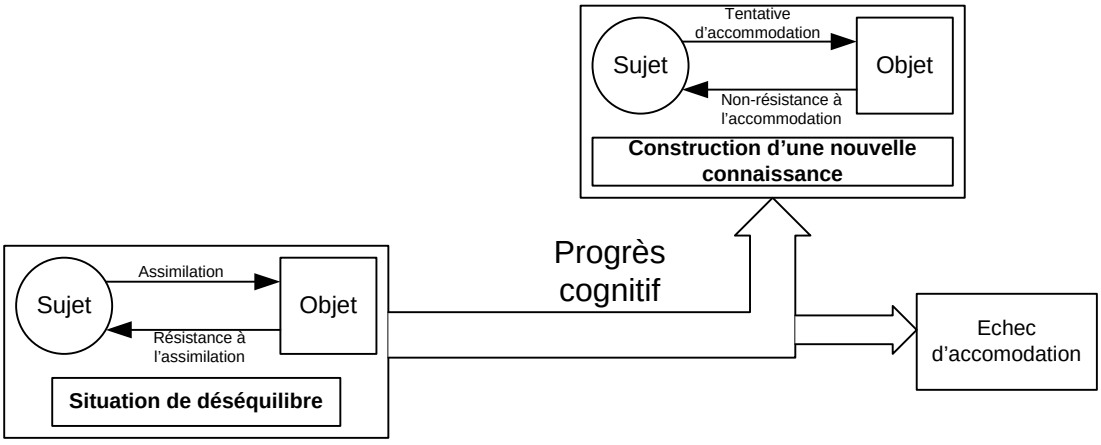


FIGURE 2.1 – Processus global d'évolution des connaissances

2.4 Modélisation de l'évolution

Dans cette section nous allons aborder le problème de la modélisation des différentes caractéristiques d'évolution des connaissances présentées à la section 2.3.

Emergence

L'émergence des connaissances va tout d'abord se traduire au niveau de l'ontologie par la définition de nouveaux éléments ontologiques (concepts, relations ou instances). De plus, afin de matérialiser davantage cette émergence, nous avons fait le choix de la caractériser par une date appelée **date d'émergence** qui correspond au moment où chaque concept et instance est ajouté à l'ontologie. Nous avons décidé d'appliquer la date d'émergence aux seuls concepts et instances car les autres éléments ontologiques et notamment les relations ne sont définies que par les concepts qu'elles relient. Le date d'émergence d'une relation est celle attribuée au concept le plus récent impliqué dans cette relation.

Persistance

La persistance que nous proposons d'intégrer à l'ontologie définit la période de validité des éléments ontologiques. Dans cet objectif, la période de validité P d'une connaissance peut être modélisée par une propriété des concepts et instances de l'ontologie dont le domaine de valeurs est l'ensemble des entiers naturels :

$$\boxed{P \in \mathbb{N}^*}$$

La valeur de P correspond aux nombres de pas d'évolution durant lesquels les éléments ontologiques doivent persister dans l'ontologie. Le choix d'un entier naturel est influencé par la nature discrète du temps dans notre vision de l'évolution d'ontologies. Dans notre approche, la durée de persistance ne s'applique qu'aux concepts et instances de concepts pour les mêmes raisons que pour le phénomène d'émergence.

Poids sémantique

Le poids sémantique SW (ou l'importance) est également modélisé par une propriété des concepts et instances de l'ontologie dont les valeurs sont prises dans l'ensemble des **entiers naturels compris entre 1 et 3**. Le choix d'une valeur entière permet avant tout de simplifier la compréhension de la notion du poids sémantique pour la personne dont la tâche est de fixer les valeurs du poids sémantique à chaque concept de l'ontologie.

$$\boxed{SW \in [\![1, 3]\!]}$$

Les valeurs prises par le coefficient représentant le poids sémantique sont représentatives de l'importance de l'élément dans l'ontologie. Ainsi, un coefficient de 1 représente un élément peu important, 2 un élément moyennement important et 3 un élément très important. Les valeurs ainsi définies sont plus faciles à comprendre (vu qu'il n'y a que trois valeurs possibles) pour l'expert du domaine chargé de construire l'ontologie. Par contre, cette plage de valeurs ne permet pas une mesure très fine de l'importance d'un concept dans l'ontologie.

Distance sémantique

Nous avons décidé de modéliser la distance sémantique SD par une propriété des relations de l'ontologie dont le domaine de valeurs est l'ensemble des **entiers naturels compris entre 1 et 10**. Tout comme pour le poids sémantique, l'utilisation de valeurs entières pour mesurer la distance sémantique permet une simplification de la compréhension de la notion de distance entre les concepts de l'ontologie. Mais contrairement au poids sémantique, nous considérons que la distance sémantique doit prendre ses valeurs dans un intervalle plus large car nous voulons avoir une mesure plus fine de la distance séparant les concepts de l'ontologie. Cet argument sera rediscuté au chapitre 5.

$$\boxed{SD \in [\![1, 10]\!]}$$

De la même façon que pour le poids sémantique, les valeurs de la distance sont représentatives de l'éloignement séparant les concepts de l'ontologie. Ainsi, 1 représente l'éloignement minimal et 10 la distance maximale. Par contre, cette valeur mesurant une distance entre concepts ou instances de l'ontologie, s'applique sur l'élément ontologique permettant de les relier entre eux, c'est-à-dire les relations.

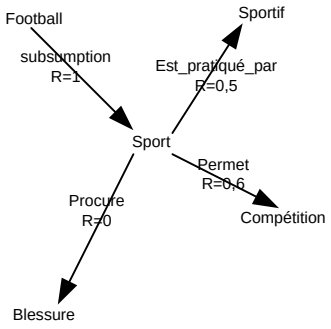
Résistance

La résistance R est modélisée dans notre approche par une propriété des relations ontologiques dont le domaine de valeur est l'ensemble des réels :

$$\boxed{R \in \mathbb{R}, 0 < R \leq R_{max}}$$

La valeur finie R_{max} représente la valeur de la résistance maximale définie par l'expert du domaine. Cette valeur s'applique aux relations ontologiques. Un peu à la manière d'un maillage, plus les cordes reliant les nœuds sont résistantes, moins l'ensemble évoluera. La résistance R_C, associée à un concept C, correspond à la moyenne des résistances appliquées aux relations ontologiques se rapportant à C.

Exemple: Considérons l'exemple suivant pour illustrer la résistance d'un concept :



Dans cet exemple, $R_A = \frac{1+0.5+0.6+0}{4} = 0,525$

Concernant les autres caractéristiques, à savoir la suppression, l'abstraction et la spécialisation des connaissances, elles ne sont que le résultat de l'évolution des valeurs des autres caractéristiques (persistance, résistance, poids et distance sémantique). Par conséquent, aucune modélisation particulière n'est nécessaire. Nous discuterons plus en détail de ces points au chapitre suivant décrivant le processus d'adaptation que nous proposons ainsi que les conséquences de l'évolution sur l'ontologie.

2.5 Conclusion

Dans ce chapitre, nous avons introduit un nouveau modèle d'ontologie, les ontologies adaptatives. Les composantes de ce modèle sont directement inspirées des idées développées dans des domaines comme la psychologie et la biologie et sont au nombre de cinq. Trois d'entre elles

(la date d'émergence, la durée de validité et le poids sémantique) s'appliquent directement sur les concepts d'une ontologie alors que les deux autres (la distance sémantique et la résistance aux changements) s'appliquent sur les relations d'une ontologie. Nous avons également proposé une modélisation mathématique de ces différents éléments. Notre étude a aussi mis en avant plusieurs autres phénomènes résultant de l'évolution, à savoir la généralisation et la spécialisation des connaissances ainsi que la suppression des connaissances d'un domaine particulier. Pour résumé, le modèle que nous proposons consiste en l'augmentation d'une ontologie au sens de Kalfoglou (voir section 1.1.2) par les notions de date d'émergence, de durée de validité et de poids sémantique pour les concepts de ce modèle et par la distance sémantique et la résistance pour les relations du modèle.

Après avoir introduit le modèle des ontologies adaptatives, nous allons à présent discuter de sa mise en œuvre. Dans le chapitre suivant, nous allons introduire un processus s'appuyant sur les caractéristiques des ontologies adaptatives, pour adapter l'ontologie aux évolutions des connaissances du domaine.

Chapitre 3

Processus d'adaptation d'ontologies

Sommaire

L'évolution des connaissances ou leur adaptation sont décrites dans toutes les approches (en psychologie ou en biologie) comme des processus, c'est à dire comme une suite d'états ou d'étapes du déroulement d'une opération ou d'une transformation. Le bon fonctionnement de ce processus d'évolution est dicté par un ensemble de règles intransgressables agissant sur les éléments de base formant la connaissance d'un individu.

La transposition du processus d'adaptation des connaissances au problème de l'adaptation des ontologies nécessite l'adaptation des règles qui régissent ce processus de telle sorte que les ontologies s'adaptent fidèlement aux évolutions du domaine. Ainsi, à l'instar du modèle d'ontologies adaptatives, la définition de ces règles est fondée sur les travaux de Piaget sur l'évolution des connaissances et leur application s'appuie sur l'utilisation d'un ensemble d'adjuvants.

Un corpus de documents significatifs des connaissances du domaine est l'un d'entre eux. Ce corpus sera l'élément de base sur lequel repose les règles d'adaptation. En conséquence, il doit être construit de manière rigoureuse et présenter certaines propriétés comme une taille adéquate afin de couvrir l'ensemble du domaine et être dynamique pour refléter les évolutions du domaine à travers le temps.

Les règles mettent en œuvre un ensemble de métriques permettant de mesurer l'évolution des connaissances. La définition de ces métriques doit prendre en compte le contenu et la structure des documents du corpus souvent révélatrices des caractéristiques des informations qu'ils contiennent.

La qualité des métriques et leur application aux éléments des ontologies adaptatives va déterminer la qualité de l'évolution de l'ontologie.

L'application des règles d'adaptation modifie la structure des ontologies. Des éléments ontologiques peuvent être ajoutés, d'autres modifiés ou supprimés. Toutes ces modifications doivent se faire tout en n'entravant pas les possibilités de raisonnement sur l'ontologie qui a évoluée.

Dans ce chapitre, nous définissons le processus d'adaptation associé aux ontologies adaptatives introduites au chapitre précédent. Nous détaillons tout d'abord les mécanismes et les caractéristiques des éléments sur lesquels sont fondées les règles d'adaptation. Nous énonçons ensuite les règles puis présentons les différents métriques utilisées par ces règles. Enfin, nous traitons des différents problèmes que pose l'application des règles d'adaptation sur les ontologies.

3.1 Motivations

La proposition d'un processus pour l'adaptation des ontologies est motivée par plusieurs raisons. Les ontologies adaptatives, telles qu'elles sont décrites au chapitre précédent, contiennent un ensemble d'éléments modélisés par des coefficients qui doivent pouvoir être modifiés de manière souple sans avoir obligatoirement besoin de l'intervention d'un expert. La proposition de règles pour l'adaptation des ontologies doivent permettre une évolution fiable limitant les interventions d'un expert du domaine.

La définition d'un processus d'adaptation nécessite une réflexion profonde sur la notion d'adaptation. Les règles émanant de cette réflexion permettent la définition de stratégies pour l'évolution des ontologies. Une stratégie est vue comme une méthode générale déterminée par l'expert lors de la construction de l'ontologie lui permettant de laisser l'ontologie évoluer seule, suivant des règles, sans avoir incessamment à intervenir d'où la proposition d'un ensemble de règles pour l'adaptation des ontologies.

Une autre lacune mise en évidence par notre étude concerne le peu d'outils supportant l'évolution des ontologies. La définition d'un processus automatique (ou semi-automatique) pour l'adaptation des ontologies facilitera le développement d'un tel outil.

3.2 Les mécanismes de l'adaptation des connaissances

Les mécanismes de l'adaptation sont décrits dans la théorie piagetienne comme reposant sur un phénomène appelé *régulation* [Piaget, 1974]. Une régulation est un processus spécialisé grâce auquel un système (ou un sous-système) finalisé tend à atteindre l'un de ses états d'équilibre. L'équilibre visé peut être statique, comme la température d'un organisme vivant, ou dynamique, comme une succession régulière et canalisée d'états (en ce dernier cas, le réglage peut porter, par exemple, sur la vitesse de transformation d'un état à un autre). Dans ses travaux, Piaget appelle régulation :

> *«les compensations partielles dues aux décentrations qui tendent à modérer les déformations inhérentes à chaque centration. La régulation est donc engagée sur la voie de la réversibilité et constitue bien l'intermédiaire entre l'assimilation déformante (centration) et l'assimilation opératoire.»*

Dans sa définition de la régulation, Piaget en distingue plusieurs types :

- Les régulations *proactives* : Elles interviennent lorsque l'apprenant se reporte aux critères de réalisation avant d'accomplir la tâche. Il agit par anticipation et régule pro-activement son inscription dans la tâche. A la différence de la remédiation, qui consiste en un réajustement de la réponse une fois que l'action est finie, la régulation proactive consiste en une anticipation. L'apprenant devance des actions qui s'avéraient non pertinentes pour la suite de son activité.
- Les régulations *interactives* : Elles font suite à une réponse ou une évaluation. Suivant cette réponse, l'apprenant peut réguler son activité. Ce type de régulations suppose un processus continu de référenciation.
- Les régulations *rétroactives* : Elles sont sans doute les plus courantes. Ce sont celles qui demandent le moins d'effort à l'apprenant. Elles consistent en la correction du système cognitif de l'individu après que la bonne information lui soit parvenue. Aucun effort de raisonnement n'est nécessaire.

Il convient également de distinguer les régulations qui construisent de celles qui réajustent simplement un déséquilibre. Ces dernières sont alors soit de nature mécaniste, fondées sur des arrangements préétablis et à rétroactions linéaires, soit de nature dynamique, à caractère homéostatique, résultat de l'interaction mutuelle entre les composantes. Dans les deux cas, elles tendent vers un équilibre ou un état stable. Les régulations qui construisent permettent, quant à elles, d'ouvrir les structures, c'est-à-dire de construire de nouveaux observables sur l'objet à connaître et ainsi de pouvoir anticiper sur de nouveaux possibles.

Cependant, l'activité de régulation peut également avoir un objectif de contrôle. En effet, toute activité (par exemple la résolution d'un problème) nécessite, pour la personne qui l'entreprend, des processus de régulation dans le sens de contrôle et de correction des productions du sujet. Selon les cognitivistes, le contrôle est donc inhérent à toute activité. Même si les observables de cette activité peuvent paraître incohérentes, toute démarche finalisée a une cohérence interne qui peut échapper à l'observateur. La régulation ou l'ajustement est par contre visible et est la conséquence immédiate de la prise en compte des éléments du contrôle. Dans ce sens, trois opérations s'inscrivent dans l'élaboration de toute activité :

- L'anticipation, c'est-à-dire l'appropriation par le sujet des critères de la tâche, l'orientation de son action future et la mobilisation des connaissances nécessaires.
- Le contrôle (ou monitorage) de l'action, c'est-à-dire un mécanisme de comparaison entre le produit attendu (représenté) et le produit réel (production du sujet). Les activités de contrôle sont reliées à la surveillance de ce que fait l'apprenant, à la vérification de ses progrès et à l'évaluation de la conformité et de la pertinence des étapes suivies, des résultats obtenus ou des stratégies utilisées.
- L'ajustement est la correction progressive et constante des écarts entre le produit attendu et le produit réel et l'éventuelle réorientation de l'action en fonction des observations faites. Les activités de régulation sont reliées aux interventions qu'on décide de faire d'après ce qui a été détecté par les activités de contrôle : apporter un correctif, changer de stratégie, arrêter une procédure ou, au contraire, continuer la démarche en cours.

Certaines activités, comme l'écriture d'un texte par exemple, mettent en œuvre d'autres types de régulations. En fait, au cours de cette tâche particulière, le scripteur effectue constamment un va-et-vient entre ce qu'il veut écrire (représentation de la tâche) et le comment il va s'y prendre pour le faire ou pour modifier son action (les processus de production). Ces allers et retours lui permettent de réguler son action (et sa production) tout au long du travail, c'est-à-dire d'adapter au fur et à mesure de son élaboration, le texte qu'il produit avec le texte qu'il désire produire. Les régulations permettent au sujet d'orienter son processus de textualisation de façon à ce qu'il corresponde aux exigences de la tâche. Ces nouvelles régulations se distinguent suivant différentes

caractéristiques :

- Les régulations implicites échappent évidemment non seulement à l'observateur puisque les régulations sont complètement intégrées aux processus cognitifs eux-mêmes, mais également au sujet qui n'a bien sûr pas conscience de ce qui se passe. Ces régulations implicites peuvent se situer à des niveaux différents : le niveau de la tâche elle-même dans le processus de textualisation proprement dit, mais également à des niveaux plus éloignés de la tâche comme l'attention, la fatigue, le temps à disposition, les besoins vitaux etc.
- Les régulations explicitables échappent au contrôle du sujet ou de l'observateur tant qu'une contrainte sur ce plan n'est pas installée.
- Les régulations explicitées et instrumentées sont celles dont le sujet peut parler avec aisance. Par le biais d'entretiens et de supports divers (graphique de production du texte, enregistrements de l'activité, traces, ...), il est possible de faciliter l'explicitation de certaines régulations.

L'idée principale apportée par la régulation vise à amener un équilibre dans le système cognitif de l'individu en **anticipant** certaines activités et en tenant compte de **l'état dans lequel se trouve la structure cognitive** et aussi les nouvelles informations qu'il perçoit tout en **contrôlant** son évolution. C'est sur ces idées que nous proposons d'établir le processus d'adaptation des ontologies.

3.3 Règles d'adaptation d'ontologies

Comme évoqué à la section précédente, le phénomène de régulation explique la façon dont les structures cognitives s'adaptent aux nouvelles connaissances parvenues à l'apprenant. Or, la régulation est régie suivant un ensemble de règles garantissant le bon fonctionnement du processus. Le processus d'adaptation des ontologies que nous proposons doit donc également se faire suivant des règles inspirées du processus de régulation de Piaget. Ces règles nécessitent la mise en œuvre de plusieurs composants, un corpus de documents et des métriques.

Dans les sous-sections qui suivent, nous allons discuter de ces différents composants ainsi que des règles que nous proposons pour l'adaptation des ontologies.

3.3.1 Corpus de documents

Le corpus de documents représentatifs des connaissances d'un domaine et son analyse tiennent une place prépondérante dans la définition des règles d'adaptation et des métriques qui leur sont associées. Dans notre approche, un corpus est constitué de documents eux mêmes constitués de mots. Ces notions sont définies de la sorte :

Définition 3.1 *Soit Σ un alphabet, un mot $\omega = a_1 a_2 \dots a_n$ avec $\forall i \in [\![1, n]\!], a_i \in \Sigma$*

Remarque: Dans notre approche, Σ représente l'alphabet latin.

Définition 3.2 *Un document d est un multi-ensemble de mots. $d = (M, m)$ où M représente l'ensemble des mots du document et $m : M \to \mathbb{N}$ représente la multiplicité (i.e. le nombre de fois que chaque mot apparaît dans le document). De plus, la cardinalité d'un document (card(d)) représente la somme des multiplicités.*

Remarque: Un multi-ensemble fini se note en utilisant des doubles accolades $\{\{\ldots\}\}$ qui encadrent les éléments ayant une multiplicité strictement positive. Par exemple $\{\{a, b, a, b, b, d\}\}$ représente le multi-ensemble $(\{a, b, c, d\}, m)$ où m est la fonction telle que m(a)=2, m(b)=3, m(c)=0 et m(d)=1. Nous serons amenés à utiliser les deux notations d'un multi-ensemble dans la suite de ce mémoire. L'ordre dans lequel apparaissent les mots dans le document est important. Ainsi dans notre approche, les ensembles $\{\{a, b, a, b, b, d\}\}$ et $\{\{a, a, b, b, b, d\}\}$ ne sont pas équivalents.

Remarque: $M \subseteq L$ où L représente l'ensemble des mots du langage naturel construit sur l'alphabet latin.

Définition 3.3 *Soit* $\omega_1 = a_1 a_2 \ldots a_n$ *et* $\omega_2 = b_1 b_2 \ldots b_m$ *deux mots,*

$$\omega_1 = \omega_2 \Leftrightarrow (n = m \wedge \forall i \in [\![1, n]\!], a_i = b_i)$$

Définition 3.4 *Un corpus de documents K est un ensemble fini de documents.* $K = \{d_1, d_2, \ldots, d_n\}$

Définition 3.5 *Soit* $K = \{d_1, d_2, \ldots, d_n\}$ *un corpus de n documents, la cardinalité de K notée card(K) est le nombre de mots présents dans le corpus.*

$$card(K) = \sum_{i=1}^{n} card(d_i)$$

Les caractéristiques de ce corpus et des documents le composant sont essentielles. La qualité des documents et de leur contenu est une de ces caractéristiques. Un corpus bien formé doit nécessairement couvrir un seul langage, et une seule déclinaison de ce langage. Il existe par exemple de subtiles différences entre l'anglais du Royaume-Uni et celui parlé aux Etats-Unis. Le temps joue un rôle important dans l'évolution du langage : le français parlé aujourd'hui ne ressemble pas au français parlé il y a 200 ans ni, de façon plus subtile, au français parlé il y a 10 ans, à cause notamment des néologismes. C'est un phénomène à prendre en compte pour toutes les langues vivantes. Un corpus ne doit donc pas contenir de textes rédigés à des intervalles de temps trop larges. Il ne faut pas non plus mélanger des registres différents et le scientifique ne peut s'autoriser à extraire des informations d'un corpus destiné à un certain registre en les appliquant à un autre. Un corpus construit à partir de textes scientifiques ne peut être utilisé pour extraire des informations sur les textes vulgarisés, et un corpus mélangeant des textes scientifiques et vulgarisés ne permettra de tirer aucune conclusion sur ces deux registres.

Le corpus doit évidemment atteindre une taille critique pour permettre des traitements statistiques fiables. Il est impossible d'extraire des informations fiables à partir d'un corpus trop petit. La taille du corpus est aussi fonction du but dans lequel il a été établi. Selon la lexicographe Renouf :

> *«the larger the amount of data available, the more reliable would be the statements which could be made about language»*

Ainsi, un corpus utilisé à des fins lexicographiques doit contenir des dizaines, voire des centaines, de millions de mots si l'on veut établir des statistiques valables sur l'utilisation de ces mots.

Ensuite, comme nous nous proposons de mesurer l'évolution des connaissances d'un domaine, le corpus doit être dynamique, c'est-à-dire être significatif de l'état des connaissances du domaine à un moment précis du temps. De ce fait, les documents constituant le corpus ne doivent pas être trop ancien, auquel cas la connaissance contenue dans ces documents risquerait d'être obsolète. Les documents ne reflétant pas suffisamment la connaissance du domaine doivent être retirés du

corpus. En contrepartie, de nouveaux documents doivent y être ajoutés afin que le corpus soit représentatif des connaissances ayant émergé dans le domaine. Cette dynamique représente en quelque sorte l'évolution des connaissances du domaine auquel se rapportent les documents du corpus.

Dans notre approche, l'analyse de ce corpus va fournir les éléments de base à l'adaptation des ontologies. L'analyse des documents du corpus a, en effet, pour but de mesurer l'évolution des connaissances du domaine.

L'analyse du corpus peut être soit statistique, soit sémantique. D'un point de vue statistique, on peut considérer un corpus comme un échantillon d'une population (d'événements langagiers). Comme tout échantillon, un corpus est passible de deux types d'erreurs statistiques qui menacent les généralisations à partir de son utilisation : *l'incertitude* (random error) et la *déformation* (bias error). L'incertitude survient quand un échantillon est trop petit pour représenter avec précision la population réelle. Une déformation se produit quand les caractéristiques d'un échantillon sont systématiquement différentes de celles de la population que cet échantillon a pour objectif de refléter. Pour ces raisons, les métriques dédiées à l'analyse statistique du corpus doivent être rigoureusement définies (voir section 3.3.3).

L'analyse sémantique (i.e. du sens du contenu) doit faire intervenir d'autres outils, comme ceux utilisés dans le domaine du traitement automatique des langues (voir section 1.1.2). Si le corpus est constitué de pages Web, l'utilisation des technologies du Web Sémantique peuvent être mises en œuvre. Ceci nécessite leur enrichissement des pages Web en métadonnées extraites d'ontologies pour faciliter la compréhension du contenu et permettre son exploitation.

Remarque : Un exemple de construction et d'analyse de corpus est donné au chapitre 7 consacré à la validation expérimentale de notre approche.

Dans la section suivante, nous allons présenter l'adaptation du principe de régulation à travers la définition des règles régissant l'adaptation des ontologies.

3.3.2 Définition des règles

Comme évoqué précédemment, le processus d'adaptation des ontologies que nous proposons est dirigé par un ensemble de règles. Les règles, qui sont au nombre de six, s'appuient sur les documents du corpus. Elles agissent sur les éléments de base des ontologies adaptatives (concepts, relations, instances de concept ...) et sur les éléments spécifiques aux ontologies adaptatives (l'émergence, la persistance, le poids sémantique, la distance sémantique et la résistance). Les règles que nous proposons sont les suivantes :

Règle 1 : Détecter puis intégrer tous les nouveaux concepts du domaine dans l'ontologie.

Cette règle fait référence au principe d'émergence des connaissances. La détection des nouveaux concepts dans notre approche s'appuie sur l'utilisation du corpus et de son analyse statistique (voir section 3.3.3). Les concepts détectés et acceptés comme pertinents par l'expert du domaine donnent lieu à la définition de nouvelles classes dans l'ontologie. L'expert du domaine a aussi à charge la définition des relations ontologiques reliant les nouveaux concepts à ceux déjà présents dans l'ontologie. La définition des valeurs des éléments spécifiques aux ontologies adaptatives associés à nouvel élément ontologique se fait soit automatiquement pour la distance sémantique, le poids sémantique et la date d'émergence, soit manuellement pour la résistance aux changements et la durée de validité.

Règle 2 : Décrémenter, à chaque pas d'évolution, la période de validité de tous les concepts de l'ontologie en fonction de leur résistance aux changements.

La règle 2 met en évidence le principe de résistance aux changements, fondement du principe d'évolution de Darwin, suivant lequel, les plus résistants persistent. La métrique utilisée pour calculer la nouvelle période de validité tient compte de cette remarque. Celle-ci est détaillée à la section 3.3.3. Elle est indépendante du corpus de documents.

Règle 3 : Adapter ; à chaque pas d'évolution, la distance sémantique entre les concepts de l'ontologie en fonction de la résistance aux changements qui leur est associée.

Le principe de régulation de Piaget est à l'origine de cette règle. L'idée est de tenir compte de l'évolution des connaissances, matérialisée par l'aspect dynamique du corpus, afin de rééquilibrer la distance sémantique à chaque pas d'évolution. La résistance aux changements interviennent pour contrebalancer l'effet de l'évolution en s'opposant à l'éloignement ou au rapprochement de concepts.

Règle 4 : Adapter, à chaque pas d'évolution, le poids sémantique associé à chaque concepts de l'ontologie en fonction de leur résistance aux changements.

De la même façon, le poids sémantique (ou l'importance) associé à chaque concept doit être réévalué en fonction du contenu des nouveaux documents ajoutés au corpus et ceux qui y ont été retirés. Le principe de régulation est également à l'origine de cette règle et notamment le phénomène de renforcement des connaissances. L'idée est d'utiliser l'analyse statistique du corpus afin d'adapter le poids sémantique de chaque concept en tenant compte de l'action de la résistance aux changements. Intuitivement, nous considérons que plus un terme est mentionné dans le corpus, plus le concept qui lui est associé est important pour le domaine et inversement.

Règle 5 : Redéfinir, à chaque pas d'évolution, la résistance aux changements de toutes les relations ontologiques.

L'expert du domaine peut également réévaluer la résistance aux changements associée à chaque relation ontologique. En procédant de la sorte, l'expert exerce un contrôle sur l'évolution de l'ontologie. La modification de la résistance aux changements permet de corriger les effets de l'évolution ou une mauvaise attribution des valeurs par défaut sur lesquelles agit la résistance aux changements.

Règle 6 : Retirer de l'ontologie tous les concepts dont la durée de validité est nulle.

La dernière des six règles que nous proposons fait intervenir le principe d'obsolescence des connaissances. Cette règle consiste en l'élimination des concepts dont la durée de validité est nulle, c'est-à-dire les concepts qui n'ont plus lieu d'exister dans l'ontologie. Si toutefois les termes associés aux concepts supprimés apparaissent fréquemment dans le corpus, le terme sera à nouveau détecté et l'expert en charge de l'ontologie pourra, s'il le juge nécessaire, redéfinir le concept dans l'ontologie par application de la règle 1.

Les règles ainsi définies suivent les idées du principe de régulation de Piaget. La règle 1 consiste en l'assimilation piagetienne. La règle 2 est inspirée à la fois du phénomène d'évolution de Darwin et du principe d'anticipation de Piaget. Les règles 3 et 4 font directement référence au principe de rééquilibration des connaissances suivant les nouvelles informations du domaine

contenues dans le corpus. Le principe de contrôle prépondérant dans le principe de régulation est mis en œuvre dans la règle 5 et enfin celui d'obsolescence régit la règle 6. La définition de ces règles fait explicitement référence à l'utilisation de métriques. Nous allons à présent les définir mathématiquement.

3.3.3 Les métriques pour l'adaptation

Les métriques que nous proposons d'utiliser dans notre approche sont de deux types différents (voir figure 3.1). Le premier type concerne l'analyse du corpus alors que le second est dédié à l'adaptation des valeurs associées aux éléments spécifiques aux ontologies adaptatives.

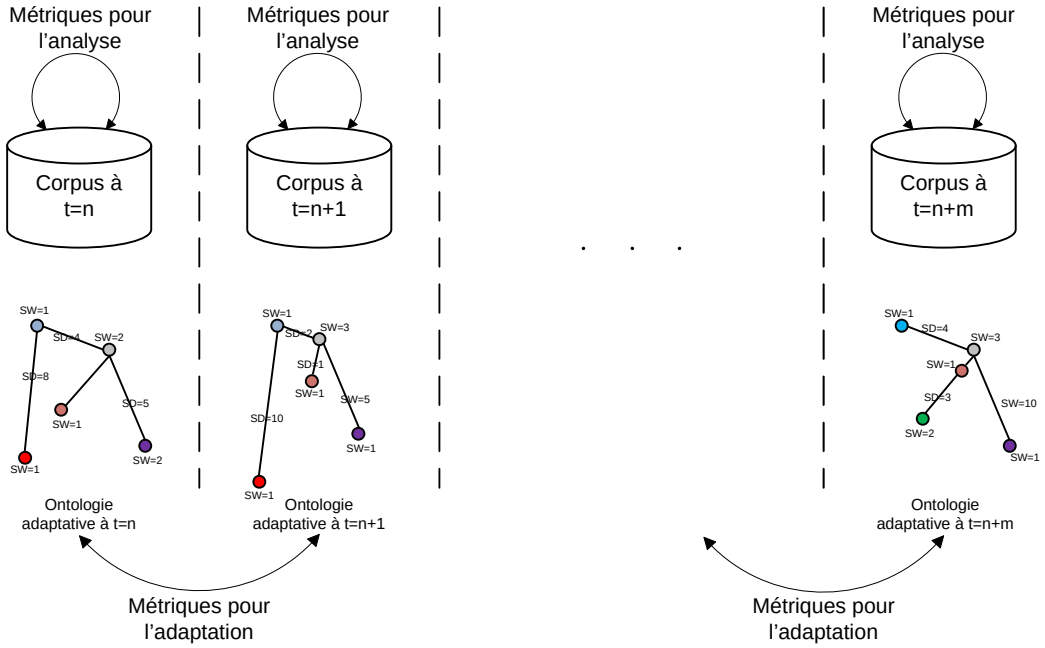


FIGURE 3.1 – Principe d'application des métriques

Métriques pour l'analyse du corpus

La première famille de métriques utilisée dans notre approche est celle dédiée à l'analyse du corpus de documents. Ces métriques sont impliquées dans la détection des nouveaux concepts, dans le calcul des poids sémantiques associés aux concepts et de la distance sémantique qui les sépare. Les métriques que nous allons présenter sont basées sur l'utilisation d'outils statistiques pour l'analyse de corpus.

Détection des nouveaux concepts potentiels

La détection de nouveaux concepts à intégrer à l'ontologie du domaine repose sur l'idée suivante : *plus un terme est cité dans le corpus de documents, plus le concept associé à ce terme à de chance d'être important pour le domaine et par conséquent doit être intégré à l'ontologie du domaine.*

Remarque : Le potentiel d'un mot peut également se refléter à travers les parties des documents

dans lesquelles apparaissent les mots (comme un titre ou un simple paragraphe). Cependant, comme les documents du corpus ne sont pas forcément homogènes du point de vue de leur structure l'utilisation de celle-ci fausserait le calcul du potentiel.

Suivant l'idée énoncée précédemment, nous définissons le potentiel d'un terme de la façon suivante :

Définition 3.6 *Soit $K = \{d_1, d_2, \ldots, d_n\}$ un corpus de n documents, soit $\omega \in d_i, i \in [\![1, n]\!]$, un mot d'un document de K. La quantité $\sum_{d \in K} m(\omega)$ est le potentiel de ω noté $\pi(\omega)$.*

Le potentiel représente le nombre d'occurrences de ω dans K. Ainsi, les termes ayant le plus fort potentiel présentent la probabilité la plus élevée d'être des termes clés du domaine et par conséquent les concepts auxquels ils sont associés doivent logiquement faire partie de l'ontologie du domaine.

Distance sémantique

La distance sémantique que nous proposons de calculer, fonctionne sur l'idée que *plus deux mots sont proches dans le document où ils apparaissent conjointement, plus le lien sémantique qui les relie est fort.* Suivant cette observation, nous pouvons définir la distance sémantique.

Définition 3.7 *Soit $d = \{\{\omega_1, \omega_2, \ldots, \omega_n\}\}$ un document de n mots, la distance sémantique SD_d entre deux termes α_1 et α_2 de d est l'écart moyen (en nombre de termes) entre α_1 et α_2*

$$SD_d(\alpha_1, \alpha_2) = \frac{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} j-i, (i,j) \in A}{\lambda(\alpha_1, \alpha_2)}$$

$$Avec\ A = \{(i,j) \in [\![1,n]\!]^2, (i<j) \wedge ((\omega_i = \alpha_1 \wedge \omega_j = \alpha_2) \wedge (\nexists k, i < k < j, \omega_k = \alpha_1)) \vee ((\omega_i = \alpha_2 \wedge \omega_j = \alpha_1) \wedge (\nexists k, i < k < j, \omega_k = \alpha_2))\}$$

$$et\ \lambda(\alpha_1, \alpha_2) = |\{(\omega_i, \omega_j) : ((\omega_i = \alpha_1 \wedge \omega_j = \alpha_2) \wedge (\nexists k, i < k < j, \omega_k = \alpha_1)) \vee ((\omega_i = \alpha_2 \wedge \omega_j = \alpha_1) \wedge (\nexists k, i < k < j, \omega_k = \alpha_2))\}|$$

Exemple: Par exemple considérons le document $d = \{\{a, b, a, b, b, d\}\}$, $SD_d(a, a) = \frac{4}{2}$.

Définition 3.8 *Soit $K = \{d_1, d_2, \ldots, d_n\}$ un corpus de n documents, soit deux mots α_1 et α_2 deux mots de K, la distance sémantique*

$$SD_K(\alpha_1, \alpha_2) = \lceil 10. \frac{\sum_{i=1}^{n} SD_{d_i}(\alpha_1, \alpha_2)}{card(K)} + 1 \rceil$$

est la distance sémantique entre α_1 et α_2 sur K.

Remarque: La distance sémantique sur un corpus n'est que la généralisation de la distance sémantique à tous les documents du corpus. Cette quantité est normalisée afin de mieux correspondre à la distance sémantique entre les concepts d'une ontologie adaptative.

Importance d'un terme

Le TF-IDF (de l'anglais term frequency-inverse document frequency) est une méthode de pondération souvent utilisée dans la fouille de textes. Cette mesure statistique permet d'évaluer l'importance d'un mot par rapport à un document extrait d'une collection ou d'un corpus. Le poids augmente proportionnellement en fonction du nombre d'occurrences du mot dans le

document. Il varie également en fonction de la fréquence du mot dans le corpus. Cette métrique est utilisée dans certains moteurs de recherche car il permet de trouver le document le plus pertinent en fonction des mots de la requête.

Définition 3.9 *Soit $K = \{d_1, d_2, \ldots, d_n\}$ un corpus de documents, soit ω un mot du corpus, la fréquence de ω notée $tf(\omega)$ est définie comme :*

$$tf_{d_i}(\omega) = \frac{\sum_{d \in K} m(\omega)}{card(K)}$$

Définition 3.10 *Soit $K = \{d_1, d_2, \ldots, d_n\}$ un corpus de documents, soit ω un mot du corpus, l'IDF de ω notée $IDF(\omega)$ est définie comme :*

$$IDF_{d_i}(\omega) = \log \frac{n}{|\{d_i : \omega \in d_i\}|}$$

Cependant la fréquence des termes est importante mais pas suffisante pour saisir l'importance d'un mot dans le corpus. En fait, si un seul document du corpus présente un grand nombre d'occurrences d'un mot ω et que les neufs autres documents du corpus ne contiennent pas une seule occurrence de ω, la fréquence sera tout de même élevée. Pour contrebalancer cet effet, il est nécessaire de considérer la répartition des mots dans le corpus. Ainsi, plus la répartition du mot dans le corpus sera homogène, plus le mot sera considéré comme important.

Définition 3.11 *Soit $K = \{d_1, d_2, \ldots, d_n\}$ un corpus de documents, un document d est de la forme $\{\{\omega_1, \omega_2, \ldots, \omega_m\}\}$. La répartition d'un mot ω dans un document d, notée $Rep_d(\omega)$ est la distance moyenne entre deux occurrences successives du même mot dans ce document :*

$$Rep_d(\omega) = \frac{SD_d(\omega, \omega)}{m(\omega)}$$

Définition 3.12 *La répartition d'un mot ω dans un corpus K, notée $Rep_K(\omega)$ est la généralisation de la définition 3.11 à l'ensemble des documents du corpus.*

La prise en compte de la répartition du mot dans le corpus va plus loin que le fait de vérifier si un document contient le mot dont on veut mesurer l'importance, comme c'est le cas dans la mesure TF-IDF. La mesure de la répartition est plus fine car elle s'appuie sur l'ensemble des mots des documents, pas seulement sur l'un d'entre eux sans considérer le reste du document. En conséquence, l'importance d'un terme dans un corpus est défini comme suit :

Définition 3.13 *Soit $K = \{d_1, d_2, \ldots, d_n\}$ un corpus et un mot ω. La quantité*

$$Imp_K(\omega) = \lceil 3.tf(\omega).\frac{Rep_K(\omega)}{card(K)} + 1 \rceil$$

représente l'importance de ω dans K.

Remarque: La valeur de l'importance est normalisée en vertu de la définition du poids sémantique d'un concept dans une ontologie adaptative.

Les métriques présentées jusqu'à présent vont nous permettre de définir la deuxième famille de métriques nécessaires dans notre approche à savoir les métriques pour l'adaptation des ontologies (voir figure 3.1).

Métriques pour l'adaptation des ontologies

Ce type de métriques permet la modification des coefficients du modèle des ontologies adaptatives en fonction de l'évolution des connaissances du corpus de documents. Comme le montre la figure 3.1, l'idée est d'adapter l'état de l'ontologie adaptative au temps t-1 aux nouvelles connaissances du domaine au temps t, l'ontologie adaptative résultante représentera l'état des connaissances du domaine au temps t. Le modèle des ontologie adaptatives contient trois éléments spécifiques dont l'évolution des valeurs peut être déterminée par l'analyse du corpus de documents. Ces éléments sont : la persistance (ou la durée de validité), le poids sémantique et la distance sémantique.

Métrique pour l'adaptation de la persistance

Comme illustré au chapitre précédent, la notion de persistance est dépendante de la résistance aux changements. L'évolution de la valeur attribuée à la durée de validité d'un concept dans l'ontologie est fonction de la résistance aux changements.

Définition 3.14 *Soit* $O_{t-1} =< C_{t-1}, R_{t-1}, A_{t-1}, I_{t-1} >$ *une ontologie adaptative au temps t, l'adaptation de la durée de validité associée à un concept* $c \in C_{t-1}$ *est définie comme :*

$$P_{c_t} = \begin{cases} 0 \quad si \quad R_{c_t} = 0 \\ P_{c_{t-1}} - \frac{1}{R_{c_t}} \quad si \quad 0 < R_{c_t} < 1 \\ P_{c_{t-1}} - 1 \quad si \quad R_{c_t} = 1 \\ P_{c_{t-1}} - 1 \quad si \quad (ED_c - t) \equiv 0 \ mod \ R_{c_t} \quad et \quad R_{c_t} > 1 \end{cases}$$

$ED_c - t$ *représente le temps écoulé entre la date d'émergence de c et t.*

La métrique illustre bien le principe de l'évolution des espèces de Darwin sur lequel nous nous appuyons, car plus la résistance est élevée, plus le concept va persister dans l'ontologie. La valeur de la résistance désigne le nombre de pas d'évolution nécessaires pour décrémenter la période de validité de 1. Une résistance de 1 désigne une résistance neutre alors qu'une résistance inférieure à 1 va avoir un effet d'accélérateur sur la diminution de la valeur de la durée de validité associée aux concepts.

Métrique pour l'adaptation du poids sémantique

L'évolution du poids sémantique suit l'idée de régulation des connaissances émise par Piaget et est fonction de la résistance aux changements. Plus la résistance du concept est forte, moins le poids sémantique associé au concept va évoluer (diminuer ou augmenter).

Définition 3.15 *Soit* $O_{t-1} =< C_{t-1}, R_{t-1}, A_{t-1}, I_{t-1} >$ *une ontologie adaptative au temps t-1 et* $K_t = \{d_1, d_2, \ldots, d_n\}$ *le corpus de document au temps t associé à l'ontologie, l'adaptation du poids sémantique associé à un concept c est définie comme :*

$$SW(c_t) = SW(c_{t-1}) + \frac{Imp_{K_t}(c_t) - SW(c_{t-1})}{R_{c_t}}$$

Métrique pour l'adaptation de la distance sémantique

De la même façon, l'adaptation de la distance sémantique associée aux relations entre les concepts d'une ontologie adaptative est fonction de la résistance aux changements associées à ces relations. De manière intuitive, plus la résistance est forte, moins la distance va évoluer au cours du temps et inversement.

Définition 3.16 *Soit* $O_{t-1} =< C_{t-1}, R_{t-1}, A_{t-1}, I_{t-1} >$ *une ontologie adaptative au temps t-1 et* $K_t = \{d_1, d_2, \ldots, d_n\}$ *le corpus de document au temps t associé à l'ontologie, l'adaptation de la distance sémantique associée à une relation ontologique* $r = (dom, ran) \in R_{t-1}$ *est définie comme :*

$$SD(r_t) = SD(r_{t-1}) + \frac{SD_{K_t}(dom,ran) - SD(r_{t-1})}{R_r}$$

Remarque: Les règles d'adaptation et les métriques mises en œuvre feront l'objet d'une validation expérimentale au chapitre 7.

3.3.4 Algorithme d'adaptation des ontologies

La définition des règles d'adaptation à la section 3.3.2 combinée aux métriques présentées ci-dessus nous ont permis de définir un algorithme pour l'adaptation des ontologies à l'évolution des connaissances du domaine au cours du temps. L'algorithme s'exécute à chaque pas d'évolution.

L'algorithme

L'algorithme que nous proposons se présente sous la forme suivante :

Algorithme 1 Algorithme d'adaptation d'une ontologie adaptative

Entrées: O une ontologie adaptative,
$t \in \mathbb{N}$ (le temps)
K un corpus de documents
Sorties: O l'ontologie évoluée
pour tous les concepts $c \in O$ **faire**
$AdapterPersistance(c, K, t)$
$AdapterPoidsSemantique(c, K, t)$
fin pour
pour toutes les relations $r \in O$ **faire**
$AdapterDistanceSemantique(r, K, t)$
$ModifierResistance(r)$
fin pour
$EliminerConcepts(O)$
$IntegrerNouveauxConcepts(O, K, t)$
Retourner O.

Les procédures AdapterPersistance, AdapterPoidsSemantique, AdapterDistanceSemantique modifient respectivement la persistance des concepts, leur poids sémantique et la distance sémantique qui les sépare en fonction des métriques présentées à la section précédente. La procédure ModifierResistance modifie la valeur de la résistance associée à la relation r en paramètre avec une valeur définie par l'expert en charge de l'ontologie. La procédure EliminerConcepts supprime les concepts obsolètes de l'ontologie (i.e. les concepts dont la durée de validité est nulle). Enfin,

IntegrerNouveauxConcepts consiste en l'ajout des nouveaux concepts du domaine à l'ontologie. Le label de ces concepts est déterminé à partir du corpus de document et du calcul du potentiel de chaque terme. Il convient ensuite de leur associer les valeurs initiales des coefficients de résistance des relations les reliant aux concepts existants, de durée de validité ainsi que la date d'émergence.

La complexité en temps de notre algorithme est $O(|C|.|R|)$ où $|C|$ représente le nombre de concepts de l'ontologie et $|R|$ le nombre de relations ontologiques. A ce temps d'exécution, il faut ajouter le temps nécessaire à la modification structurelle de l'ontologie suivant les concepts qui ont été supprimés. Ces modifications sont plus largement discutées à la section 3.4.

Exemple

Afin d'illustrer le fonctionnement de l'algorithme, nous considérons l'exemple suivant illustrant l'adaptation d'une ontologie construite sur une page du site Wikipédia [1] aux modifications des données de cette page Web. Dans cet exemple, les figures 3.2 et 3.3 représentent les premières versions de la page Web et de l'ontologie qui lui est associée. Les figures 3.4 et 3.5 représentent les deuxièmes versions de la page Web et de l'ontologie.

Remarque: Cet exemple ne fait qu'illustrer le fonctionnement de l'algorithme et en aucun cas il ne sert de validation.

Une **ontologie** en informatique est un ensemble structuré de concepts permettant de donner un sens aux informations.

Les concepts sont organisés dans un graphe dont les relations peuvent être :

- des relations sémantiques ;
- des relations de composition et d'héritage (au sens objet)

L'objectif premier d'une ontologie est de modéliser un ensemble de connaissances dans un domaine donné.

FIGURE 3.2 – Page Web version 1

L'unique document constituant le corpus est constitué de 56 mots (la ponctuation est ignorée). Parmi tous ces mots, seuls ceux contenant de l'information donneront lieu à la définition d'un concept dans l'ontologie. Nous entendons par là que tous les mots vides, comme les articles indéfinis, sont exclus.

Sur la figure représentant l'ontologie, les valeurs vertes représentent le poids sémantique, les valeurs mauves les distances sémantiques entre les concepts et en rouge figure la valeur de la résistance aux changements associée à chaque relation. Les valeurs des différents coefficients sont obtenues par application des différentes métriques. Par exemple, le poids sémantique associé au concept ontologie est en fait l'importance du concept dans le corpus car l'ontologie est la première version (aucune version précédente n'existe)

$$Imp(ontologie) = \lceil 3.(tf(ontologie) + \frac{Rep(ontologie)}{card(K)}) + 1 \rceil = \lceil 3.(\frac{2}{56} + \frac{21}{56}) + 1 \rceil = 1$$

. De la même façon, la distance sémantique entre ontologie et connaissance

$$SD(ontologie, connaissance) = \lceil 10.\frac{8}{56} + 1 \rceil = 2$$

1. http ://www.wikipedia.org/

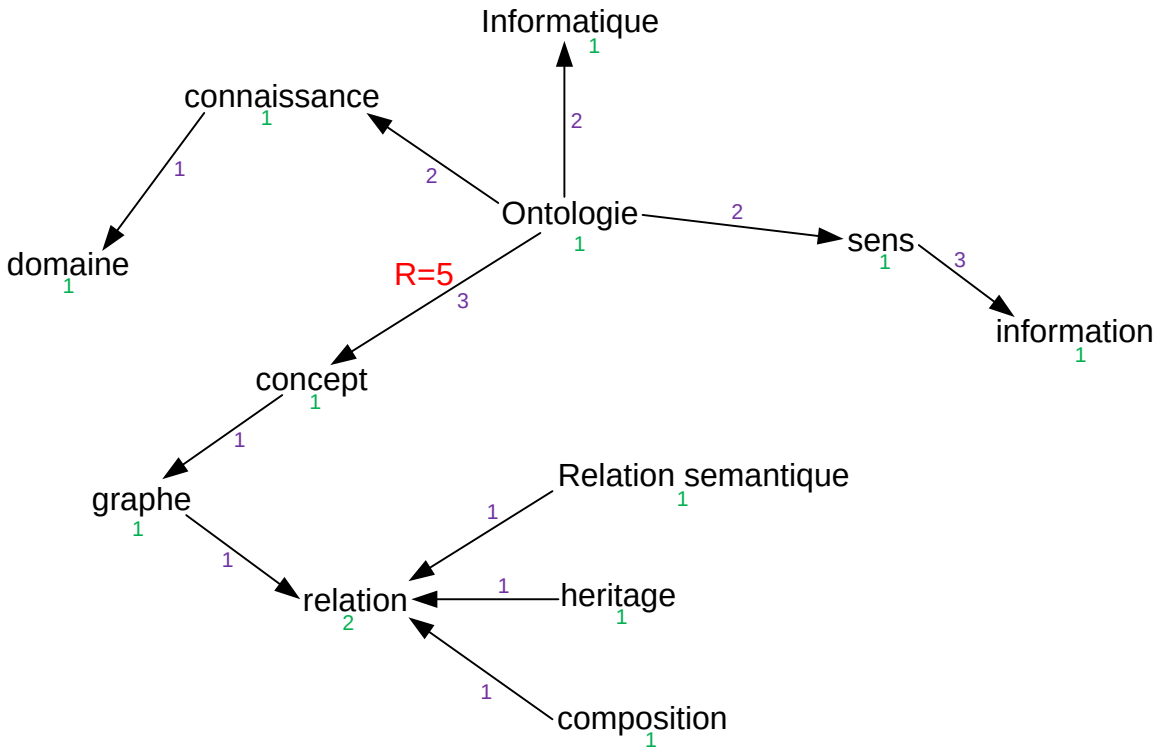


FIGURE 3.3 – Ontologie associée à la page Web version 1

Remarque: Les labels des relations ainsi que certains coefficients ont été volontairement omis pour des raisons de lisibilité.

En informatique et en science de l'information, une **ontologie** est un ensemble structuré de concepts permettant de donner un sens aux informations. Elle est aussi un modèle de données qui représente un ensemble de concepts dans un domaine et des rapports entre ces concepts. Elle est employé pour raisonner au sujet des objets dans ce domaine.

Les concepts sont organisés dans un graphe dont les relations peuvent être :

- des relations sémantiques ;
- des relations de composition et d'héritage (au sens objet)

L'objectif premier d'une ontologie est de modéliser un ensemble de connaissances dans un domaine donné.

On peut aussi dire que les ontologies sont employées dans l'intelligence artificielle, le Web sémantique, le génie logiciel, l'informatique biomédicale et l'architecture de l'information comme une forme de représentation de la connaissance au sujet d'un monde ou d'une certaine partie de lui

FIGURE 3.4 – Page Web version 2

L'évolution de la page Web a porté le nombre de mots de la page à 137, entraînant une évolution des valeurs des coefficients comme le montre l'ontologie de la figure 3.5. Les termes pouvant donner lieu à un nouveau concept de l'ontologie et leur potentiel sont représentés dans

le tableau suivant :

Termes	Potentiel
science	1
modèle	1
données	2
rapports	1
sujet	2
intelligence	1
artificielle	1
web	1
sémantique	1
génie	1
logiciel	1
biomédical	1
architecture	1
forme	1
représentation	1
monde	1
partie	1

TABLE 3.1: Termes et potentiels

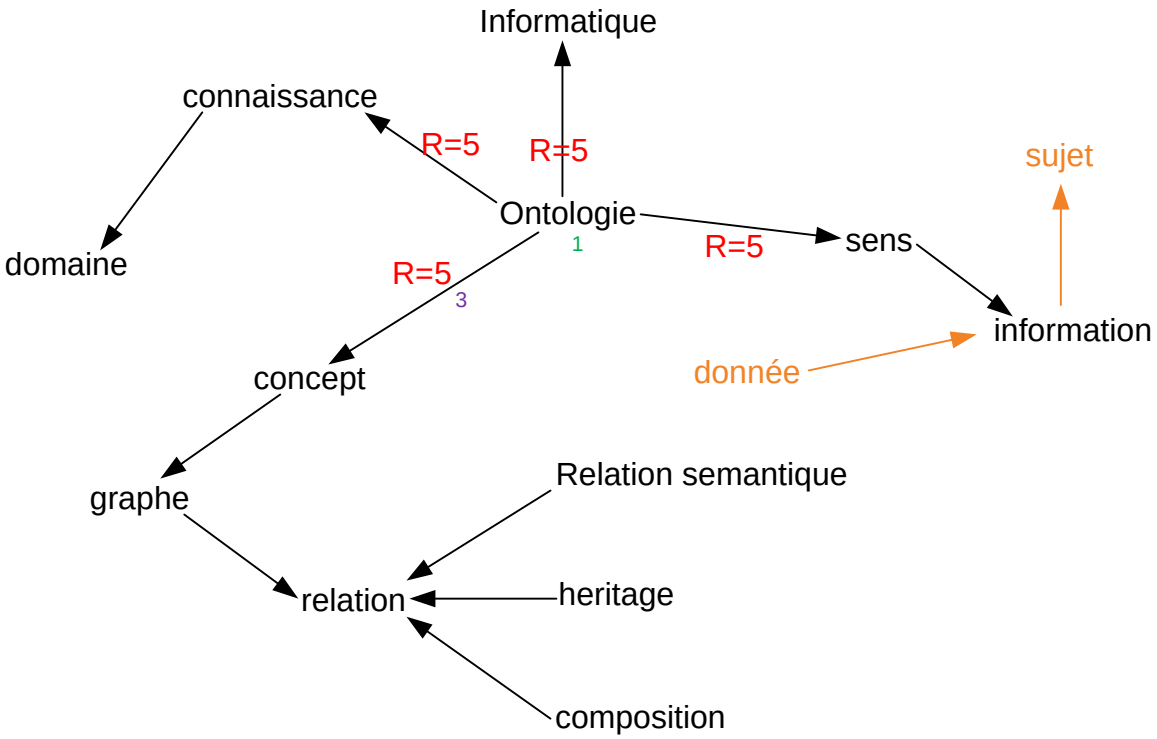


FIGURE 3.5 – Ontologie associée à la page Web version 2

L'évolution des connaissances se traduit au niveau de l'ontologie par l'ajout de nouveaux

concepts (en orange sur la figure 3.5) et l'adaptation de la valeur des coefficients existants. Par exemple, l'adaptation du poids sémantique du concept *ontologie* est maintenant de :

$$SW(ontologie_2) = \lceil SW(ontologie_1) + \frac{Imp_{K_2}(ontologie_2) - SW(ontologie_1)}{R_{ontologie_2}} \rceil = \lceil 1 + \frac{1-1}{5} \rceil = 1$$

. La distance sémantique entre les concepts *ontologie* et connaissance est maintenant de :
$SD(ontologie_2, connaissance_2) = \lceil SD(ontologie_1, connaissance_1) +$
$\frac{SD_{K_2}(ontologie, connaissance) - SD(ontologie_1, connaissance_1)}{R_r} \rceil = \lceil 2 + \frac{2-1}{5} \rceil = 2.$

L'évolution de la distance sémantique sur cet exemple montre bien le rôle de la résistance qui empêche la réduction de cette distance.

Remarque: Pour des raisons de lisibilité, sur la figure 3.5 n'apparaît qu'une partie des éléments qui devraient y figurer.

3.4 Impacts des règles d'adaptation sur les ontologies adaptatives

L'application des règles d'adaptation et des métriques va potentiellement modifier la structure de l'ontologie. L'ajout et la suppression de concepts et de relations nécessitent la reconstruction partielle de l'ontologie. Dans cette section nous discutons de l'impact des règles et des métriques sur les éléments de l'ontologie adaptatives.

3.4.1 Les éléments de l'ontologie liés à l'adaptation

Le processus d'adaptation que nous proposons s'appuie sur un corpus de documents afin de déterminer les évolutions du domaine à reporter au niveau de l'ontologie. Cependant, l'évolution des connaissances est parfois chaotique et de ce fait, l'analyse du corpus ne permet pas de caractériser complètement l'évolution. L'évolution des valeurs des coefficients d'une ontologie adaptative est révélatrice d'un ensemble de modifications potentielles à apporter à l'ontologie en plus de celles prises en charge dans l'algorithme.

Le poids sémantique se révèle être un indicateur important pour prévenir une suppression des concepts de l'ontologie. Un poids sémantique demeurant important sur une durée significative souligne l'importance du concept dans l'ontologie. En revanche, si la durée de validité du concept expire, le concept sera automatiquement retiré de l'ontologie en vertu du processus d'adaptation. L'expert en charge de l'ontologie peut alors intervenir sur la valeur de la résistance aux changements pour faire persister le concept dont le poids sémantique est élevé et empêcher sa suppression.

La distance sémantique, quant à elle, est intéressante pour la généralisation ou la spécialisation de concepts. Un ensemble de concepts relié par des relations dont la distance sémantique est petite sur une période de temps significative, nécessite peut-être son remplacement par un ou plusieurs concepts plus généraux ou plus spécifiques suivant les besoins (voir la sous-section suivante). Tout comme pour l'évolution du poids sémantique, c'est à l'expert du domaine de définir ces besoins et de faire les ajustements qui s'imposent sur les bons paramètres de l'ontologie.

3.4.2 Impact sur la structure de l'ontologie

Dans cette section, nous discutons de l'impact de l'ajout et de la suppression d'éléments ontologiques sur la structure de l'ontologie. Il est important que les propriétés structurelles de

l'ontologie soient conservées afin que l'ontologie puisse être réutilisée dans des tâches de raisonnement. Ce problème peut s'apparenter à celui de l'ajout ou de la suppression de nœuds dans un graphe orienté pondéré.

L'ajout de concept est représenté sur la figure 3.6. L'opération consiste à définir le nouveau concept, dans notre exemple le concept E en rouge, et les relations, représentées par des flèches rouges sur la figure, permettant de relier le concept à ceux de l'ontologie.

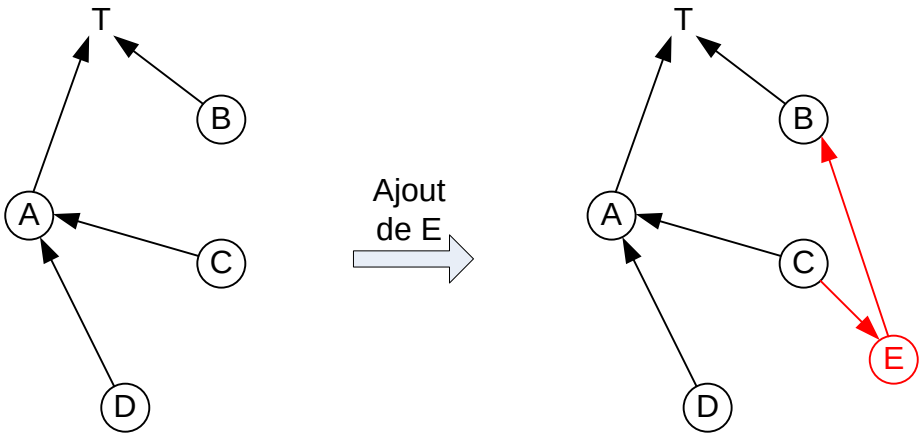


FIGURE 3.6 – L'ajout de concepts

La suppression d'éléments ontologiques est plus problématique. Nous distinguons trois cas principaux pour la suppression de concepts et la reconstruction d'une partie (ou de la totalité) de l'ontologie.

Le premier cas correspond au cas ou le concept à supprimer se trouve relié à un seul autre concept comme le montre la figure 3.7. Dans ce cas le plus basique, le concept est simplement supprimé car aucune reconstruction de l'ontologie n'est nécessaire.

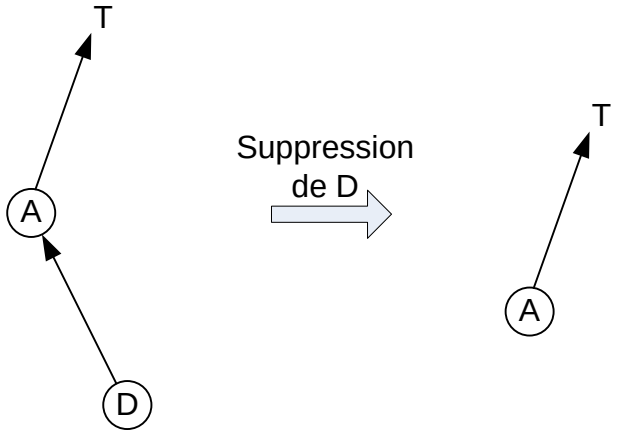


FIGURE 3.7 – Suppression de concept : cas de base

Le deuxième cas est plus complexe. Il intervient lorsque le concept à supprimer se trouve relié à plusieurs autres concepts. Lors de cette opération, une reconstruction partielle de l'ontologie

peut être effectuée en s'appuyant notamment sur les relations liant le concept supprimé aux autres concepts. Plusieurs éventualités doivent être considérées (voir figure 3.8).

La première d'entre elles est celle dans laquelle trois concepts A, B et C sont impliqués et il existe une relation r_1 entre B et A et une relation r_2 différente de r_1 entre C et B. Il est important de considérer la signature de la relation (domaine et codomaine) pour comprendre la reconstruction. La suppression du concept B va entraîner le rattachement du concept C au super concept car aucune information sur les relations impliquées ne permet de relier C et A. Une autre relation devra être définie par l'expert pour relier A et C.

La deuxième éventualité est celle où une reconstruction plus fine est possible. Dans cette situation, les trois concepts A, B et C sont reliés par des relations r_1 et r_2 transitives et de même type comme par exemple la relation de subsumption. La suppression de B entraîne le rattachement direct de C à A par la relation r_1 ou r_2.

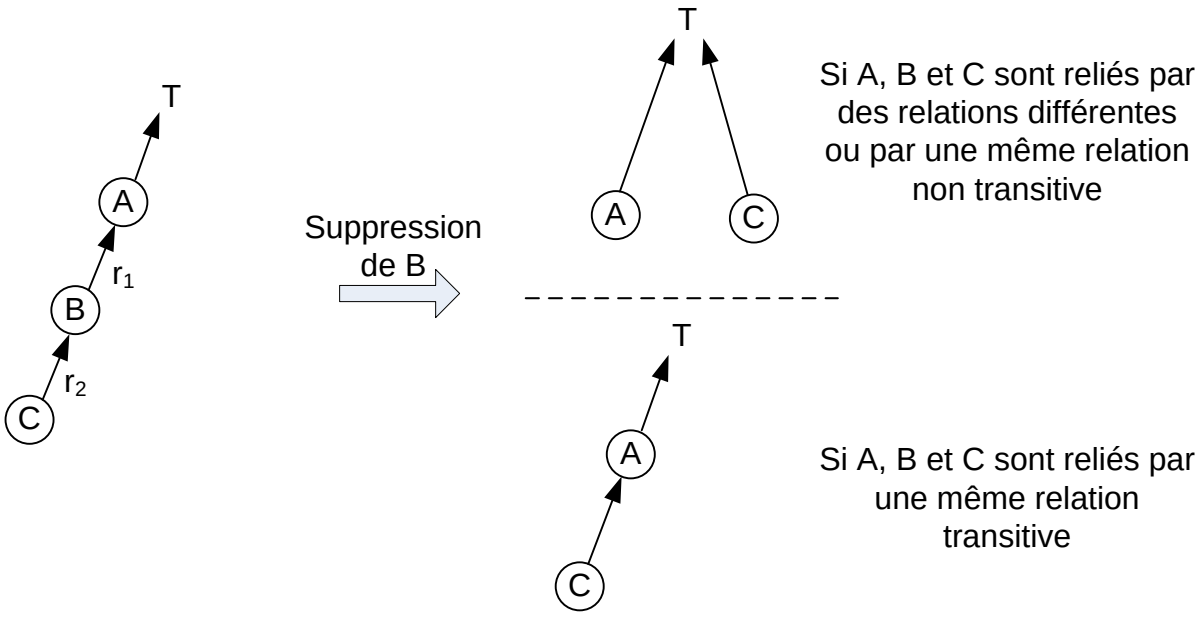


FIGURE 3.8 – Suppression de concept : reconstruction

Il existe d'autres cas pouvant survenir lors de la suppression de concept. La complexité de la reconstruction va dépendre du modèle de l'ontologie utilisé et principalement des propriétés du modèle pour définir les relations ontologiques. Les cas que nous discutons sont principalement ceux qui font intervenir la relation de subsumption commune à la grande majorité des modèles ontologiques. Cependant, la suppression de concepts et la reconstruction de l'ontologie en s'appuyant sur la relation de subsumption peut entraîner la création de cycles. Ce cas de figure doit donc être traité avec attention afin de préserver les propriétés de l'ontologie.

Le troisième cas concerne les instances du concept supprimé. Comme le montre la figure 3.9, deux cas de figures peut survenir. Dans le premier cas, le concept supprimé est subsumé par un autre concept. Dans ce cas, les instances du concept supprimé peuvent être rattachées au concept plus général. Dans le second cas, où aucune subsumption n'existe entre le concept supprimé et un autre concept, rattacher les instances à d'autres concepts n'a aucun sens. Les instances sont donc supprimés en même temps que le concept.

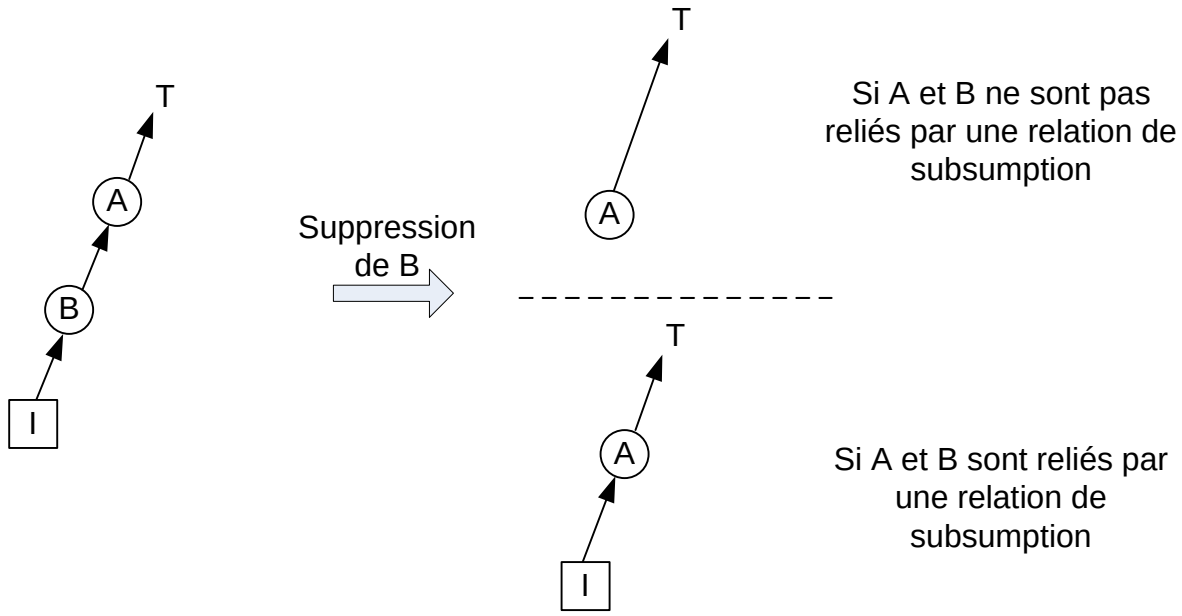


FIGURE 3.9 – Suppression de concept : cas des instances de concepts

L'ajout et la suppression de concepts sont suffisants pour expliquer les évolutions de l'ontologie lorsque survient le phénomène de généralisation et spécialisation des connaissances (voir chapitre 2). Au cours du processus de généralisation ou spécialisation des connaissances, certaines connaissances sont supprimées de l'ontologie pour laisser place à des concepts plus généraux ou plus spécifiques. Par conséquent, nous avons bien une composition d'ajouts et de suppressions de concepts et de relations.

3.5 Conclusion

Dans ce chapitre, nous avons présenté le processus d'adaptation des ontologies adaptatives dont le modèle a été introduit au chapitre précédent. Ce processus est directement inspiré des idées émises par Piaget dans le cadre de ces travaux sur l'adaptation des connaissances. Il est régit suivant un ensemble de six règles basées sur le phénomène de régulation, pierre angulaire de la théorie piagetienne. La définition des règles met en œuvre des métriques dont l'objectif est d'adapter les coefficients des éléments ontologiques du modèle d'ontologies adaptative à l'évolution des connaissances contenues dans un corpus de documents.

Les règles régissant le processus respectent le phénomène de régulation et de ce fait sont suffisamment générales afin de pouvoir définir des stratégies pour l'évolution d'ontologie. L'expert est par définition la personne la plus à même à définir cette stratégie vu qu'il doit avoir une idée précise de l'évolution des connaissances de son domaine. Les éléments mis en œuvre dans les différentes métriques ont été suffisamment discutés pour que l'expert puisse définir d'autres métriques plus spécifiques au domaine.

Le processus d'adaptation est également suffisamment précis pour servir de base au développement d'un outil pour l'automatisation de l'adaptation des ontologies. Les règles bien que générales peuvent facilement être implémentées tout comme les différentes métriques qui ont été formalisées.

Nous allons, dans les chapitres à venir, démontrer l'utilité des ontologies adaptatives et du processus d'adaptation qui leur est associé dans l'amélioration de la recherche d'information sur le Web.

Chapitre 4

Des ontologies adaptatives pour la représentation des données du Web

Sommaire

L'utilisation d'ontologies pour représenter la notion de contexte en recherche d'information a donné lieu à de nombreux travaux (voir section 1.3.1). Notre étude a toutefois souligné un ensemble de lacunes parmi lesquelles figurent les problèmes liés à la prise en compte de l'évolution de ce contexte. Les propriétés des ontologies adaptatives pour représenter le contexte et ses évolutions permettent de combler cette lacune. Dans notre approche, le contexte est défini comme étant le domaine de recherche visé par une requête, restreint grâce aux caractéristiques de l'utilisateur posant la requête. Il convient alors de considérer ce contexte sous deux angles différents : le domaine et les vues de l'utilisateur sur ce domaine puis d'exprimer toutes leurs caractéristiques grâce aux ontologies adaptatives.

Tout d'abord, le domaine de recherche visé par une requête présente des spécificités qu'il est important de bien comprendre. Les évolutions de ce domaine et principalement la fréquence

des changements sont, dans la majorité des cas, porteuses d'informations sur les spécificités du domaine. Il en va de même concernant les documents du Web le décrivant. Ces derniers sont généralement structurés de manière à ce que leur simple visualisation permet à l'utilisateur de déterminer son appartenance au domaine auquel il s'intéresse. Par conséquent, nous pensons que les ontologies adaptatives pour représenter le contexte défini par le domaine dans lequel la requête est soumise sont d'un réel apport.

La connaissance des utilisateurs est également une des composantes du contexte d'une requête supposée évoluer au cours du temps. La difficulté pour représenter les connaissances de chaque utilisateur relatives à un domaine précis en plus de problèmes éthiques liés à la collecte et à la conservation des données relatives à chaque utilisateur doivent cependant être traités. En conséquence, nous introduisons la notion de catégorie d'utilisateurs. Ces catégories sont définies en relation avec le domaine de recherche. En conséquence, les liens existants entre le domaine de recherche et les différents profils des catégories d'utilisateurs (points communs, différences, etc) doivent être analysés pour une meilleure exploitation des ontologies adaptatives pour représenter les connaissances caractérisant le mieux chaque catégorie.

La principale utilisation des ontologies dans la vision du Web Sémantique est celle concernant l'annotation des ressources du Web. En plus des possibilités d'expression des ontologies «classiques», les ontologies adaptatives permettent de représenter des données propres à l'évolution d'un domaine. En conséquence, ces informations peuvent être utilisées pour enrichir les annotations du Web et de son contenu. Nous montrons à travers la définition des structures de données que sont les WPGraphs et W^3Graphs, l'exploitation des ontologies adaptatives pour décrire le contenu du Web de manière plus riche. Ces structures de graphes permettent une représentation enrichie du contenu du Web en mettant l'accent sur sa sémantique et non pas uniquement sur la structure d'hyperliens du Web comme c'est le cas dans la plupart des approches existantes (voir chapitre 1). Les propriétés de ces graphes vont permettre de faciliter la recherche d'information et d'améliorer la pertinence des résultats.

Ce chapitre, décrit l'utilisation des ontologies adaptatives pour représenter les données du Web. Nous traitons d'abord de la représentation des caractéristiques des ontologies adaptatives au niveau d'une ontologie OWL. Puis nous montrons comment une telle ontologie OWL permet de décrire le contexte de la recherche. Nous montrons comment représenter un domaine de recherche ainsi que les vues des utilisateurs sur un domaine. Nous montrons ensuite comment les ontologies adaptatives peuvent être utilisées pour décrire le contenu du Web à travers la définition de structures spécifiques : les WPGraphs et W^3Graphs.

4.1 Représentation de l'évolution au niveau de l'ontologie

La représentation des éléments de l'évolution au niveau de l'ontologie nécessite une réflexion sur le mode de représentation et sur les possibilités offertes par les différents langages ontologiques. Etant la pierre angulaire du Web Sémantique, un ensemble de standards a été défini pour la représentation d'ontologies (voir section 1.2.2). Dans notre approche, nous préconisons l'utilisation de OWL, principalement pour son expressivité et pour sa large utilisation par les concepteurs d'ontologie.

Dans le but d'exprimer les éléments concernant l'évolution introduits dans ce chapitre, nous proposons d'utiliser la balise owl :annotationProperty. Le langage OWL Full ne place aucune contrainte pour les annotations d'une ontologie. Il admet des annotations sur les classes, les

propriétés, les individus et les en-têtes d'ontologies, mais seulement aux conditions suivantes :

- Les ensembles de propriétés d'objets, de propriétés de types de données, de propriétés d'annotations et de propriétés d'ontologies doivent être mutuellement disjoints. Dans OWL DL, une annotation ne peut donc pas être en même temps une propriété de type de donnée et une propriété d'annotation.
- Les propriétés d'annotations doivent avoir un triplet de typage explicite de la forme : AnnotationPropertyID rdf :type owl :AnnotationProperty
- Les propriétés d'annotations ne doivent pas être utilisées dans les axiomes de propriété. Dans OWL DL, on ne peut donc pas définir des sous-propriétés ou des contraintes de domaine/image de propriétés d'annotations
- L'objet d'une propriété d'annotation doit être ou bien un littéral de donnée, un appel d'adresse URI, ou bien un individu.

Les éléments concernant l'évolution que nous avons définis dans ce chapitre ne servent qu'à annoter des concepts (i.e. des classes OWL), des relations (i.e. des propriétés OWL) ou des instances de concepts (i.e. individus OWL). Par conséquent, l'utilisation de la primitive owl :annotationProperty peut se faire dans le cadre de OWL DL, ce qui permet de garantir la décidabilité et la complétude du raisonnement basé sur ce type d'ontologies. De plus, OWL tolère l'utilisation des types de données définis dans la spécification du langage XML, ce qui nous permet de définir plus facilement les éléments d'évolution en tant qu'annotations.

Le phénomène d'émergence est sans doute le plus difficile à traiter du fait qu'il requiert, en plus de la saisie de la date d'émergence, la caractérisation complète de la nouvelle connaissance dans l'ontologie, c'est à dire la définition de nouveaux concepts, de nouvelles relations et de nouveaux individus. Dans cet objectif, nous préconisons l'utilisation de la méthode des ontologies différentielles établie par Bachimont [Bachimont et al., 2002] (voir section 1.1.2). La définition d'une propriété d'annotation OWL ayant pour type une date [1] est requise pour l'annotation de toutes les nouvelles classes et nouveaux individus de l'ontologie.

```
<owl:DatatypeProperty rdf:ID="Emergence_Date">
        <rdf:type rdf:resource="&owl;AnnotationProperty"/>
        <rdfs:range rdf:resource="&xsd;date"/>
</owl:DatatypeProperty>
```

Remarque : L'utilisation des types de données XML dans OWL est une des raisons techniques pour lesquelles nous utilisons un temps absolu dans notre approche.

De la même façon, les nouvelles classes et individus sont annotés par une propriété de type entier pour la persistance et le poids sémantique (voir ci-après),

```
<owl:DatatypeProperty rdf:ID="Persistence_Duration">
        <rdf:type rdf:resource="&owl;AnnotationProperty"/>
        <rdfs:range rdf:resource="&xsd;int"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="Semantic_Weight">
        <rdf:type rdf:resource="&owl;AnnotationProperty"/>
        <rdfs:range rdf:resource="&xsd;int"/>
</owl:DatatypeProperty>
```

1. & owl et & xsd sont des notations utilisées pour faire référence à des espaces de nommage

Enfin, deux propriétés, l'une de type entier et l'autre de type flottant, sont nécessaires pour annoter les relations avec la distance sémantique et la résistance.

```
<owl:DatatypeProperty rdf:ID="Semantic_Distance">
        <rdf:type rdf:resource="&owl;AnnotationProperty"/>
        <rdfs:range rdf:resource="&xsd;int"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="Resistance">
        <rdf:type rdf:resource="&owl;AnnotationProperty"/>
        <rdfs:range rdf:resource="&xsd;float"/>
</owl:DatatypeProperty>
```

Remarque : Les valeurs par défaut des différents artefacts pour l'évolution sont discutées plus largement au chapitre suivant.

Les spécificités de la syntaxe de la famille de langage OWL nous oblige à adopter certaines conventions lors de l'application des annotations précédemment définies. Comme l'application de la distance sémantique et de la résistance ne se fait que sur les relations ontologiques, un problème se pose lorsque surviennent des relations d'équivalence ou de subsumption. Ces deux relations particulières ne sont pas considérées comme des propriétés OWL mais sont décrites à travers l'utilisation de balises spécifiques (owl :subClassOf et owl :equivalentClass). Ainsi, lorsque ces types de relations ontologiques apparaissent dans l'ontologie, l'application des annotations se fait au niveau du concept le plus spécifique dans le cadre d'une subsumption et sur un des concepts équivalents dans le cadre d'une relation d'équivalence.

4.2 Ontologies adaptatives et domaine de recherche

Le domaine de recherche visé par une requête est un des types de contexte défini dans la littérature (voir section 1.3.1). La compréhension des éléments caractéristiques de ce domaine et leur exploitation va contribuer à l'amélioration de la recherche d'information sur le Web. La connaissance du domaine est l'élément principal. Sur le Web, elle est contenue dans les documents qui constituent le domaine. Ceux-ci ont des caractéristiques particulières suivant le domaine auquel ils appartiennent qui méritent d'être prises en compte dans sa représentation.

Dans cette section, nous étudions les propriétés d'un domaine de recherche. Nous détaillons les caractéristiques de sa connaissance du domaine, celles des documents le constituant et nous discutons de l'apport des ontologies adaptatives pour représenter ce type de contexte.

4.2.1 Caractéristiques du domaine de recherche

L'observation du Web et de son contenu permet de définir deux types de Web : un Web dit «professionnel» et un Web dit «loisir» [Guelfi et al., 2008]. Cette distinction s'appuie sur trois critères principaux :

- La thématique à laquelle se rapporte le contenu des documents du Web,
- Le mode d'accès aux données,
- L'évolution des données.

Le Web dit «professionnel» regroupe l'ensemble des ressources du Web dédiées aux personnes utilisant le Web comme un média pour exercer leurs activités professionnelles. Les sites Web de commerce en ligne, la bourse ou la gestion des chaînes d'approvisionnement en font partie. Deux types d'activités peuvent également être associés au Web professionnel. Une première activité a pour but d'informer les utilisateurs comme, par exemple, les sites rapportant les évolutions des cours de la bourse. L'autre activité donne à l'utilisateur un rôle plus important. Elle lui permet d'interagir avec les ressources. C'est principalement le cas des sites de commerce électronique.

L'accès aux données du Web professionnel est particulier. La grande majorité des données est considérée comme confidentielle. Leur accès est étroitement contrôlé à l'aide de mécanismes d'authentification basés sur des mots de passe. Ces données font partie du Web «caché» et ne sont pas accessibles pour les crawlers. Par conséquent, elles ne sont pas référencées par les moteurs de recherche usuels. La partie publique du Web professionnel est généralement générée automatiquement grâce à l'utilisation de scripts à partir des ressources stockées dans des bases de données.

Le Web professionnel regroupe des données très importantes pour les entreprises auxquelles elles appartiennent. L'évolution de son contenu est donc primordiale pour la survie de ces entreprises. Cette évolution est généralement bien définie. Les données évoluent de manière régulière selon les besoins des utilisateurs. Les évolutions des cours de la bourse sont rapportées à intervalles de temps très resserrés (i.e. toutes les deux minutes). La période de temps est définie et portée à la connaissance des utilisateurs de l'application du Web professionnel pour des raisons évidentes d'exploitation commerciale.

Le Web dit «loisir» représente l'ensemble des ressources du Web destinées aux personnes utilisant le Web pour des activités de loisir. On y retrouve, par exemple, les pages personnelles, les blogs ou les sites Web construits dans un but non professionnel. Les informations du Web loisir sont purement informatives comme peut l'être la page personnelle d'un chercheur. Elle recense ses activités de chercheur, ses tâches d'enseignement, le sujet de ses recherches, ses publications ou des données personnelles comme son adresse e-mail, son numéro de téléphone et son affiliation.

Dans la mesure où la grande majorité du Web loisir est purement informative, toutes ses informations sont publiques et exprimées grâce aux standards du Web comme HTML. Ces données peuvent être consultées par les crawlers des moteurs de recherche usuels et être indexées par ces derniers.

L'évolution des données du Web loisir est moins importante que celle des données du Web professionnel du fait du caractère très peu critique des données du Web loisir. La fréquence des changements ainsi que les intervalles de temps les séparant ne sont pas rigoureusement définis. Les modifications des données des pages personnelles et des blogs sont totalement irrégulières. De plus, les nouvelles technologies utilisées dans les blogs ou les wikis facilitent la modification des données mais renforcent le caractère irrégulier de l'évolution. Dans le cas extrême, certains documents du Web loisir sont amenés à ne jamais évoluer. C'est le cas des sites Web développés sur un sujet précis, comme la description d'un fait historique.

La catégorisation du contenu des documents du Web permet une première distinction du domaine de recherche. Il est alors nécessaire d'analyser attentivement la structure ainsi que les types d'objets constituant ces documents.

4.2.2 Les documents relatifs à un domaine de recherche

Les documents relatifs à un domaine ont des caractéristiques qui dépendent du type de Web (professionnel ou loisir) auquel ils appartiennent.

La structure des documents du Web professionnel est plus rigoureuse que celle du Web loisir. Cette différence s'explique principalement par l'utilisation de standards comme XML ou de *templates* dont le rôle est de structurer rigoureusement les données. La structure du Web professionnel est basée sur l'observation suivante : *les utilisateurs survolent rapidement le contenu de la page pour accéder rapidement à l'information pertinente*. Par conséquent, les données importantes des documents du Web professionnel doivent être bien mises en évidence au travers de la structure du document. L'information principale est localisée dans les titres ou mise en évidence par l'utilisation de balises dédiées, comme la balise <b> de HTML pour mettre le texte en caractères gras. L'utilisation massive de métadonnées est également fréquente dans les documents du Web professionnel. Ces métadonnées enrichissent le contenu, ce qui a pour effet d'augmenter le score de la page calculé par les moteurs de recherche (voir section 1.3.1).

La structure du Web loisir est beaucoup plus difficile à définir du fait de l'utilisation simple de HTML pour écrire les documents. L'exploitation des constructeurs du langage est peu rigoureuse. La mise en évidence du contenu des documents est souvent disproportionnée et ne reflète pas suffisamment les informations importantes contenues dans la page Web. L'utilisation des métadonnées est moindre car l'intérêt pour ces pages d'être retournées en priorité par les moteurs de recherche n'est pas important.

La forme sous laquelle est exprimée le contenu du Web professionnel est de deux types différents selon que le site Web est à but informatif ou actif. Les sites Web informatifs sont majoritairement constitués de textes. Les sites Web actifs sont développés pour que les utilisateurs puissent interagir avec les systèmes d'information de l'entreprise. Pour cette raison, le contenu de ces pages Web est décrit à l'aide de formulaires, de menus déroulant, etc. L'utilisation de textes ou de formulaires est donc révélateur de l'appartenance d'une page au Web professionnel.

Le contenu du Web loisir est davantage basé sur l'utilisation des technologies multimédia. Dans ces documents, on retrouve surtout des images, des fichiers audio ou des vidéos, contrairement au Web professionnel. Néanmoins, le contenu de ce type d'objets est incompréhensible pour les machines. Par conséquent, des métadonnées doivent être rajoutées pour pallier à ce manque.

Exemple: Dans le domaine de la recherche scientifique, notre expérience personnelle nous permet de dire que la grande majorité des documents sont des articles scientifiques essentiellement constitués de texte et respectant une même structure. Cette structure est liée à la façon dont la plupart des chercheurs parcourent un article. Concrètement, les chercheurs lisent dans l'ordre :

- Le **titre** de l'article : Cette partie est la plus importante du document car elle synthétise tout le contenu. En lisant le titre, le lecteur sait s'il a une chance de trouver l'information qu'il recherche.
- Les **informations sur les auteurs** : Les données contenues dans cette partie sont localisées sous le titre. Elles fournissent des informations sur le type de recherche scientifique détaillée dans le papier. Par exemple une affiliation à une entreprise sous entend une recherche plus appliquée alors qu'une affiliation à une institution universitaire laisse présager une recherche plus fondamentale. Il en est de même concernant les auteurs de l'article. Les chercheurs sont connus pour leurs contributions dans des domaines précis. Lorsque le lecteur lit et reconnaît le nom de l'auteur, il peut en déduire le domaine auquel la recherche du papier

fait référence.
- Le **résumé** : C'est la partie la plus importante de l'article. Elle contient, comme son nom l'indique, un résumé du contenu du papier. Le lecteur peut, suivant les informations contenues dans le résumé, définir avec plus de précisions si l'article l'intéresse ou non.
- Les **mots clés** : Cette partie de l'article permet de caractériser le domaine auquel la recherche présentée dans l'article fait référence. Les mots clés reprennent en général certains mots du titre et du résumé.
- Les **titres des sections** : L'information contenue dans les titres de sections permet de localiser, dans l'article, les données évoquées dans le résumé.
- Les **objets graphiques** (tableaux, figures,ect) ou **section spéciales** (définition, théorème, etc) : Ils permettent également de caractériser le type de contenu. Par exemple des théorèmes et des définitions sont caractéristiques de la recherche fondamentale alors que des figures font référence à des travaux plus appliqués.

Cet exemple montre qu'au domaine de la recherche scientifique est associée une structure de documents bien particulière, qui peut être exploitée pour la recherche d'information.

Dans cette section, nous avons montré qu'a un domaine est associé une connaissance qui est décrite dans un ensemble de documents dont la structure présente des caractéristiques intéressante pour l'amélioration de la recherche d'information. L'exploitation des ontologies adaptatives pour la représentation de ce domaine doit tenir compte de la connaissance du domaine mais également des caractéristiques liées à leur structure.

Au cours de la section suivante, nous allons détailler le processus de construction des ontologies adaptatives du domaine de recherche. Nous montrerons comment construire l'ontologie représentant la connaissance du domaine mais aussi celle dédiée aux documents. Nous allons montrer ensuite, en sections 4.4.1 et 4.4.1, l'exploitation de ces ontologies adaptatives pour l'enrichissement du contenu des pages Web et l'amélioration de la recherche de documents sur le Web.

4.2.3 Utilisation des ontologies adaptatives pour la représentation d'un domaine de recherche

Le domaine de recherche visé par une requête est en constante évolution du fait de la nature dynamique du Web, en constante évolution. Les travaux portant sur l'utilisation des ontologies pour la représentation d'un domaine de recherche ne prennent pas en compte l'évolution de ce domaine. Les ontologies adaptatives qui permettent de modéliser les caractéristiques d'évolution d'un domaine sont une solution. Dans cette section, nous allons discuter de l'utilisation des ontologies adaptatives pour la description d'un domaine à travers une méthodologie de construction.

Méthodologie de construction

La construction de l'ontologie représentant les connaissances d'un domaine de recherche est à la charge d'un expert de ce domaine. Cette construction doit se faire de manière rigoureuse suivant une méthodologie de construction (voir section 1.1.2). La méthodologie des ontologies différentielles de Bachimont (voir section 1.1.2) présente l'avantage de s'appuyer sur des idées constructivistes, comme le sont celles de Piaget sur lesquelles nous avons défini les ontologies adaptatives, et sur un corpus de documents concentrant la connaissance du domaine. Le modèle des ontologies adaptatives contient des éléments spécifiques c'est pourquoi, l'expert doit

définir les valeurs de ces éléments dans les étapes de la méthode des ontologies différentielles et principalement dans la phase de normalisation sémantique.

Exemple

L'exemple que nous détaillons illustre la méthodologie de construction des ontologies (domaine et document) que nous proposons. Il est basé sur un cas d'étude dont le domaine de recherche est celui de la conférence World Wide Web (voir chapitre 7). L'ontologie des connaissances du domaine à un moment précis du temps est construite sur la base de l'appel à communication pour la conférence à ce moment du temps et du contenu des articles acceptés pour publication lors des événements antérieurs à ce moment. L'ensemble de ces documents constituent le corpus de travail. Par exemple, pour construire l'ontologie de la conférence WWW 1998, nous avons utilisé l'appel à communication de WWW 1998 et l'ensemble des articles acceptés aux conférences WWW antérieures à celle de 1998. L'expert chargé de construire cette ontologie peut, par exemple, être un des membres du comité de programme de la conférence. L'ontologie représentant la connaissance du domaine est exprimée en OWL.

L'ensemble des concepts de l'ontologie est déterminé à partir des appels à communication. Chaque «topic» mentionné dans l'appel donne lieu à un nouveau concept dans l'ontologie. La date d'émergence des concepts est fixée à la date de publication de l'appel et la période de validité est déterminée arbitrairement dans cet exemple.

La hiérarchie des concepts, les relations les liant ainsi que la distance et le poids sémantiques sont établis sur la base des informations contenues dans les articles des années précédentes principalement dans *le résumé* de ces documents. Les relations sont déterminées par une analyse linguistique du contenu des papiers alors que le poids et la distance sémantique sont obtenus par application des métriques présentées au chapitre 3. Enfin, la résistance aux changements est fixée à 1 pour laisser l'ontologie suivre les futures évolutions des connaissances du corpus. L'ontologie représentant les connaissances du domaine de la conférence WWW 1998 est représentée sur la figure 4.1.

L'ontologie des documents du domaine est construite par l'expert. Elle décrit la structure des documents du domaine (voir l'exemple de la section 4.2.2). Les labels des concepts de l'ontologie font référence aux parties de document contenant l'information pertinente. Cette ontologie sera, par la suite, exploitée pour l'enrichissement du contenu de ces documents (voir section 4.4.1). La figure 4.2 représente l'ontologie des documents constituant le domaine de la recherche scientifique.

Remarque: Bien que le domaine de la conférence World Wide Web évolue sensiblement, la structure des documents décrivant ce domaine évolue peu. Par conséquent, l'utilisation des ontologies adaptatives pour décrire les documents du domaine n'est pas nécessaire vu que les données relatives aux documents sont stables et évoluent peu. Mais dans un souci d'homogénéité, nous les avons utilisées et fixé la résistance aux changements à 10 pour prévenir toutes évolutions.

4.3 Ontologies adaptatives et connaissances des utilisateurs sur un domaine

Les caractéristiques des utilisateurs sont également une des composantes du contexte d'une requête. Dans cette section, nous allons montrer l'intérêt d'utiliser les ontologies adaptatives pour la représentation des vues des utilisateurs sur un domaine de recherche. Nous allons tout d'abord présenter les relations existantes entre un domaine et les utilisateurs posant des requêtes

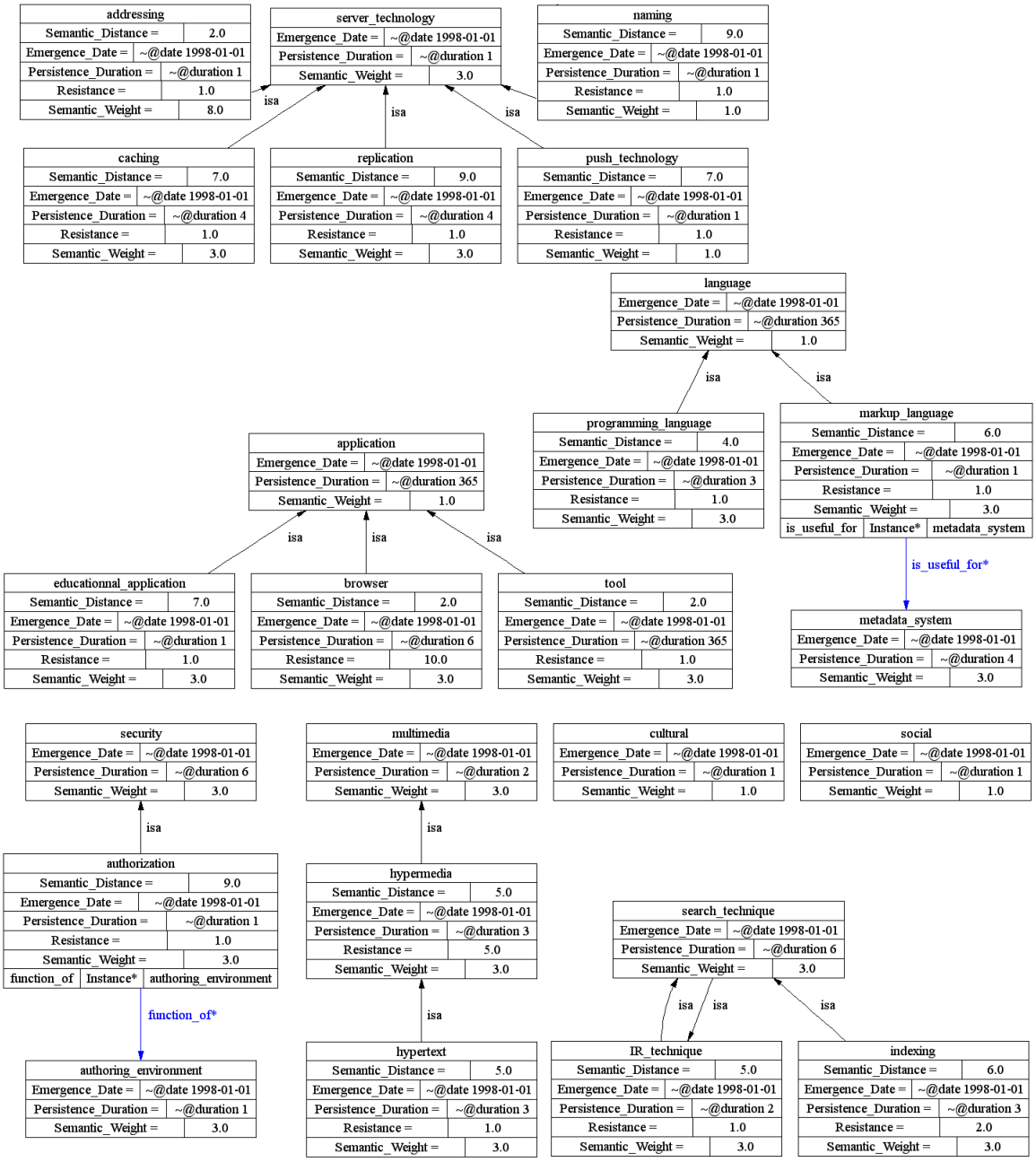


FIGURE 4.1 – Ontologie du domaine de la conférence WWW 1998

dans ce domaine et définir la notion de catégorie d'utilisateurs. Nous allons ensuite illustrer la construction d'une ontologie adaptative représentant des catégories d'utilisateurs à travers un exemple emprunté à notre étude de cas portant sur la conférence internationale World Wide Web.

4.3.1 Relations entre le domaine et ses utilisateurs

Dans le cadre de la recherche d'information sur le Web, les connaissances du domaine auquel se rapporte une requête et celles des utilisateurs posant la requête sur le domaine sont liées.

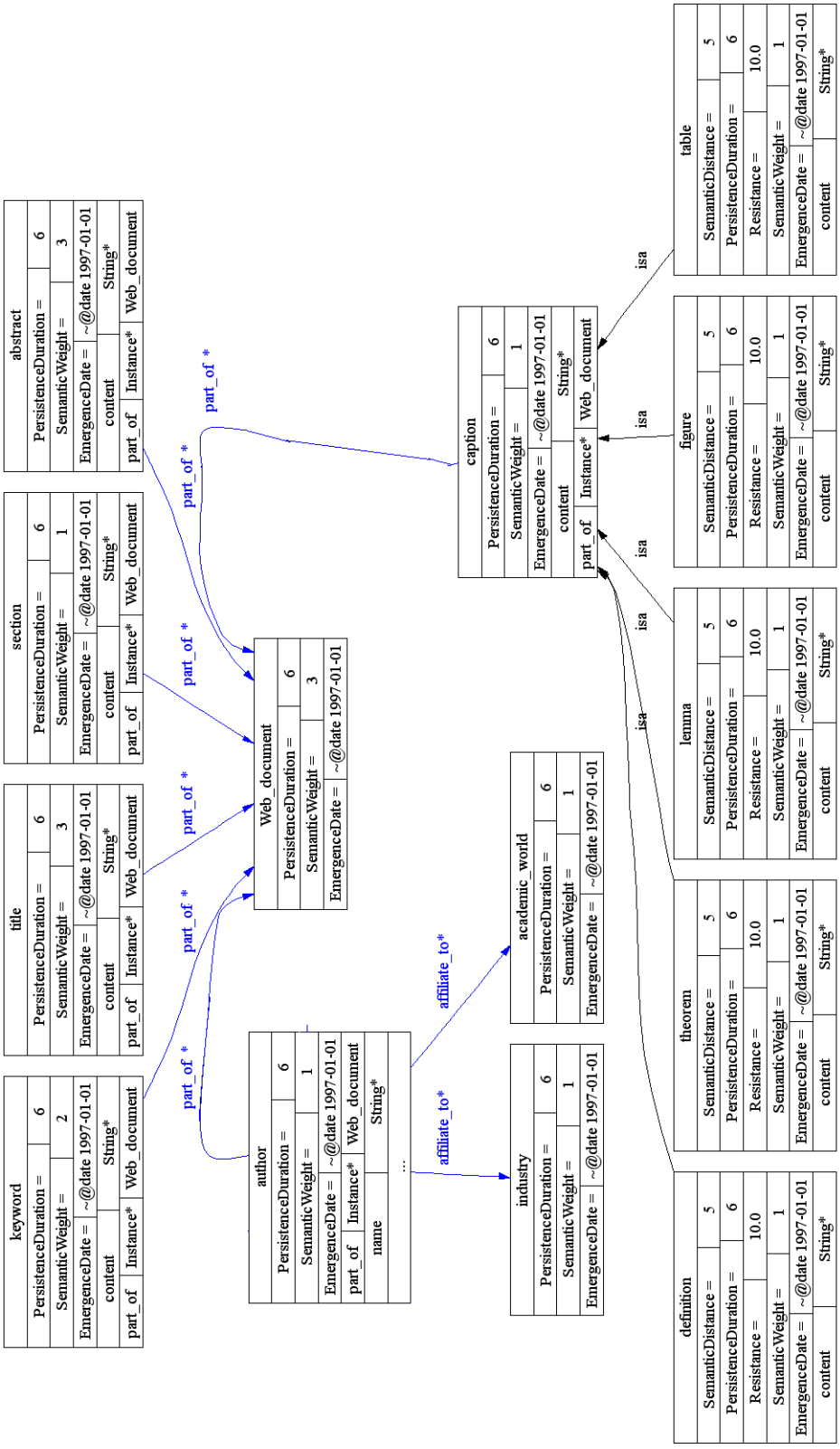


FIGURE 4.2 – Ontologie des documents du domaine de la conférence WWW

Comme le montre la figure 4.3, les connaissances d'un utilisateur couvrent (partiellement ou en totalité) celles du domaine associé à sa requête.

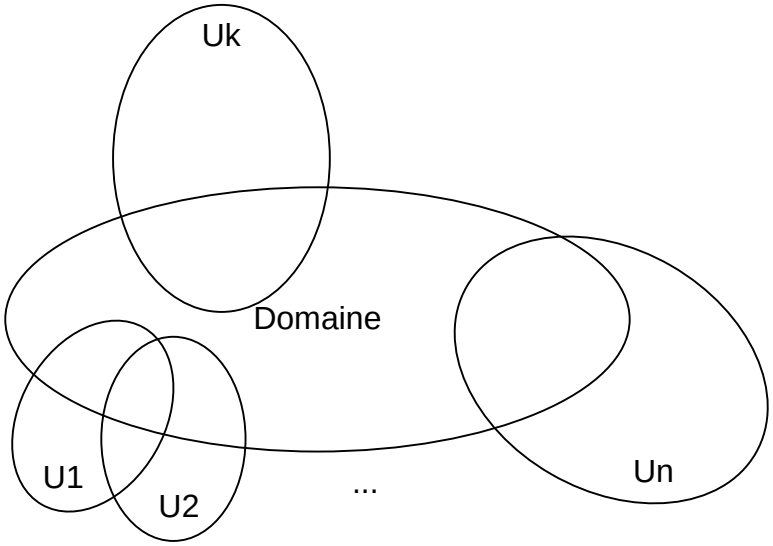


FIGURE 4.3 – Relations entre la connaissance du domaine d'une requête et celle d'utilisateurs posant la requête

La prise en compte des caractéristiques de l'utilisateur pose un certain nombre de problèmes. Tout d'abord, nous pensons que le profil d'un utilisateur doit représenter de manière très rigoureuse ses connaissances et ses centres d'intérêt. En effet, un profil de mauvaise qualité aura un impact négatif sur la pertinence des résultats retournés lors de l'évaluation de la requête enrichie par les informations du profil. Ensuite la collecte d'information sur l'utilisateur pour la gestion de son profil pose des problèmes d'ordre éthique. Pour contourner ces problèmes, nous proposons de considérer des catégories d'utilisateurs (voir figure 4.4).

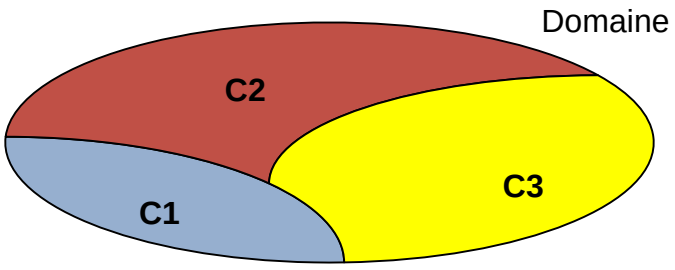


FIGURE 4.4 – Catégories d'utilisateurs

Une catégorie représente un point de vue partagé par un ensemble d'utilisateurs sur le domaine. Les catégories sont définies en considérant plusieurs aspects parmi lesquels :

– Le **nombre significatif** d'utilisateurs ayant un même point de vue : plus ce nombre sera important, et moins les connaissances collectées pour la catégorie fera référence à un utilisateur particulier. Ceci évite les problèmes éthiques mentionnés précédemment.

- Les **recouvrements** : le regroupement des connaissances du domaine pour la création d'une catégorie doit être fait de manière à ce que les connaissances soient communes à un nombre minimal de catégories (i.e. sur la figure 4.4, $C_1 \cap C_2 \cap C_3 = \emptyset$ dans le meilleur des cas). Le rattachement d'un utilisateur à une catégorie se fera alors sans ambiguïté.

Les catégories d'utilisateurs ainsi définies permettent de réduire le domaine de recherche associé à une requête.

Exemple 1 : Le domaine médical peut, par exemple, être subdivisé en deux sous-domaines :
- Le premier regroupant les connaissances communes aux médecins. Cette partie du domaine regroupe les connaissances caractéristiques que les professionnels de la santé maîtrisent. Les labels des concepts de l'ontologie associée à cette catégorie d'utilisateurs feront, par exemple, référence aux noms scientifiques des pathologies.
- Le second regroupant les connaissances des personnes malades n'ayant aucune formation médicale. Les labels des concepts de l'ontologie seront, par exemple, à choisir parmi les noms communs donnés aux maladies.

Exemple 2 : Le domaine de la recherche scientifique peut également se diviser en deux sous domaines :
- Le premier regroupant les connaissances pouvant intéresser un chercheur fondamental. Le profil des utilisateurs représentant cette catégorie sera composé de concepts faisant, par exemple, référence aux mathématiques.
- Le second regroupant les connaissances pouvant intéresser un chercheur appliqué. Le profil associé à ce type d'utilisateur contiendra plutôt des concepts faisant référence à des outils logiciels ou des expérimentations.

Nous proposons d'utiliser la notion d'ontologie adaptative pour représenter les profils de catégories d'utilisateurs et leur évolution.

4.3.2 Les ontologies adaptatives pour représenter les caractéristiques des utilisateurs

Tout comme pour les connaissances du domaine, celles associées aux différentes catégories d'utilisateurs évoluent au cours du temps. Les ontologies adaptatives offrent suffisamment d'expressivité pour la représentation des caractéristiques des utilisateurs. Elles permettent, par ailleurs, de représenter les évolutions des connaissances de chaque catégorie.

Construction d'une ontologie adaptative associé à une catégorie d'utilisateurs

La construction d'une ontologie adaptative associée à une catégorie d'utilisateur doit tenir compte des deux points définissant cette catégorie : le point de vue commun sur le domaine d'un nombre significatif d'utilisateurs et le non recouvrement des catégories.

C'est l'expert du domaine qui détermine les sous-domaines associés aux différentes catégories. Il connaît les caractéristiques des utilisateurs intéressés par son domaine, il peut donc les regrouper par catégorie.

La définition des labels de concepts contenus dans les différentes ontologies est important car il permet de distinguer les différentes catégories entre elles. Par exemple, dans le domaine médical le concept de la *grippe* pourra être désigné par *Myxovirus influenzae* dans l'ontologie

des médecins et par *grippe* dans celle des personnes non initiées. De plus, le poids sémantique attribué à chaque concept permet d'accentuer l'importance de certains concepts de l'ontologie en vue d'une future exploitation (voir section 5.2). La définition des autres coefficients est laissée à la discrétion de l'expert. Ils doivent néanmoins être déterminés en fonction des besoins pour lesquels l'ontologie est construite. L'expert peut, par exemple, choisir de ne pas laisser l'ontologie évoluer auquel cas, les coefficients de période de validité et de résistance aux changements seront élevés.

Exemple

Dans cette section, nous proposons un exemple pour illustrer la construction d'une ontologie décrivant les connaissances d'une catégorie d'utilisateurs. L'exemple, extrait de notre étude de cas, se propose de construire l'ontologie représentant la catégorie des chercheurs du domaine général de la recherche scientifique en informatique.

Notre expérience personnelle dans ce domaine nous permet de répartir les chercheurs en deux grandes catégories :

- Les chercheurs dits «fondamentaux» représentent la catégorie des chercheurs intéressés par la recherche fondamentale. Ces personnes sont majoritairement des universitaires ayant des connaissances avancées en mathématiques.
- Les chercheurs dits «appliqués» représentent les personnes ayant un profil d'industriel plutôt intéressées par la conception d'outils logiciels et leur utilisation pour des expérimentations.

La définition des concepts de l'ontologie, et principalement de leur label, doit permettre de distinguer les types de chercheurs, c'est pourquoi, nous l'avons réalisé sur la base des centres d'intérêt des personnes des différentes catégories :

- Concernant les chercheurs fondamentaux, les concepts formant l'ontologie font référence au monde des mathématiques. On y trouve des concepts comme *formal*, *theorem*, *algebra*, *logic*, etc. (voir figure 4.5).
- Concernant les chercheurs appliqués, les concepts sont *tool*, *application*, etc (voir Annexe C).

Les relations liant ces concepts ainsi que les coefficients assignés aux concepts et relations de l'ontologie adaptative ont été déterminés par l'expert du domaine. Dans l'exemple de la figure 4.5, nous avons choisi de figer l'ontologie d'où une période de validité et une résistance aux changements élevées.

4.4 Les ontologies adaptatives pour la représentation du Web et de son contenu

Une des spécificités du Web Sémantique consiste à utiliser des ontologies pour annoter des ressources du Web. Dans cette section, nous allons présenter une utilisation des ontologies adaptatives pour réaliser cette tâche à travers la notion de WPGraphs et W^3Graphs [Guelfi and Pruski, 2006]. Ces nouvelles structures de graphes, construites à l'aide d'une ontologie adaptative, ont pour but de représenter le contenu du Web en s'appuyant sur sa sémantique et non pas sur sa structure d'hyperliens. Cette représentation enrichie du contenu du Web a pour objectif l'amélioration de la recherche documentaire sur le Web.

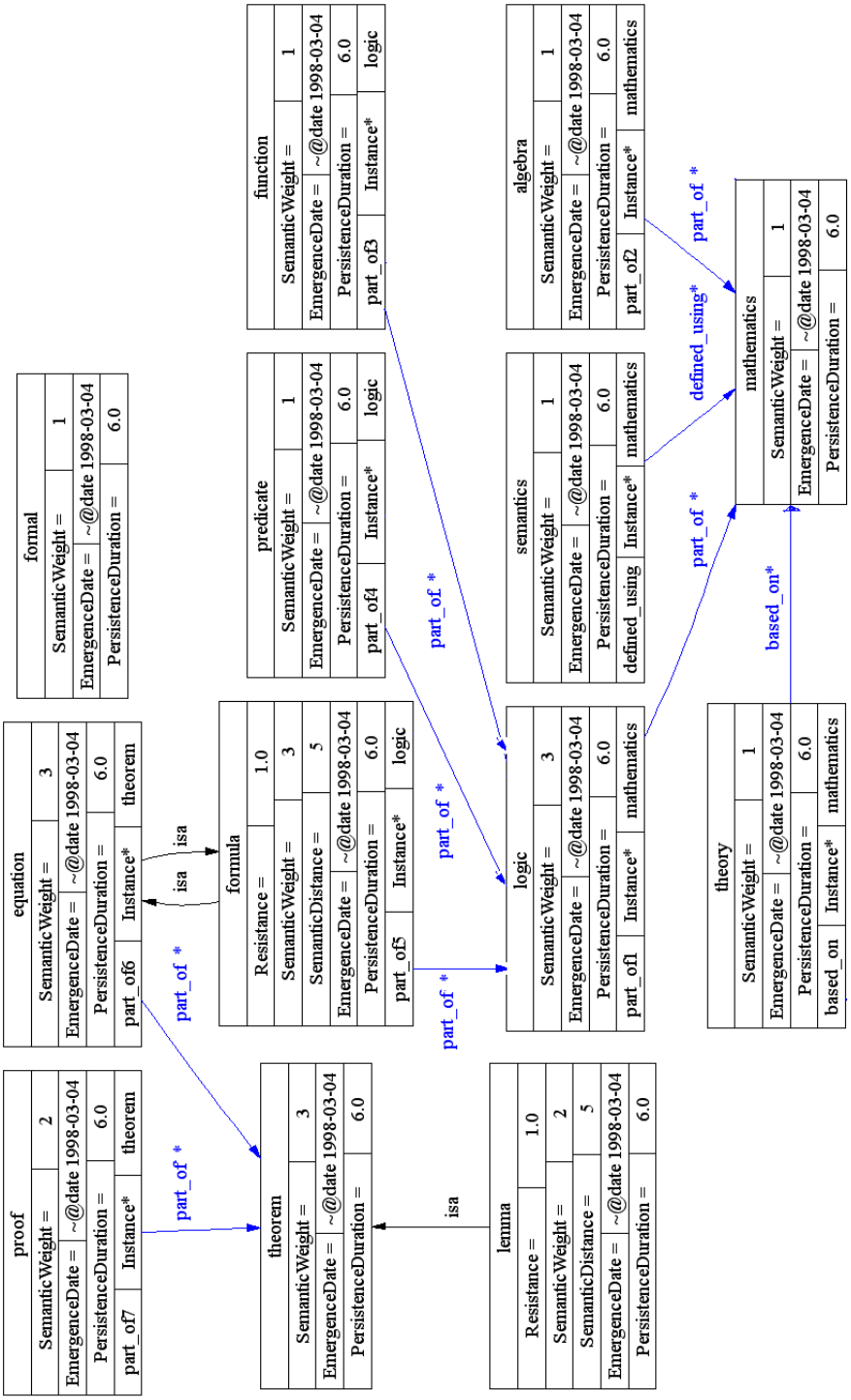


FIGURE 4.5 – Catégorie des chercheurs fondamentaux

4.4.1 Les modèles associés à ces structures

L'objectif de ces structures étant de représenter le contenu du Web, le modèle de ces graphes prend en compte les spécificités des données du Web et les combine à celles des ontologies adaptatives afin d'obtenir une représentation plus riche du contenu du Web.

Le modèle des WPGraphs

Un WPGraph est une structure représentant le contenu enrichi d'une page Web. La définition du modèle de WPGraph s'inspire des graphes conceptuels [Sowa, 1984] et s'appuie sur la théorie des graphes. Un WPGraph est construit à l'aide d'une ontologie du domaine et de celle des documents du domaine.

Définition 4.1 *Soit Σ un alphabet*[1]*, nous définissons GD le WPGraph associé à une page Web comme un graphe $(V, E, T, \varphi, \rho_v, \rho_e)$ où :*

1. $V = \{x | x \in \Sigma^+\}$ *est l'ensemble des sommets,*
2. $E = \{\{u, v\} | u, v \in V\}$ *est l'ensemble des arêtes,*
3. $T = \{\perp, img, vid, snd, fil\}$ *est l'ensemble des types de documents,*
4. $\varphi : V \longmapsto (\Sigma^*, T)$ *est une fonction de labélisation des sommets,*
5. $\rho_v : V \longmapsto \mathbb{R}^+$ *est une fonction de pondération des sommets,*
6. $\rho_e : E \longmapsto \mathbb{R}^+$ *est une fonction de pondération des arêtes.*

L'ensemble V des **sommets** d'un WPGraph contient les termes figurant sur la page Web, soit dans la partie visible de la page, soit en tant que métadonnées (partie du code HTML n'apparaissant pas à l'écran). Les termes sont sélectionnés parmi ceux porteurs d'information, nous excluons donc tous les mots comme les articles, les pronoms, certains adjectifs, etc.

L'ensemble E des **arêtes** du graphe est construit à partir de *l'ontologie adaptative du domaine* auquel se rapporte la page Web. Soient u et v deux sommets d'un WPGraph, **il existe une arête entre u et v si les concepts associés aux termes u et v sont reliés par un chemin dans l'ontologie**. Dans une ontologie, nous définissons le chemin comme une suite finie de relations consécutives reliant deux concepts, par analogie avec la théorie des graphes.

Définition 4.2 *Soient $O = (C, R, A, I)$ une ontologie, $c_1 \in C$ et $c_2 \in C$. On définit l'ensemble des chemins entre c_1 et c_2 de longueur d ($d \in \mathbb{N}$) par $Path(c_1, c_2) = \{(k_i)_{1 \leq i \leq d} | k_1 = c_1, k_d = c_2, \forall i, 1 \leq i \leq d-1, (k_i, k_{i+1}) \in R\}$. De plus, $\forall P \in Path(c_1, c_2)$, on note $l(P)$ la longueur de P.*

L'ensemble T des **types de documents** fait référence à la nature des objets que peut contenir une page Web : $\perp$ pour du texte, *img* pour des images, *vid* pour des vidéos, *snd* pour des sons et *fil* pour d'autres types de fichier.

La **fonction de labelisation** φ des sommets permet d'annoter le contenu des sommets par des informations complémentaires comme, par exemple, l'extension du fichier pour une vidéo.

La **fonction de pondération des sommets** ρ_v permet de mesurer l'importance d'un terme dans le WPGraph. Cette fonction prend en compte l'ontologie des documents du domaine.

Définition 4.3 *Soient p une page Web, $W = (V, E, T, \varphi, \rho_v, \rho_e)$ le WPGraph associé à cette page, $t \in V$ un terme et $O_{Doc} = (C, R, A, I)$ une ontologie adaptative :*

$$\rho_v(t) = Imp_p(t) + SW_{O_{Doc}}(t)$$

Plus la valeur de ρ_v associée à un sommet est importante, plus le terme du sommet est important dans le WPGraph et par conséquent dans la page Web.

La **fonction de pondération des arêtes** ρ_e permet de mesurer la distance sémantique entre deux sommets du graphe.

1. Σ^+ représente les mots non vides construits sur Σ, et Σ^* représente l'ensemble des mots construits sur Σ y compris le mot vide

Définition 4.4 *Soient p une page Web, $W = (V, E, T, \varphi, \rho_v, \rho_e)$ le WPGraph associé à cette page, $O_{dom} = (C, R, A, I)$ une ontologie adaptative et $e = (u, v) \in E$ une arête. La distance sémantique entre u et v est :*

$$\rho_e(u, v) = \min_{\Gamma \in Path(u,v)} \sum_{i=1}^{l(\Gamma)} SD[\Phi(i, \Gamma)]$$

avec :

$$\begin{aligned} \Phi : [\![1, n]\!] \times Path &\longrightarrow R \\ (i, \Gamma) &\longrightarrow (k_i, k_{i+1}) \end{aligned}$$

Cette valeur correspond au chemin le plus court, du point de vue de la distance sémantique entre u et v. Ainsi, plus la valeur de ρ_e entre deux sommets est faible, plus les deux concepts associés à ces sommets sont proches dans l'ontologie.

Un WPGraph enrichie le contenu de la page Web qui lui est associée car le WPGraph intègre des informations sur les relations sémantiques entre les termes de la page. L'utilisation de l'ontologie adaptative du domaine, et en particulier la distance sémantique utilisée pour construire les arêtes du graphe, met en relation les termes de la page entre eux. L'utilisation de l'ontologie des documents du domaine et du poids sémantique associé aux concepts de l'ontologie permettent de mettre en relief les termes les plus importants de la page.

Le modèle des W^{3}Graphs

Les W^{3}Graphs sont des graphes permettant de représenter le contenu enrichi d'un ensemble de pages Web et les liens sémantiques qui existent entre ces pages. Contrairement aux approches classiques utilisant des graphes pour modéliser le Web [Kleinberg et al., 1999, Broder et al., 2000] nous allons définir un graphe dont les arêtes ne représentent pas un hyperlien pointant d'une page vers une autre, mais une relation définie à partir de la sémantique du contenu des pages.

Définition 4.5 *Soient Σ un alphabet fini et un ensemble de WPGrpah W, on définit GW un W^{3}Graph comme un triplet (S, A, ρ_{GW}) où :*

1. *$S = \{(ws, url) | ws \in WS, url \in \Sigma^*\}$ est l'ensemble des sommets du graphe (url est l'adresse de la page Web associée à ws)*
2. *$A = \{\{x, y\} | x \in S, y \in S\}$ est l'ensemble des arêtes*
3. *$\rho_{GW} : A \longmapsto \mathbb{R}^+$ est la fonction de pondération des arêtes*

Un W^{3}Graph est un graphe dont les sommets sont des couples (WP, url) où WP est un WPGraph et url est l'adresse url de la page Web associée à WP. L'ensemble des arêtes représente les liens sémantiques entre les sommets. Le poids associé à chaque arête représente une distance sémantique entre les sommets. Plus la valeur du poids est élevée, plus les sommets sont éloignées du point de vue de la sémantique de leur contenu.

Le poids associé à chaque arête du graphe est calculé à partir de l'ontologie du domaine. L'idée est de reprendre les sommets dont les poids sont les plus élevés des deux WPGraph formant l'arête, puis d'utiliser l'ontologie du domaine pour calculer la distance sémantique entre ces deux sommets. On procède de la même manière avec les deux sommets les deuxièmes les plus importants puis les troisièmes plus importants et ainsi de suite, puis d'en faire la moyenne.

Définition 4.6 *Soient* $G = (V_G, E_G, T, \varphi_G, \rho_{vG}, \rho_{eG})$ *et* $H = (V_H, E_H, T, \varphi_H, \rho_{vH}, \rho_{eH})$ *deux WPGraph, on note* $D_G = \{g_i \mid \forall i, j\ 1 \leq i \leq j \leq |V_G|\ \rho_{vG}(g_i) \leq \rho_{vG}(g_j)\}$ *et* $D_H = \{h_i \mid \forall i, j\ 1 \leq i \leq j \leq |V_H|\ \rho_{vH}(h_i) \leq \rho_{vH}(h_j)\}$ *l'ensemble des concepts les plus importants de G et H. On définit la fonction de pondération*

$$\rho_{GW}(G, H) = \sum_{i=1}^{k} \frac{\rho_e(g_i, h_i)}{k}$$

avec $g_i \in D_G$, $h_i \in D_H$ *et* $k = \min(|V_G|, |V_H|)$

4.4.2 Formalisation logique

Dans cette section, nous présentons la formalisation en logique du premier ordre des WP-Graphs et W^{3}Graphs. Cette formalisation est nécessaire dans le but d'homogénéiser les différents concepts que nous allons introduire dans la suite de ce mémoire notamment le langage de requête ASK (voir section 5.1) et permettre l'utilisation combinée de tous ces éléments.

Formalisation des WPGraphs

La structure logique des WPGraphs respecte le formalisme donné à la section 4.4.1. Etant donné un WPGraph $GD = (V, E, T, \varphi, \rho_v, \rho_e)$, on définit la structure

$$SLG_{GD} =< D_{GD}, V_{GD}, E_{GD}, \varphi_{GD}, \rho_{vGD}, \rho_{eGD} >$$

Soit $D_{GD} = \Sigma^+ \cup \mathbb{R} \cup T$ on définit alors les relations $V_{GD}, E_{GD}, \varphi_{GD}, \rho_{vGD}, \rho_{eGD}$

$V_{GD} \subseteq D_{GD}$:
$$V_{GD} = \{v \mid v \in V\}$$

$E_{GD} \subseteq D_{GD}^2$:
$$E_{GD} = \{(v_1, v_2) \mid v_1, v_2 \in V \wedge \{v_1, v_2\} \in E \vee \{v_2, v_1\} \in E\}$$

$\varphi_{GD} \subseteq D_{GD}^3$:
$$\forall v, l, t \in D_{GD}\ \varphi(v) = (l, t) \Rightarrow (v, l, t) \in \varphi_{GD}$$

$\rho_{vGD} \subseteq D_{GD}^2$:
$$\forall v, r \in D_{GD}\ \rho_v(v) = r \Rightarrow (v, r) \in \rho_{vGD}$$

$\rho_{eGD} \subseteq D_{GD}^3$:
$$\forall v_1, v_2, r \in D_{GD}\ \rho_e(\{v_1, v_2\}) = r \Rightarrow (v_1, v_2, r) \in \rho_{eGD}$$

Formalisation des W^{3}Graphs

La formalisation logique des W^{3}Graphs respecte celle proposée la formalisation des WP-Graphs.

Soit un W^3Graph $GW = (S, A, \rho_{GW})$. Nous pouvons alors définir la structure

$$SLG_{GPW} =< D_{GPW}, S_{GPW}, A_{GPW}, \rho_{GPW} >$$

telle que :
$D_{GPW} = \Sigma^+ \cup \mathbb{R}$ avec les relations S_{GPW}, A_{GPW} et ρ_{GPW} :

$$S_{GPW} \subseteq D^2_{GPW} : \quad S_{GPW} = \{(v,u) \in S\}$$

$$A_{GPW} \subseteq D^4_{GPW} : \quad A_{GPW} = \{(S_1,S_2) \mid S_1,S_2 \in S \wedge \{S_1,S_2\} \in A \vee \{S_2,S_1\} \in A\}$$

$$\rho_{GPW} \subseteq D^5_{GPW} : \quad \forall s_1,s_2,r \in D_{GPW},\ \rho_{GW}(\{s_1,s_2\}) = r \Rightarrow (s_1,s_2,r) \in \rho_{GPW}$$

4.4.3 Construction des graphes du Web

Dans cette section nous présentons les algorithmes de construction des WPGraphs et W^{3}Graphs à partir des données du Web et d'ontologies adaptatives du domaine et des documents.

Le cas des WPGraphs

La construction d'un WPGraph est faite suivant une analyse syntaxique et sémantique de la page Web associée. L'algorithme de construction d'un WPGraph prend en paramètres une page Web, l'ontologie adaptative du domaine et des documents du domaine. Il se détaille de la façon suivante :

Algorithme 2 Construction d'un WPGraph

Entrées: O_{dom}, O_{doc} des ontologies adaptatives,
P une page Web
Sorties: W un WPGraph
pour i=1 à nbtermes(P) **faire**
si !vide(P[i]) **alors**
AjouterSommet(P[i],W)
AjouterArete(P[i],O_{dom},W)
finsi
fin pour
Retourner W.

L'analyse syntaxique nécessite l'utilisation de l'ontologie des documents du domaine. L'idée est d'utiliser les concepts de l'ontologie des documents représentant les parties de la page Web sensées contenir l'information la plus pertinente de la page pour construire l'ensemble des sommets du graphe et leur poids. Le contenu de ces parties est analysé en s'appuyant sur la syntaxe du code HTML. Concrètement, le code HTML de ces parties est parcouru afin d'en extraire les termes, d'éliminer les mots vides (à travers la fonction *vide* qui teste si un terme est porteur d'information), et de calculer l'importance des termes retenus selon la métrique 4.3 avec la procédure *AjouterSommet*.

L'analyse sémantique met en œuvre l'ontologie du domaine pour construire l'ensemble des arêtes et le poids qui leur est associé, à partir de l'ensemble des sommets construit lors de l'analyse syntaxique. L'idée est de considérer les sommets deux à deux et de leur appliquer la métrique 4.4 à travers la procédure *AjouterArete*.

Le cas des W^{3}Graphs

La construction d'un W^{3}Graphs s'appuie sur le contenu des WPGraphs constituant les sommets du graphe et sur l'ontologie du domaine pour calculer le poids de chaque arête du graphe.

Algorithme 3 Construction d'un W^{3}Graph

Entrées: O_{dom}, O_{doc} des ontologies adaptatives,
WP un ensemble de pages Web
Sorties: W un W^{3}Graph
pour chaque $S = (p, url) \in WP$ **faire**
wp $\longleftarrow$ ConstruireWPGraph(p,O_{dom}, O_{doc})
AjouterWPGraph(wp,url,W)
ConstruireArete(wp,O_{dom},W)
fin pour
Retourner W.

La fonction *ConstruireWPGraph* correspond à l'algorithme de construction d'un WPGraph, présenté à la section 4.4.3. La procédure *AjouterWPGraph* construit, au fur et à mesure, les sommets du W^{3}Graph. Enfin, la procédure *ConstruireArete* construit l'ensemble des arêtes du graphe et les poids qui leur sont associés.

4.5 Conclusion

Dans ce chapitre, nous avons présenté comment des ontologies adaptatives pouvaient être utilisées pour la représentation du contexte d'une requête en recherche d'information et pour l'enrichissement des données du Web.

Les ontologies adaptatives permettent de représenter des informations contextuelles : la représentation du domaine de recherche visé par une requête, et les vues des catégories d'utilisateurs sur ce domaine. Les caractéristiques des ontologies adaptatives permettent de combler les lacunes mises en évidence par notre étude, en prenant en compte l'évolution en recherche d'information.

Les ontologies adaptatives représentant des informations contextuelles peuvent être utilisées pour enrichir les données du Web. Nous avons proposé de nouvelles structures de données, construites à partir de ces ontologies, pour la représentation des données du Web. Ces structures permettent de représenter le Web et son contenu en tenant compte de la sémantique du contenu des pages Web et non suivant la seule structure du Web, comme c'est le cas dans la plupart des approches. Ces nouvelles structures vont permettre l'amélioration de la recherche documentaire sur le Web du point de vue de la pertinence.

Dans le chapitre suivant, nous allons présenter une utilisation des ontologies adaptatives, représentant un domaine de recherche et les vues des utilisateurs sur ce domaine, pour l'extraction des informations du Web. Le processus décrit exploite les structures de graphes présentées dans ce chapitre (WPGraph et W^{3}Graph). Nous présentons également un langage de requête spécifiquement conçu pour interroger ces structures ainsi qu'une technique d'enrichissement de requête basée sur les ontologies adaptatives.

Chapitre 5

Langage de requête et règles d'enrichissement de requêtes

Sommaire

Dans le processus de recherche d'information sur le Web, la requête est l'élément clé. Les possibilités offertes par les langages de requête, le choix des mots clés et la façon dont ils sont combinés, déterminent, en partie, la qualité des informations retournées suite à l'évaluation de la requête.

Le langage de requête qui va permettre à l'utilisateur d'exprimer ses besoins en information est important. La syntaxe de ce langage doit être à la fois suffisamment intuitive pour permettre à l'utilisateur de construire des requêtes simples et aussi suffisamment expressive afin que les utilisateurs puissent spécifier de manière précise leurs besoins réels en matière d'information. Les structures de données introduites au chapitre précédent (les WPGraphs et les W^3Graphs) supportent une nouvelle méthode de représentation du contenu du Web. Un langage de requête

spécifique est nécessaire afin de pouvoir exploiter au mieux les caractéristiques de ces structures et d'en extraire l'information la plus pertinente.

Le choix des mots clés et la façon dont l'utilisateur va les combiner dans la requête est le deuxième facteur qui conditionne le succès du processus de recherche d'information. Les techniques automatiques d'enrichissement des requêtes permettent en quelque sorte de faciliter ce choix à travers la proposition des mots clés additionnels les plus pertinents par rapport à ceux déjà présents dans la requête. Notre étude, détaillée au chapitre 1 section 1.3.2, a mis en avant l'utilisation de ressources terminologiques, comme les ontologies, pour représenter le vocabulaire à sélectionner. Cependant, les techniques existantes n'exploitent que partiellement les caractéristiques des éléments ontologiques pour faire ce choix. Les éléments du modèle des ontologies adaptatives ainsi que l'utilisation de relations ontologiques supplémentaires méritent d'être exploiter pour la définition de nouveaux mécanismes d'enrichissement de requêtes pour augmenter le pouvoir filtrant de ces dernières.

Dans ce chapitre, nous allons présenter le langage de requête ASK pour l'extraction de l'information pertinente des WPGraphs et W^{3}Graphs. Nous détaillons la syntaxe ainsi que la sémantique du langage. Dans la seconde partie de ce chapitre, nous décrivons une nouvelle technique d'enrichissement des requêtes basée sur les ontologies adaptatives. Cette technique exploite les connaissances de l'ontologie du domaine de recherche et celle représentant les vues des utilisateurs sur ce domaine.

5.1 Le langage de requête ASK

Le langage de requête ASK que nous introduisons dans cette section est conçu pour l'extraction des données pertinentes des WPGraphs et W^{3}Graphs, tous deux introduits au chapitre précédent. Le langage est construit sur le modèle booléen présenté au premier chapitre section 1.3.1. Nous présentons dans un premier temps la syntaxe du langage puis, dans un deuxième temps, nous introduisons sa sémantique en nous appuyant sur la structure logique des WPGraphs et W^{3}Graphs.

5.1.1 Syntaxe abstraite

La syntaxe du langage ASK est directement inspirée de celle des langages des moteurs de recherche (voir section 1.3.1). Les requêtes ASK sont des combinaisons de mots clés reliés par des connecteurs logiques. Le langage offre également d'autres constructeurs pour exprimer, par exemple, le type de documents recherchés et le domaine de recherche visé par la requête.

La grammaire Gr du langage ASK est la suivante :

$$Gr = (X_{Gr}, V_{Gr}, expr\_list, R_{Gr})$$

$$\begin{array}{rcl} X_{Gr} & = & \{\text{"("},\text{")"},\text{"\{"},\text{"\}"},\text{" }-\text{"},\text{" :"},\text{"."},\text{"!"},\text{"|"},\text{"\&"},\text{"\#"},\text{" ="}, \\ & & \text{"}img\text{"},\text{"}snd\text{"},\text{"}fil\text{"},\text{"}empty\text{"},\text{"}vid\text{"}\} \\ V_{Gr} & = & \{expr\_list, expr\_part, expr, type, X\} \\ R_{Gr} & = & \{expr\_list := expr\_list\ expr\_part \\ & & \qquad\qquad \mid expr\_part \\ & & expr\_part := expr : X \\ & & expr := X \end{array}$$

$$
\begin{array}{rl}
 & \mid X.X(=X) \\
 & \mid expr\ \text{"}\{\text{"}\ type\ \text{"}\}\text{"} \\
 & \mid \text{"}(\text{"}\ expr\ \text{"})\text{"} \\
 & \mid \text{"}!\text{"}\ expr \\
 & \mid \text{"}-\text{"}\ expr \\
 & \mid expr\ \text{"}\&\text{"}\ expr \\
 & \mid expr\ \text{"}\#\text{"}\ expr \\
 & \mid expr\ \text{"}|\text{"}\ expr \\
type := & \text{"}img\text{"} \\
 & \mid \text{"}snd\text{"} \\
 & \mid \text{"}empty\text{"} \\
 & \mid \text{"}vid\text{"} \\
 & \mid \text{"}fil\text{"} \\
X := & \ 'a'..'z' \\
 & \mid 'A'..'Z'
\end{array}
$$

5.1.2 Sémantique

Nous allons à présent définir la sémantique du langage ASK. Pour cela, nous nous appuyons sur la formalisation en logique du premier ordre des structures de données introduites au chapitre précédent. ASK est défini spécialement afin d'extraire l'information pertinente des WPGraphs et W^{3}Graphs, les expressions du langage respectent la structure de ces graphes. Dans le tableau ci-dessous, on considère un WPGraph $SLG_{GD} =< D_{GD}, V_{GD}, E_{GD}, \varphi_{GD}, \rho_{vGD}, \rho_{eGD} >$.

Expression du langage :	**Equivalent en logique du premier ordre :**
mot_1	$\exists v \in V_{GD}\ v = mot_1 \wedge \varphi_{GD}(mot_1) = (\epsilon, \perp) \wedge \rho_{vGD}(mot_1) \geq 0$
$mot_1\{t\}$	$\exists v \in V_{GD}\ \forall x \in \Sigma^+ \subseteq D_{GD}\ v = mot_1 \wedge \varphi_{GD}(mot_1) = (x, t) \wedge \rho_{vGD}(mot_1) \geq 0$
$mot_1 : dom$	$\exists v, d \in V_{GD}\ v = mot_1 \wedge d = dom\varphi_{GD}(mot_1) = (\epsilon, \perp) \wedge \rho_{vGD}(mot_1) \geq 0$
$!mot_1$	$\forall v \in V_{GD}\ v \neq mot_1$
$mot_1 \& mot_2$	$\exists v_1, v_2 \in V_{GD}\ v_1 = mot_1 \wedge v_2 = mot_2 \wedge \varphi_{GD}(mot_1) = (\epsilon, \perp) \wedge \varphi_{GD}(mot_2) = (\epsilon, \perp) \wedge \rho_{vGD}(mot_1) \geq 0 \wedge \rho_{vGD}(mot_2) \geq 0$
$mot_1 \mid mot_2$	$\exists v_1, v_2 \in V_{GD}\ (v_1 = mot_1 \wedge \varphi_{GD}(mot_1) = (\epsilon, \perp) \wedge \rho_{vGD}(mot_1) \geq 0) \vee (v_2 = mot_2 \wedge \varphi_{GD}(mot_2) = (\epsilon, \perp) \wedge \rho_{vGD}(mot_2) \geq 0)$

$mot_1 \# mot_2$	$\exists v_1, v_2 \in V_{GD}\ ((v_1 = mot_1 \wedge v_2 \neq mot_2 \wedge \varphi_{GD}(mot_1) = (\epsilon, \perp) \wedge \rho_{vGD}(mot_1) \geq 0) \vee (v_1 \neq mot_1 \wedge v_2 = mot_2 \wedge \varphi_{GD}(mot_2) = (\epsilon, \perp) \wedge \rho_{vGD}(mot_2) \geq 0))$
$(mot_1 \& mot_2)\{t\} : dom$	$\exists v_1, v_2, d \in V_{GD}\ \forall x, y \in \Sigma^+ \subseteq D_{GD}\ v_1 = mot_1 \wedge v_2 = mot_2 \wedge \varphi_{GD}(mot_1) = (x, t) \wedge \varphi_{GD}(mot_2) = (y, t) \wedge d = dom$
$(mot_1 \mid mot_2)\{t\} : dom$	$\exists v_1, v_2 \in V_{GD}\ \forall x, y \in \Sigma^+ \subseteq D_{GD}\ (v_1 = mot_1 \wedge \varphi_{GD}(mot_1) = (x, t)) \vee (v_2 = mot_2 \wedge \varphi(mot_2) = (y, t)) \wedge d = dom$
$(mot_1 \# mot_2)\{t\} : dom$	$\exists v_1, v_2 \in V_{GD}\ \forall x, y \in \Sigma^+ \subseteq D_{GD}\ ((v_1 = mot_1 \wedge v_2 \neq mot_2 \wedge \varphi_{GD}(mot_1) = (x, t)) \vee (v_1 \neq mot_1 \wedge v_2 = mot_2 \wedge \varphi_{GD}(mot_2) = (y, t))) \wedge d = dom$
$(mot_1\ op\ mot_2)\{t\} : dom \quad \Leftrightarrow$	$\begin{cases} (mot_1\{t_1\}\ op\ mot_2)\{t\} : dom \\ (mot_1\ op\ mot_2\{t_1\})\{t\} : dom \\ (mot_1\{t_1\}\ op\ mot_2\{t_2\})\{t\} : dom \end{cases}$
$mot.att(= val)$	$\exists v_1, v_2, v_3 \in V_{GD}\ v_1 = mot \wedge v_2 \neq att \wedge v_3 = val \wedge \varphi_{GD}(mot) = (\epsilon, \perp) \wedge \varphi_{GD}(att) = (\epsilon, \perp) \wedge \varphi_{GD}(val) = (\epsilon, \perp) \wedge \rho_e(mot, att) = \rho_e(mot, val) = \rho_e(val, att)$
$-mot_1$	Voir section 5.2.4

TABLE 5.2: Sémantique du langage ASK

De manière plus pragmatique, les constructeurs ont la signification suivante :

- L'opérateur binaire "&" représente la conjonction. Il force la présence des deux termes dans les pages recherchées.
- L'opérateur binaire "|" représente la disjonction. Il force la présence d'au moins un des deux termes dans les pages recherchées.
- L'opérateur binaire "#" représente la disjonction exclusive. Les pages recherchées contiennent un des deux termes seulement.
- L'opérateur unaire " !" représente la négation. Les pages recherchées ne contiennent pas le terme.
- L'opérateur unaire "-" représente l'opposé du terme auquel il est associé.
- Les opérateurs .(=) permettent de faire référence à des attributs de concepts au niveau de la requête. L'étude menée par Jansen et al. [Jansen et al., 2000] montre que les adjectifs sont la deuxième catégorie de termes la plus employée dans les requêtes pour le Web. De plus, ils sont la plupart du temps utilisés pour exprimer les attributs. Cet opérateur permet donc de préciser davantage les informations recherchées (voir exemple ci-après).
- L'opérateur "{}" permet de spécifier le type de document recherché. L'ensemble des types possibles correspond à celui des étiquettes des WPGraphs (i.e. snd, img, vid, fil).
- L'opérateur " :" permet de spécifier le domaine de recherche visé.

Remarque: Dans les explications ci-dessus, une analogie est faite entre la notion de page Web et un WPGraph.

Le langage ASK offre d'autres spécificités parmi lesquelles on retrouve :

- La non distinction entre les majuscules et les minuscules.
- L'élimination des mots courants comme "de", "le", "pour" ainsi que les signes de ponctuation.
- L'évaluation des expressions de gauche à droite.
- L'utilisation des parenthèses est possible pour instaurer des priorités. L'expression entre parenthèses sera évaluée en priorité. De plus, si l'opérateur { } est placé après une parenthèse fermante, alors il s'applique à tous les termes de l'expression entre parenthèses.
- L'opérateur "" (proposé par Google) n'est pas autorisé (car les WPGraphs ne traitent que des concepts isolés et non pas des chaînes de caractère comportant plusieurs mots).

Exemple: Le tableau ci-dessous contient diverses requêtes ASK ainsi que leur sémantique et les pages Web visées.

Objectif	**Syntaxe**	**Sémantique**	**Exemple**	**Résultat**
Un terme unique dans un domaine particulier	A :d	Page contenant au moins une occurrence de A dans le domaine d	luxembourg : pays	les pages contenant le terme luxembourg en tant que pays
Deux termes dans la même page	A&B	pages contenant à la fois les mots A et B	Tom&Jerry	Pages contenant à la fois "Tom" et "Jerry"
Au moins un terme dans la page	A|B	pages contenant soit A soit B soit les deux	vis|clou	Pages contenant un des termes mentionnés ou les deux
La page sans le terme	!B	pages ne contenant pas B	!Seine	ne contenant pas "Seine"
Seulement un des deux termes dans la page	A#B	pages contenant soit A soit B	fruit#glace	Pages contenant "fruit" ou "glace" mais pas les deux termes simultanément
Le contraire du terme	-A	page contenant l'antonyme de A	-salé :cuisine	Pages faisant référence au contraire du salé en cuisine
Des pages sur des choses plus précises	A.B(=C)	Page contenant l'objet A dont son attribut B à la valeur C	voiture.couleur (=rouge)	Pages contenant des informations sur les voitures rouges
Des types de documents précis	A{t}	Page contenant des objet de type t sur A	chien{img}	Pages contenant des images de chien

TABLE 5.3: Exemples de requêtes ASK

Si le langage ASK présente des similarités avec les langages des moteurs de recherche usuels comme les opérateurs de conjonction, de disjonction et de négation, celui-ci offre par ailleurs des opérateurs différents. L'opérateur de disjonction exclusive, celui pour exprimer la notion d'attri-

but de concept, l'opérateur pour désigner le sens opposé d'un terme et l'opérateur permettant de spécifier au niveau de la requête le domaine de recherche visé n'existent pas dans les langages des moteurs de recherche existants.

5.2 Relations ontologiques et enrichissement des requêtes ASK

Le langage ASK a également été conçu pour favoriser l'enrichissement des requêtes. Dans cette section, nous allons détailler le mécanisme d'enrichissement des requêtes que nous proposons afin d'améliorer la pertinence des résultats lors d'une recherche documentaire sur le Web. Le mécanisme proposé [Guelfi et al., 2007a] s'appuie sur l'utilisation des ontologies adaptatives et plus particulièrement sur les propriétés de certaines relations ontologiques et des éléments caractéristiques des ontologies adaptatives.

La relation ontologique de base présente dans la plupart des modèles d'ontologies est la subsumption. De plus, le modèle OWL permet d'exprimer la relation d'équivalence entre les concepts issues d'ontologies différentes. Ces deux relations sont également celles utilisées dans les approches existantes pour l'enrichissement des requêtes basées sur des ontologies (voir section 1.3.2). En plus de ces deux relations proposées par le modèle de OWL, nous proposons dans un premier temps d'utiliser les relations d'instanciation, de méronymie et d'antonymie. Nous discuterons par la suite de l'apport des ontologies adaptatives.

5.2.1 Le cas de l'équivalence

Le langage OWL permet de construire des ontologies contenant des concepts définis comme équivalents (voir le tableau 2 de la section 1.1.2). La relation d'**équivalence** peut s'apparenter à la relation de **synonymie** en linguistique. La synonymie est un rapport de proximité sémantique entre des mots d'une même langue. La proximité sémantique indique que les mots ont des significations très semblables. Deux termes sont synonymes s'ils sont interchangeables dans la phrase où ils sont employés sans que celle-ci ne change de sens.

Du point de vue de l'enrichissement des requêtes la considération de la relation d'équivalence entre les concepts de l'ontologie pour sélectionner des termes supplémentaires pour la requête permet principalement de désambiguïser le contexte visé par celle-ci.

Exemple: Pour illustrer l'apport de la relation d'équivalence pour l'enrichissement des requêtes, supposons que la requête initiale *pot* soit posée. Le système doit pouvoir lever l'ambiguïté entre les différents contextes pouvant être visés par la requête. La requête peut viser les pots en tant que récipients ou en tant qu'apéritifs par exemple. Si l'ontologie utilisée pour l'enrichissement des requêtes définit pot et récipient comme équivalents, la requête enrichie pourra être *pot&recipient* prévenant ainsi toutes ambiguïtés.

5.2.2 Le cas de la subsumption et des instances de concepts

La relation de **subsumption** dans les modèles d'ontologies trouve son pendant dans la relation linguistique d'**hyponymie**. L'hyponymie est la relation sémantique d'un lexème à un autre selon laquelle l'extension du premier est incluse dans l'extension du second. Le premier terme est dit hyponyme de l'autre. C'est le contraire de l'**hyperonymie**. Par exemple, «haut-de-forme» est un hyponyme de «chapeau» et «chapeau» est un hyponyme de «coiffure». La relation de

subsumption permet d'établir une hiérarchie entre les concepts de l'ontologie, elle est souvent représentée par le label "is-a" (i.e. est un type de).

Les **instances de concepts** (ou individus en OWL) sont des éléments ontologiques représentant les objets du monde réel. La relation qui relie une instance à un concept peut être vue comme une relation de subsumption. Considérons, par exemple, le concept *personne*, l'instance *Cédric Pruski* "est une" *personne*.

Du point de vue de l'enrichissement des requêtes, les relations d'hypéronymie et d'instanciation permettent, tout comme l'équivalence, de désambiguïser le domaine de recherche. L'ajout d'un terme dénotant un concept plus général que ceux de la requête initiale permet de caractériser le domaine de recherche (i.e. augmenter la précision de la recherche documentaire) sans pour autant contraindre de manière excessive la requête. Une requête sur-contrainte aurait pour effet d'accroître le rappel de la recherche (voir exemples ci-dessous).

Exemple 1 : Considérons, pour montrer l'intérêt d'utiliser la relation de subsumption, la requête initiale *souris* et comme domaine visé le monde animalier. Comme *animal* est l'hyperonyme de *souris*, son ajout à la requête initiale permet de préciser le domaine. Toutes les pages faisant référence aux souris en tant que périphériques d'ordinateur seront filtrées.

Au contraire, si la requête initiale était *coiffure*, l'ajout d'un hyponyme de *coiffure*, à savoir *chapeau*, dans la requête n'aurait pas le même effet. Le pouvoir filtrant de la requête enrichie est trop important. En saisissant la requête *coiffure*, l'utilisateur s'attend à se voir proposer un ensemble de pages Web traitant de coiffure en général et non pas seulement de coiffure en tant que chapeau. En conséquence, beaucoup de pages qui auraient dues être retournées sont éliminées lors de l'évaluation de la requête étendue.

Exemple 2 : Supposons, pour illustrer l'utilité de la notion d'instanciation, qu'un utilisateur souhaite obtenir des informations sur le poids d'une Giulia (voiture de la marque Alpha Romeo). Si une requête ne contient que *poids Giulia*, beaucoup de pages retournées concerneront des personnes dont le prénom est Giulia. En revanche, si une ontologie du domaine automobile contient Giulia en tant qu'instance de voiture, et qu'il a été explicitement déclaré que l'interprétation de cette requête devait être faite dans le domaine de l'automobile, l'ajout du concept voiture servira de filtre et permettra de ne retenir que les pages concernant les voitures Giulia.

5.2.3 Le cas de la relation "partie de"

La relation de méronymie est une relation hiérarchique partitive. En d'autres termes, c'est la relation qui relie une partie à un tout, comme par exemple une voiture (tout) et une roue (partie). La méronymie correspond à une relation relativement générale. En effet, les liens entre composant/objet, membre/collection ou encore matériau/objet sont tous qualifiés de liens de méronymie alors que la relation de composition liant le composant au composé n'est pas exactement identique. Il convient donc de considérer plusieurs types de relations de méronymie comme le fait, par exemple, UML [1], en distinguant l'agrégation de la composition. Ces deux types de relation se différencient par le fait qu'une instance d'une partie d'un objet composite ne peut appartenir à un autre objet (agrégat ou composite). La sémantique que nous adoptons pour cette primitive correspond à celle proposée dans UML [Barbier et al., 2003] pour l'agrégation et la composition. La composition et l'agrégation sont des relations s'appliquant aux concepts. Elles sont toutes

1. http ://www.uml.org/

deux transitives et asymétriques. Les contraintes de cardinalité sont en revanche différentes. Les deux relations reposent sur les axiomes suivants :

Agrégation :

1. $\forall c_1, c_2, c_3\ \forall x, y, z\ Agg(c_1(x), c_2(y)) \wedge Agg(c_2(y), c_3(z)) \longrightarrow Agg(c_1(x), c_3(z))$
2. $\forall c_1, c_2\ \forall x, y\ Agg(c_1(x), c_2(y)) \longrightarrow \neg Agg(c_2(y), c_1(x))$
3. $\forall c_1, \exists x, y\ 1 \leq |\{c_2(y), Agg(c_1(x), c_2(y))\}|$
4. $\forall c_2, \exists x, y\ 1 \leq |\{c_1(x), Agg(c_1(x), c_2(y))\}|$

Remarque: c_i sont des concepts, x, y et z des instances. Pour des raisons de lisibilité, nous avons utilisé $c_i(x)$ dans les axiomes. Il faudrait écrire $Agg(c_1, x, c_2, y)$ et l'axiome $Agg(c_1, x, c_2, y) \longrightarrow c_1(x), c_2(y)$. Ceci vaut également pour composedOf (voir ci-dessous).

Suivant cette définition axiomatique, la relation d'agrégation peut exprimer qu'à un même moment, une instance d'élément agrégé peut être liée à plusieurs instances d'autres concepts (i.e. l'élément agrégé peut être partagé). De plus, une instance d'élément agrégé peut exister sans agrégat (et inversement) : les cycles de vies de l'agrégat et de ses éléments agrégés peuvent être indépendants. Par exemple, une *commande* (l'agrégat) est constituée de plusieurs *produits* mais la destruction de l'objet *commande* n'implique pas la destruction des objets *produits*.

Composition :

1. $\forall c_1, c_2, c_3\ \forall x, y, z\ composedOf(c_1(x), c_2(y)) \wedge composedOf(c_2(y), c_3(z))$
   $\longrightarrow composedOf(c_1(x), c_3(z))$
2. $\forall c_1, c_2\ \forall x, y\ composedOf(c_1(x), c_2(y)) \longrightarrow \neg composedOf(c_2(y), c_1(x))$
3. $\forall c_1, \exists x, y\ 1 \leq |\{c_2(y), composedOf(c_1(x), c_2(y))\}|$
4. $\forall c_2, \exists x, y\ |\{c_1(x), composedOf(c_1(x), c_2(y))\}| = 1$

Les axiomes de la composition permettent de dire que la composition est une agrégation forte dans la mesure où :

- Les cycles de vies des éléments (les "composants") et de l'agrégat sont liés : si l'agrégat est détruit, ses composants le sont aussi.
- A un même moment, une instance de composant ne peut être liée qu'à un seul agrégat.

Considérons, pour illustrer ces propriétés, un objet *genou* composé d'une *rotule* et d'autres composants. La destruction de la rotule entraîne la destruction du genou et réciproquement. De plus, une rotule ne peut faire partie que d'un seul genou.

Une manière d'exprimer la méronymie dans le langage OWL consiste soit à utiliser des collections d'objets (via les primitives containers en RDF) ou la réunion de plusieurs concepts (avec la primitive owl :unionOf). Ces solutions permettent de représenter la notion d'agrégat au niveau des classes (avec owl :unionOf) et au niveau des instances (avec les containers). Cependant, ces solutions sont très peu intuitives et accroissent la complexité du raisonnement. Par ailleurs, il n'est pas possible de définir la composition de cette manière car une instance impliquée dans les relations mises en œuvre par les containers ou par la primitive owl :unionOf n'est pas obligatoirement reliée uniquement au concept agrégat, d'où la nécessité d'introduire une primitive spécifique dans notre approche. La formalisation de ces deux types de méronymies que nous

proposons peut servir de base à leur intégration à OWL.

L'idée de distinguer les notions d'agrégation et de composition est également nécessaire d'un point de vue expansion de requête comme le montre l'exemple suivant. Dans cet objectif, seule la composition peut être considérée. L'ajout de l'élément composé à la requête contenant un terme dénotant un composant permet de caractériser les informations recherchées en vertu des propriétés du cycle de vie des objets.

Exemple 1 : Considérons la requête initiale *roue* et l'ontologie des moyens de locomotion dans laquelle la roue est un objet faisant partie d'une *automobile* et d'une *bicyclette*. Le lien de méronymie reliant ces objets peut être qualifié d'agrégation car d'une part la destruction de la *roue* n'entraîne pas celle de la *voiture* et de la *bicyclette* et d'autre part, la *roue* peut appartenir à la fois à la *voiture* et à la *bicyclette*. Cette dernière propriété rend l'enrichissement de la requête initiale indécidable car il n'y a aucun moyen de décider quel terme (automobile ou bicyclette) rajouter.

Exemple 2 : Supposons que la requête initiale soit *rotule* et que l'ontologie utilisée pour l'expansion contienne le concept *rotule* relié au concept *genou* par une relation de composition. La propriété de la composition selon laquelle un composant ne peut faire partie que d'un unique composé permet d'ajouter le terme genou à la requête. La requête fait explicitement référence au domaine médical et permet ainsi de filtrer l'ensemble des pages Web n'appartenant pas à ce domaine et notamment toutes celles parlant de rotule dans le domaine de la mécanique.

5.2.4 Le cas de la relation contraire

L'autre relation que nous proposons de considérer pour l'enrichissement des requêtes est l'opposition. S'apparentant à l'antonymie, cette relation permet de spécifier l'antagonisme entre concepts, relations, attributs et instances. C'est une relation complexe qui revêt différentes formes. La première est dite " opposition complémentaire " (pair/impair, présence/absence, etc.). Selon cette forme d'opposition, l'affirmation de l'un des termes entraîne nécessairement la négation de l'autre. La seconde, opposition entre des termes dits " mesurables " (petit/grand, chaud/froid, etc.), est fortement dépendante du contexte et de la valeur de référence des attributs qualifiés. Ce type d'opposition s'applique principalement sur des propriétés. Une troisième forme d'opposition affecte les valeurs spatio-temporelles et culturelles que l'on attribue aux termes (soleil/lune, départ/arrivée). Ce dernier type d'antonymie est important pour caractériser l'opposition entre concepts comme, par exemple, le bruit et le silence mais aussi entre relations comme départ et arrivée. Enfin, dans certains contextes, l'opposition concerne des instances, par exemple le fait d'opposer des personnages particuliers comme Laurel et Hardy.

La notion d'opposé est très utilisée en recherche d'information dans la vie courante et qui plus est, peu d'applications informatiques la mettent en œuvre. Ceci est particulièrement vrai pour les principaux moteurs de recherche. Ces derniers permettent d'exprimer directement la négation mais pas l'opposition entre les termes d'une requête. La principale différence entre ces deux notions peut s'exprimer d'un point de vue ensembliste (voir figure 5.1) où l'opposé représente un sous-ensemble du complémentaire. Ainsi, si un utilisateur est intéressé pour avoir des informations sur la cuisine sucrée, par exemple, et que salée est l'antonyme de sucrée, le système devra être capable d'interpréter que l'utilisateur ne cherche pas de documents sur la cuisine salée. Afin de caractériser cette relation, nous nous appuyons sur les axiomes proposés par le linguiste Edmundson [Edmundson, 1967]. Ce dernier définit l'antonymie comme une relation irréflexive,

symétrique, antitransitive, identité droite et non-vide. Cette dernière propriété forcerait tous les concepts d'une ontologie à avoir un antonyme, nous avons décidé de ne pas prendre en compte cette propriété dans notre définition de l'antonymie. Les axiomes suivants définissent la relation d'opposition. Dans ces axiomes, les C_i et les I_i sont des prédicats unaires et les R_i sont des relations. A travers la définition de l'opposition, nous montrons comment exprimer cette relation en OWL. Ainsi, contraryOf est un prédicat binaire, sameAs, equivalentClass et equivalentProperty sont les primitives de OWL.

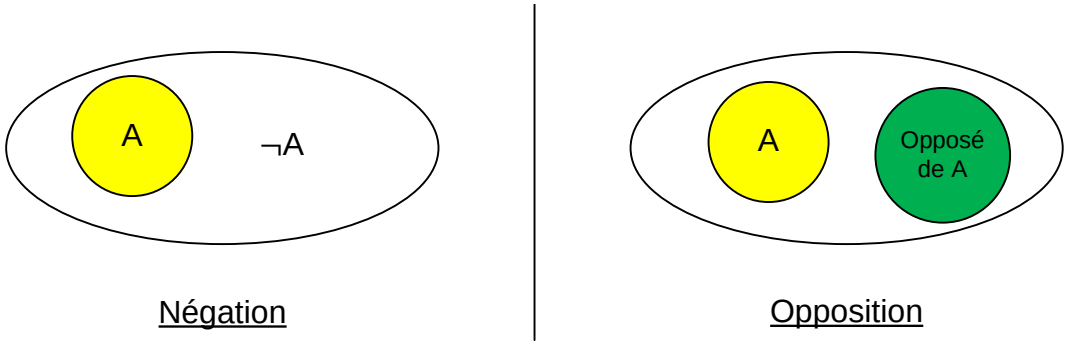


FIGURE 5.1 – Différence entre négation et opposition en RI

Opposition sur les concepts :

1. $\forall C, \neg contraryOf(C, C)$
2. $\forall C_1, C_2, contraryOf(C_1, C_2) \longrightarrow contraryOf(C_2, C_1)$
3. $\forall C_1, C_2, C_3, contraryOf(C_1, C_2) \wedge contraryOf(C_2, C_3) \longrightarrow equivalentClass(C_1, C_3)$
4. $\forall C_1, C_2, C_3, contraryOf(C_1, C_2) \wedge equivalentClass(C_2, C_3) \longrightarrow contraryOf(C_1, C_3)$

Opposition sur les instances :

1. $\forall I, \neg contraryOf(I, I)$
2. $\forall I_1, I_2, contraryOf(I_1, I_2) \longrightarrow contraryOf(I_2, I_1)$
3. $\forall I_1, I_2, I_3, contraryOf(I_1, I_2) \wedge contraryOf(I_2, I_3) \longrightarrow sameAs(I_1, I_3)$
4. $\forall I_1, I_2, I_3, contraryOf(I_1, I_2) \wedge sameAs(I_2, I_3) \longrightarrow contraryOf(I_1, I_3)$

Opposition sur les relations :

1. $\forall R, \neg contraryOf(R, R)$
2. $\forall R_1, R_2, contraryOf(R_1, R_2) \longrightarrow contraryOf(R_2, R_1)$
3. $\forall R_1, R_2, R_3, contraryOf(R_1, R_2) \wedge contraryOf(R_2, R_3) \longrightarrow equivalentProperty(C_1, C_3)$
4. $\forall R_1, R_2, R_3, contraryOf(R_1, R_2) \wedge equivalentProperty(R_2, R_3) \longrightarrow contraryOf(R_1, R_3)$

Exemple: Considérons le domaine culinaire dans lequel l'opposé de la notion de *salé* est le *sucré*. Il est clair que d'un point de vue ensembliste, la négation de *salé* (ou son complémentaire) représente tous les autres types de cuisine exceptée la cuisine salé. Par contre, l'opposé du salé (tel que nous avons défini la notion d'opposition) représente simplement la cuisine sucrée. D'un point de vue enrichissement de requête, si la requête initiale ASK *sucré :cuisine* est saisie, le système pourra enrichir la requête de telle façon que lors de son évaluation il ne retourne aucune page concernant la cuisine salée.

5.2.5 Le cas des attributs de concepts

Dans notre approche, nous souhaitons disposer d'une ontologie qui permettent d'exprimer une requête correspondant à une expression composée d'un adjectif associé à un nom (i.e. le bras rouge). Cette volonté est motivée par le fait que les adjectifs sont la deuxième catégorie de mots utilisée dans les requêtes pour le Web [Jansen et al., 2000]. OWL distingue la représentation des propriétés qui relient des individus à des valeurs de données (Datatype property) des propriétés qui relient des individus à d'autres individus (Object Property). Si une propriété n'a pas de caractère d'abstraction (i.e. nous ne souhaitons pas considérer la notion de couleur ou bien ne souhaitons pas avoir plusieurs instances de rouge), elle ne correspondra pas dans l'ontologie à une relation entre individus. Il s'agit d'un attribut, représenté en OWL par une propriété de type Datatype Property. De façon générale, les valeurs d'attributs sont très discriminantes sur un domaine donné. Elles sont donc très utiles dans notre approche pour réduire l'ensemble des résultats d'une recherche (i.e. il y a moins de pages parlant de «bras rouge» que de pages parlant de «bras»).

5.3 Ontologies adaptatives et enrichissement de requêtes ASK

Nous venons de discuter de l'apport d'un certain nombre de relations ontologiques pour l'enrichissement des requêtes pour le Web. Les ontologies adaptatives introduites dans ce mémoire présentent un ensemble de caractéristiques modélisant l'évolution des connaissances, intéressantes d'un point de vue enrichissement des requêtes. Dans cette section, nous détaillons l'apport intrinsèque des ontologies adaptatives pour l'enrichissement des requêtes ASK ainsi que l'apport spécifique des caractéristiques de l'ontologie adaptative du domaine et de celle représentant la vue de catégories d'utilisateurs sur un domaine.

5.3.1 Apport des éléments caractéristiques des ontologies adaptatives pour l'enrichissement de requêtes

Nous proposons d'affiner le choix des termes à ajouter à une requête à l'aide des éléments caractéristiques d'une ontologie adaptative. L'exploitation des relations ontologiques décrites en section 5.2 conduit très souvent à l'ajout d'un nombre important de concepts (certaines ontologies comportent plusieurs milliers d'éléments). Ce nombre doit être réduit pour ne pas sur-contraindre la requête et dégrader la recherche d'information.

La distance sémantique telle qu'elle est définie aux chapitres 2 et 3, est obtenue à partir de données statistiques calculées sur un corpus de documents du domaine. La valeur de cette distance représente la proximité dans le texte entre deux concepts d'une relation. Deux concepts proches du point de vue de la distance sémantique voient le label qui leur est associé proche dans le corpus de documents sur lequel la distance a été calculée. Par conséquent, les concepts proches doivent être privilégiés pour augmenter la précision de la requête. Ceci correspond à un premier argument pour orienter le choix des termes pour l'enrichissement de la requête initiale.

La distance sémantique étant une valeur bornée entre 1 et 10, il est fort possible que certains concepts soient à égale distance les uns des autres. Il est donc nécessaire de trouver un autre critère de raffinement. Le poids sémantique associé à un concept peut être utilisé à cet effet. La valeur du poids sémantique est obtenue à partir d'un calcul statistique sur un corpus de documents du domaine. Les termes les plus mentionnés sont associés aux concepts de l'ontologie dont le poids sémantique est le plus élevé. En conséquence, si plusieurs concepts sont à la même

distance sémantique, l'enrichissement devra être réalisé avec ceux dont le poids sémantique est le plus élevé.

La persistance des concepts à travers le temps, c'est-à-dire le temps écoulé entre l'émergence et la date actuelle, est le troisième critère de choix des termes utilisables. La persistance d'un concept est représentative de l'importance d'un concept dans le domaine. Un concept a persisté à cause de sa faculté pour résister aux évolutions. En conséquence, les concepts les plus persistants peuvent être considérés comme les plus représentatifs du domaine. Cet argument nous permet de définir un dernier raffinement dans le cas où le filtrage des termes suivant le poids sémantique n'est pas suffisant. Nous éliminons les concepts les plus récents du domaine.

5.3.2 Apport spécifique des ontologies du domaine et des vues d'utilisateurs sur un domaine

Les éléments d'une ontologie adaptative sont donc intéressants pour l'enrichissement des requêtes. Toutefois, la fonction de l'ontologie (i.e. l'objectif pour lequel l'ontologie a été construite) doit également être considérée. Dans notre approche, deux types d'ontologies adaptatives sont utilisés : une première ontologie représente le domaine de recherche visé par une requête ; une deuxième ontologie représente les vues des utilisateurs sur ce domaine.

L'ontologie du domaine est celle qui doit être utilisée en priorité pour l'enrichissement. L'objectif principal de notre approche étant l'amélioration de la pertinence des résultats obtenus en réponse à une requête posée, l'enrichissement des requêtes doit permettre d'atteindre cet objectif. Un des critères principaux pour définir la pertinence est l'appartenance d'un document retourné au domaine visé par la requête. La connaissance de ce domaine étant représentée par l'ontologie adaptative du domaine, il est naturel d'utiliser cette dernière en priorité pour l'enrichissement des requêtes ASK.

L'ontologie modélisant les vues d'utilisateurs sur un domaine intervient en complément du premier type d'enrichissement. L'ontologie adaptative représentant la vue d'utilisateurs sur un domaine permet de restreindre le domaine considéré. En effet, cette ontologie est en général beaucoup plus petite (en nombre d'éléments) que celle du domaine. De plus, nous privilégions le choix des concepts dont le poids sémantique et le plus élevé pour enrichir la requête.

5.4 Règles d'enrichissement des requêtes ASK

L'étude menée par Joho [Joho et al., 2002] portant sur la pertinence des résultats d'une recherche en fonction de la façon dont la requête a été étendue nous permet de privilégier certaines formes d'expansion au dépend d'autres. L'étude montre que les relations de synonymie entre termes associées à un même concept sont privilégiées par les utilisateurs, viennent ensuite, dans l'ordre, les relations de subsumption entre concepts, les relations partie-de et d'opposition. Nous avons ainsi établi des priorités entre les relations ontologiques intervenant dans le processus d'expansion des requêtes. L'ordre selon lequel celles-ci sont considérées est le suivant : (1) relations d'équivalence, (2) subsumption (ajout des concepts plus généraux) et instanciation, (3) composition (composition stricte et agrégation), (4) opposition. Etant donnée une requête, les ajouts ne portent que sur les concepts liés à ceux qui sont déjà présents par une seule de ces relations, celle présente dans l'ontologie ayant la plus forte priorité. Enfin, le nom du domaine de recherche est également introduit dans la requête. Ce dernier a un fort caractère discriminant car le fait de forcer le système à trouver des pages contenant, en plus des termes de la requête, le

nom du domaine permet de bien filtrer les résultats. Concernant la relation d'instanciation (non traitée par l'étude de Joho), nous avons considéré qu'elle avait la même priorité qu'une relation de subsumption. En effet, dans notre exemple précédent, une Alpha Romeo Guilia peut être considérée comme une instance du concept voiture mais aussi comme une sorte de voiture (classe liée à voiture par une relation de subsumption). Ainsi, dans notre implémentation, le choix entre subsumption et instanciation est fait aléatoirement. Ensuite, les connecteurs logiques de ASK et en particulier la conjonction, sont exploités de façon à ce que les termes ajoutés à la requête aient pour effet de contraindre davantage l'espace des résultats, ce qui donnera à l'utilisateur des informations plus précises par rapport à la requête posée initialement. Un gain de temps sera obtenu, l'utilisateur n'ayant plus à filtrer manuellement les résultats, comme c'est le cas avec les moteurs de recherche usuels.

5.4.1 Enoncé des règles

Les règles de bases que nous proposons pour l'enrichissement des requêtes ASK sont classées dans le tableau 5.4. Ces règles ne s'appuient que sur la forme de la requête initiale et les relations ontologiques présentées ci-dessus. Les éléments caractéristiques d'une ontologie adaptative interviennent plus tard pour réduire l'ensemble des mots clés répondant à une des règles.

Requête initiale	**Requête enrichie**
$-\omega : O$	1) $\omega_1 \& (\omega_2 \mid \ldots \mid \omega_n)$ $\forall i \in [\![1, n]\!]\ contraryOf(\omega, \omega_i)$
	2) $!\omega \& O$ $\nexists \alpha\ contraryOf(\alpha, \omega)$
$\omega : O$	3) $\omega \& (\omega_1 \mid \ldots \mid \omega_n \mid O)$ $\forall i \in [\![1, n]\!]\ equivalentClass(\omega\ \omega_i) \vee sameAs(\omega\ \omega_i)$
	4) $\omega \& \omega_1 \& O$ $subclassOf(\omega\ \omega_1) \vee InstanceOf(\omega\ \omega_1)$
	5) $\omega \& \omega_1 \& O$ $composedOf(\omega\ \omega_1)$
	6) $\omega \& (!\omega_1) \& O$ $contraryOf(\omega\ \omega_1)$
$\omega_1 \& \omega_2 : O$	7) $(\omega_1 \& \omega_2) \& (S_1 \mid \ldots \mid S_n \mid S_{n+1} \mid \ldots \mid S_m \mid O)$, $\forall i \in [\![1, n]\!]\ equivalentClass(\omega_1\ S_i) \vee sameAs(\omega_1\ S_i)$ $\forall j \in [\![n+1, m]\!]\ equivalentClass(\omega_2\ S_j) \vee sameAs(\omega_2\ S_j)$
	8) $(((\omega_1 \& h_1) \& (\omega_2 \& h_2)) \mid ((\omega_1 \& h_1) \& \omega_2) \mid (\omega_1 \& (\omega_2 \& h_2))) \& O$ $(subclassOf(\omega_1\ h_1) \wedge subclassOf(\omega_2\ h_2)) \vee (InstanceOf(\omega_1\ h_1) \wedge$ $InstanceOf(\omega_2\ h_2))$
	9) $(((\omega_1 \& h_1) \& (\omega_2 \& h_2)) \mid ((\omega_1 \& h_1) \& \omega_2) \mid (\omega_1 \& (\omega_2 \& h_2))) \& O$ $composedOf(\omega_1\ h_1) \wedge composedOf(\omega_2\ h_2)$
	10) pas d'enrichissement si $contraryOf(\omega_1\ \omega_2)$ $\omega_1 \& \omega_2 \& (!a_1 \mid !a_2)$ si $contraryOf(\omega_1\ a_1) \wedge contraryOf(\omega_2\ a_2)$
$\omega_1 \mid \omega_2 : O$	11) $(\omega_1 \mid \omega_2) \& (S_1 \mid \ldots \mid S_n \mid S_{n+1} \mid \ldots \mid S_m \mid O)$, $\forall i \in [\![1, n]\!]\ equivalentClass(\omega_1\ S_i) \vee sameAs(\omega_1\ S_i)$ $\forall j \in [\![n+1, m]\!]\ equivalentClass(\omega_2\ S_j) \vee sameAs(\omega_2\ S_j)$
	12) $((\omega_1 \& h_1) \mid (\omega_2 \& h_2)) \& O$ $(subclassOf(\omega_1\ h_1) \wedge subclassOf(\omega_2\ h_2)) \vee (InstanceOf(\omega_1\ h_1) \wedge$ $InstanceOf(\omega_2\ h_2))$

	13) $((\omega_1 \& h_1)\|(\omega_2 \& h_2)) \& O$ $composedOf(\omega_1\ h_1) \wedge composedOf(\omega_2\ h_2)$
	14) $(\omega_1\|\omega_2)\&O$ $contraryOf(\omega_1\ \omega_2)$ $(\omega_1\|\omega_2)\&(!a_1\|!a_2)$ $contraryOf(\omega_1\ a_1) \wedge contraryOf(\omega_2\ a_2)$
$\omega_1 \# \omega_2 : O$	15) $(\omega_1 \& (S_1\| \ldots \|S_n\|O)) \# (\omega_2 \& (S_{n+1}\| \ldots \|S_m\|O))$ $\forall i \in [\![1, n]\!]\ equivalentClass(\omega_1\ S_i) \vee sameAs(\omega_1\ S_i)$ $\forall j \in [\![n+1, m]\!]\ equivalentClass(\omega_2\ S_j) \vee sameAs(\omega_2\ S_j)$
	16) $(\omega_1 \& h_1 \& O) \# (\omega_2 \& h_2 \& O)$ $(subclassOf(\omega_1\ h_1) \wedge subclassOf(\omega_2\ h_2)) \vee$ $(InstanceOf(\omega_1\ h_1) \wedge InstanceOf(\omega_2\ h_2)) \vee$ $(composedOf(\omega_1\ h_1) \wedge composedOf(\omega_2\ h_2))$
	17) $(\omega_1 \# \omega_2) \& O$ $contraryOf(\omega_1\ \omega_2)$

TABLE 5.4: Règles d'enrichissement des requêtes ASK

Les règles ont pour objectif d'améliorer la pertinence des résultats d'une recherche d'information sur le Web. Nous proposons de valider cet aspect expérimentalement au chapitre 7.

5.4.2 Algorithme d'application des règles d'enrichissement

Les règles énoncées dans la table 5.4 s'appliquent suivant un algorithme prenant en compte la forme de la requête initiale, les mots clés qui la composent et les éléments ontologiques (les relations entre les concepts et les caractéristiques des ontologies adaptatives). L'algorithme d'enrichissement des requêtes est l'algorithme 4 ci-après.

Remarque: Certaines parties de l'algorithme ont été volontairement omises pour des raisons de lisibilité. Les partie manquantes s'articulent de la même façon que celles détaillées dans la première moitié de l'algorithme.

L'algorithme s'appuie sur un ensemble de fonctions et de procédures dont les fonctionnalités sont les suivantes :

- *determinerPattern(Requête)* : Cette fonction permet de déterminer la forme de la requête initiale donnée en paramètre. Les valeurs de sortie respectent l'ordre des patterns figurant dans la colonne de gauche du tableau 5.4.
- *ExtraireOppose(Requête,Ontologie adaptative)* : Cette fonction permet d'extraire les antonymes des mots clés composant la requête initiale à partir de l'ontologie adaptative toutes deux données en paramètres.
- *AppliquerRègle(Identifiant,Liste_ règle,Ensemble)* : Cette fonction applique la règle *identifiant* de la *liste de règles* en utilisant les mots clés fournis dans l'*ensemble* pour produire la requête enrichie à retourner à l'utilisateur.
- *Reduire(Ensemble)* : Cette fonction permet de réduire l'ensemble des mots clés passé en paramètre suivant les caractéristiques de l'ontologie adaptative du domaine donné en entrée de l'algorithme.
- *ExtraireSynonyme(Requête,Ontologie adaptative)* : Cette fonction permet d'extraire les synonymes des mots clés composant la requête initiale à partir de l'ontologie adaptative

Algorithme 4 Algorithme d'enrichissement des requêtes

Entrées: R_i une requête, LR la liste des règles, O_{dom}, O_{user} des ontologies adaptatives,
Sorties: R_e la requête enrichie

$p \longleftarrow determinerPattern(R_i)$,
si $p = 1$ **alors** {La requête est de la forme $-\omega : O$}
  $Op \longleftarrow ExtraireOppose(R_i, O_{dom})$
  si $Op = \emptyset$ **alors**
    $R_e \longleftarrow AppliquerRegle(2, LR, \emptyset)$ {Application de la règle 2}
  sinon
    $K \longleftarrow Reduire(Op)$
    $R_e \longleftarrow AppliquerRegle(1, LR, K)$ {Application de la règle 1}
  finsi
sinon si $p = 2$ **alors** {La requête est de la forme $\omega : O$}
  $Sy \longleftarrow ExtraireSynonyme(R_i, O_{dom})$
  si $Sy = \emptyset$ **alors**
    $Hy \longleftarrow ExtraireHyperonyme(R_i, O_{dom})$
    $Hy \longleftarrow ExtraireInstance(R_i, O_{dom})$
    si $Hy = \emptyset$ **alors**
      $Me \longleftarrow ExtraireHolonyme(R_i, O_{dom})$
      si $Me = \emptyset$ **alors**
        $Op \longleftarrow ExtraireOppose(R_i, O_{dom})$
        si $Op = \emptyset$ **alors**
          Retourner R_i
        sinon
          $K \longleftarrow Reduire(Op)$
          $R_e \longleftarrow AppliquerRegle(6, LR, K)$ {Application de la règle 6}
        finsi
      sinon
        $K \longleftarrow Reduire(Me)$
        $R_e \longleftarrow AppliquerRegle(5, LR, K)$ {Application de la règle 5}
      finsi
    sinon
      $K \longleftarrow Reduire(Hy)$
      $R_e \longleftarrow AppliquerRegle(4, LR, K)$ {Application de la règle 4}
    finsi
  sinon
    $K \longleftarrow Reduire(Sy)$
    $R_e \longleftarrow AppliquerRegle(3, LR, K)$ {Application de la règle 3}
  finsi
sinon si $p = 3$ **alors** {application des règles 7-8-9-10}

sinon si $p = 4$ **alors** {application des règles 11-12-13-14}

sinon si $p = 5$ **alors** {application des règles 15-16-17}

finsi
$c \longleftarrow ExtraireConceptsImportants(O_{user})$
Retourner $AjouterConcepts(c, R_e)$

toutes deux données en paramètres.

- *ExtraireHyperonyme(Requête,Ontologie adaptative)* : Cette fonction permet d'extraire les hypéronymes des mots clés composant la requête initiale à partir de l'ontologie adaptative toutes deux données en paramètres.
- *ExtraireInstance(Requête,Ontologie adaptative)* : Cette fonction permet d'extraire les instances des mots clés composant la requête initiale à partir de l'ontologie adaptative toutes deux données en paramètres.
- *ExtraireHolonyme(Requête,Ontologie adaptative)* : Cette fonction permet d'extraire les holonymes (inverse de méronyme) des mots clés composant la requête initiale à partir de l'ontologie adaptative toutes deux données en paramètres.
- *ExtraireConceptsImportants(Ontologie adaptative)* : Cette fonction permet d'extraire les labels des concepts les plus importants du point de vue du poids sémantique de l'ontologie représentant la catégorie d'utilisateurs sélectionnée.
- *AjouterConcepts(Ensemble,Requête)* : Cette fonction permet d'ajouter les concepts extraits par la fonction *ExtraireConceptsImportants* à la requête enrichie. L'ajout des termes et fait de telle façon que lors de l'interprétation de la requête, le système est forcé de retourner les pages contenant au moins un des termes importants de l'ontologie de l'utilisateur.

Exemple: Pour illustrer le fonctionnement de l'algorithme 4 pour l'enrichissement de requête considérons le cas où la requête initiale est *voiture :locomotion* et les ontologies utilisées sont représentées sur les figures 5.2 (pour l'ontologie adaptative du domaine) et 5.3 (pour l'ontologie représentant la vue de l'utilisateur sur le domaine).

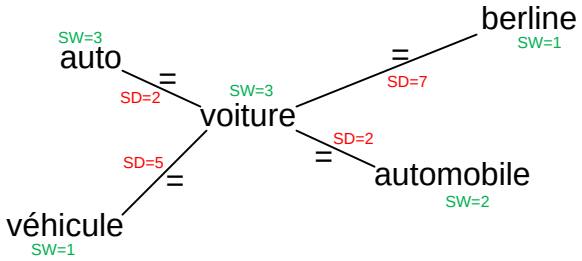


FIGURE 5.2 – Partie de l'ontologie du domaine

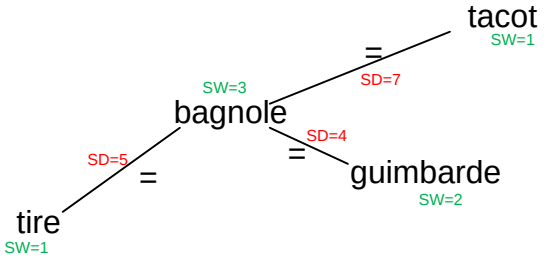


FIGURE 5.3 – Partie de l'ontologie de la catégorie d'utilisateurs

L'enrichissement de la requête initiale consiste dans un premier temps en l'identification de la forme de la requête initiale. Dans notre cas, la requête est de la forme $\omega : O$.

Dans un deuxième temps, l'algorithme procède à l'extraction des termes à ajouter à la requête en s'appuyant sur les éléments de l'ontologie adaptative du domaine. Dans notre exemple, le terme *voiture* de la requête initiale a 4 synonymes dans l'ontologie du domaine. L'idée est alors de s'appuyer sur les éléments de l'ontologie adaptative pour réduire cet ensemble de synonymes. L'exploitation de la distance sémantique permet d'éliminer les candidats *véhicule* et *berline*, puis l'exploitation du poids sémantique permet d'éliminer le terme *automobile* pour ne conserver que le terme *auto*. La règle 3 d'enrichissement est alors appliquée et la requête enrichie est donc : *voiture&auto&locomotion*.

Dans un troisième temps, l'ontologie de l'utilisateur est analysée. Dans notre exemple, cette ontologie représente la catégorie des jeunes et le concept le plus important de l'ontologie est *bagnole*. La requête enrichie finale est donc dans notre exemple *voiture&auto&locomotion&bagnole*. Cette requête est finalement évaluée par le système.

5.5 Conclusion

Dans ce chapitre, nous avons présenté le langage de requête ASK, adapté aux structures de données que sont les WPGraphs et W^{3}Graphs, ainsi qu'un mécanisme d'enrichissement des requêtes ASK basé sur certaines relations ontologiques et sur les éléments spécifiques des ontologies adaptatives.

Le langage ASK est construit selon un modèle booléen et sa syntaxe est inspirée de celle des langages de requêtes des principaux moteurs de recherche. Il propose un ensemble de constructeurs pour connecter les mots clés de la requête et pour exprimer le type de document recherché. La sémantique du langage est définie de telle sorte que les requêtes construites en ASK respectent la structure logique des WPGraphs et W^{3}Graphs.

Le mécanisme d'enrichissement des requêtes ASK basé sur les ontologies adaptatives du domaine et de la catégorie d'utilisateurs que nous proposons s'appuie sur un ensemble de 17 règles. Ces règles sont définies en tenant compte de la forme de la requête initiale et d'un ensemble de relations ontologiques plus grand que celui utilisé dans les approches existantes. Parmi les relations exploitées, nous avons proposé l'équivalence, la subsumption, l'instanciation, la méronymie et la relation d'opposition. L'exploitation de ces relations est complétée par l'utilisation des éléments caractéristiques des ontologies adaptatives dans le but de sélectionner les mots clés les plus pertinents pour enrichir la requête initiale. Enfin, l'enrichissement de la requête est complété par la sélection des concepts de l'ontologie des vues de l'utilisateur dont le poids sémantique est le plus élevé. Cet enrichissement a pour but de contraindre davantage la requête pour réduire l'espace des documents visés.

Les requêtes enrichies suivant les règles que nous avons présentées dans ce chapitre doivent à présent être évaluées sur les WPGraphs et W^{3}Graphs afin d'en extraire les informations les plus pertinentes. De plus, les résultats validant la requête enrichie doivent être classés de manière adéquate afin que l'utilisateur ayant posé la requête puisse accéder directement à l'information qu'il recherche. Ces deux points sont discutés dans le chapitre suivant.

Chapitre 6

Extraction de l'information pertinente et classement des résultats

Sommaire

L'évaluation des requêtes, l'extraction de l'information pertinente et le classement des résultats sont trois éléments fondamentaux dans les systèmes dédiés à la recherche d'information. Ces trois composantes jouent un rôle prépondérant dans la qualité des résultats retournés aux utilisateurs.

La méthode d'évaluation des requêtes c'est à dire le processus suivant lequel le système interprète la requête est importante pour les utilisateurs. Ce facteur va grandement influencer la façon dont les utilisateurs vont construire leurs requêtes. L'expérience accumulée par un utilisateur au cours des différentes interactions que ce dernier a eues avec l'application lui permet de comprendre la façon dont le système interprète les requêtes. De ce fait, il va pouvoir anticiper ce phénomène en construisant directement de bonnes requêtes, c'est-à-dire des requêtes lui retournant des résultats proches de ceux qu'il attend. Ce point peut également décider de la popularité de l'application. Si l'interprétation de la requête est totalement anarchique et ne conduit jamais aux résultats escomptés, les utilisateurs vont progressivement délaisser l'application au profit de systèmes concurrents.

Après avoir évaluée la requête, l'application s'appuie sur les structures de données sous-jacentes pour en extraire l'information la plus pertinente par rapport à son interprétation de la requête. L'extraction des informations doit être efficace afin de répondre le plus rapidement possible aux requêtes des utilisateurs. Par conséquent, un algorithme adapté aux structures de données utilisées par l'application doit permettre cette extraction efficace.

Enfin, le classement des résultats est également capital pour les systèmes de recherche d'information. Dans le cadre du Web, l'algorithme du PageRank utilisé par Google pour classer les

résultats est même le fondement de la réussite de ce moteur de recherche. La méthode de classement des résultats doit permettre à l'utilisateur d'accéder directement à l'information la plus pertinente en fonction de sa requête initiale. Les concepts introduits tout au long de ce mémoire, et principalement l'ontologie adaptative du domaine, permettent de définir une méthode de classement des pages Web selon la sémantique de leur contenu. L'utilisateur peut alors accéder plus directement à l'information pertinente.

Au cours de ce chapitre, nous présentons les trois éléments évoqués dans les paragraphes précédents. Nous introduisons, dans un premier temps, la façon dont les requêtes ASK enrichies sont évaluées. Puis nous présentons l'algorithme d'extraction de l'information pertinente vérifiant la requête des structures de données que sont les WPGraphs et W^3Graphs. Pour finir, nous discutons de la méthode de classement des résultats permettant de satisfaire au mieux les besoins des utilisateurs dans leur quête d'informations.

6.1 Motivations

Faciliter le travail de construction des requêtes aux utilisateurs constitue notre motivation principale pour la définition d'une méthode d'évaluation des requêtes facile à appréhender. Comme mentionnée dans l'introduction de ce chapitre, la façon selon laquelle l'application évalue les requêtes influence à moyen terme la manière dont les utilisateurs posent leurs requêtes. L'idée de construire une application qui va évaluer les requêtes comme l'attendent les utilisateurs va contribuer à améliorer la pertinence des résultats.

Ensuite, l'utilisateur doit pouvoir consulter les résultats en réponse à la requête posée dans un temps acceptable. En conséquence, les algorithmes d'extraction de l'information pertinente vérifiant la requête doivent permettre de répondre à ce critère. Cet aspect constitue une motivation suffisante pour la proposition d'un algorithme efficace pour l'extraction des données pertinentes des structures que sont les WPGraphs et W^3Graphs.

Le classement des résultats doit permettre à l'utilisateur de retrouver l'information pertinente en priorité. Cependant, dans notre analyse des travaux existants dans ce domaine, nous avons montré que les approches existantes pour le classement des résultats utilisent la structure du Web et notamment les hyperliens pointant d'une page à une autre. Notre motivation est de proposer une approche de classement des résultats en s'appuyant sur la sémantique du contenu des pages Web. L'objectif est de solutionner en partie les problèmes posés par l'utilisation de la structure du Web.

Dans ce chapitre, nous allons aborder successivement les trois points évoqués dans cette partie. Nous allons commencer par l'évaluation des requêtes, puis nous enchaînons avec l'extraction des informations pertinentes et pour finir nous abordons le problème du classement des résultats.

6.2 Evaluation des requêtes et extraction de l'information pertinente

L'évaluation des requêtes et la sélection des pages Web pertinentes sont deux aspects fortement liés. Ils reposent tous deux sur une base commune à savoir la sémantique du langage de

requête et les structures de données contenant les informations du Web à extraire. Dans cette section, nous allons discuter de ces deux aspects. Nous commençons par présenter l'évaluation des requêtes ASK dans notre approche. Nous introduisons ensuite l'algorithme d'extraction de l'information du Web contenue dans les WPGraphs et les W^{3}Graphs répondant à la requête enrichie.

6.2.1 Evaluation des requêtes ASK

Le système en charge de l'évaluation des requêtes ASK procède suivant deux étapes fondamentales. Ces deux étapes s'exécutent successivement et ont pour objectifs :

1. L'analyse et la traduction de la requête dans un langage adapté facilitant à la fois son interprétation et l'extraction de l'information pertinente.
2. La compréhension de la requête avec pour objectif l'extraction de l'information la plus pertinente.

Au cours de la phase d'analyse, le système vérifie si la requête est bien formée d'un point de vue syntaxique. Pour la réalisation de cette tâche, nous nous appuyons sur la grammaire du langage ASK donnée au chapitre 5 section 5.1.2. L'analyse syntaxique est faite, à l'instar des approches existantes, grâce à la construction de l'arbre d'analyse syntaxique. Cela consiste en la réécriture d'une requête ASK sous la forme d'arbre pour, d'une part, faciliter la vérification de la syntaxe de la requête (i.e. si la requête respecte bien la syntaxe du langage ASK) et, d'autre part, permettre (éventuellement) la traduction de la requête dans un langage facilitant l'extraction des données des structures sous-jacentes. D'un point de vue plus technique, nous avons utilisé le générateur CUP [1] pour automatiser la construction de l'arbre d'analyse syntaxique (nous détaillons ce point au chapitre suivant).

Une fois que le système est assuré que la requête est syntaxiquement valide, il poursuit avec la traduction de la requête. La traduction de la requête se fait selon un ensemble de règles (quelque peu similaires à celles de la section 5.1.2), il en résulte une expression algébrique équivalente à la requête ASK soumise. Ces expressions algébriques vont par la suite permettre l'extraction des données des WPGraphs et W^{3}Graphs. Les règles que nous avons définies se présentent sous la forme :

$$< Expression\ ASK > \Longrightarrow < Expression\ algebrique\ correspondante >$$

Elles permettent la mise en correspondance des expressions ASK avec les expressions exprimées dans l'algèbre associée au langage. Les règles de traduction sont celles définies dans le tableau 5.2 du chapitre 5. Elles se proposent de traduire les expressions ASK en expression algébrique s'appuyant sur la structure logique des WPGraphs et W^{3}Graphs. Dans les règles qui suivent, $SLG_{GD} =< D_{GD}, V_{GD}, E_{GD}, \varphi_{GD}, \rho_{vGD}, \rho_{eGD} >$ représente la structure d'un WPGraph.

1. http ://www.cs.princeton.edu/∼ appel/modern/java/CUP/

Traduction des expressions simples :

$$\begin{aligned}
t : d &\Longrightarrow t \in V_{GD} \\
t.a(= v) : d &\Longrightarrow t \in V_{GD} \wedge a \in V_{GD} \wedge v \in V_{GD} \\
&\quad \wedge \rho_e(t,a) = \rho_e(a,v) \wedge \rho_e(a,v) = \rho_e(t,v) \\
t\{img\} &\Longrightarrow \exists X \in D_{GD} \wedge t \in V_{GD} \wedge \varphi_{GD}(t) = (X, img) \\
t\{snd\} &\Longrightarrow \exists X \in D_{GD} \wedge t \in V_{GD} \wedge \varphi_{GD}(t) = (X, snd) \\
t\{vid\} &\Longrightarrow \exists X \in D_{GD} \wedge t \in V_{GD} \wedge \varphi_{GD}(t) = (X, vid) \\
t\{fil\} &\Longrightarrow \exists X \in D_{GD} \wedge t \in V_{GD} \wedge \varphi_{GD}(t) = (X, fil) \\
t\{empty\} &\Longrightarrow \exists X \in D_{GD} \wedge t \in V_{GD} \wedge \varphi_{GD}(t) = (X, \perp)
\end{aligned}$$

Traduction des opérations sur les termes :

$$\begin{aligned}
(< Expr.\ ASK >) &\Longrightarrow (< Expr.\ alg. >) \\
< Expr.\ ASK > \,\&\, < Expr.\ ASK > &\Longrightarrow < Expr.\ alg. > \wedge < Expr.\ alg. > \\
< Expr.\ ASK > \,|\, < Expr.\ ASK > &\Longrightarrow < Expr.\ alg. > \vee < Expr.\ alg. > \\
< Expr.\ ASK > \,\#\, < Expr.\ ASK > &\Longrightarrow (< Expr.\ alg. > \vee < Expr.\ alg. >) \wedge \\
&\quad \neg(< Expr.\ alg. > \wedge < Expr.\ alg. >) \\
!\, < Expr.\ ASK > &\Longrightarrow \neg < Expr.\ alg. > \\
-\, < Expr.\ ASK > &\Longrightarrow \textit{Pas de traduction directe}
\end{aligned}$$

Traduction des autres primitives :

$$\begin{aligned}
< Expr.\ ASK > \{img\} &\Longrightarrow < Expr.\ alg. > \wedge \varphi_{GD}(\ldots) = (X, img) \\
< Expr.\ ASK > \{snd\} &\Longrightarrow < Expr.\ alg. > \wedge \varphi_{GD}(\ldots) = (X, snd) \\
< Expr.\ ASK > \{vid\} &\Longrightarrow < Expr.\ alg. > \wedge \varphi_{GD}(\ldots) = (X, vid) \\
< Expr.\ ASK > \{fil\} &\Longrightarrow < Expr.\ alg. > \wedge \varphi_{GD}(\ldots) = (X, fil) \\
< Expr.\ ASK > \{empty\} &\Longrightarrow < Expr.\ alg. > \wedge \varphi_{GD}(\ldots) = (X, \perp)
\end{aligned}$$

Remarque: Le symbole "..." utilisé dans la définition des règles remplace les termes de l'expression algébrique correspondante.

La phase de traduction précède la phase de compréhension de la requête. Au cours de cette phase, la sémantique du langage est utilisée. Une optimisation des requêtes peut également être effectuée à travers laquelle une réécriture (totale ou partielle) de la requête peut être faite. La requête traduite selon les règles de traduction est ensuite interprétée sur la structure logique des WPGraphs et W^{3}Graphs pour en extraire l'information pertinente et répondre ainsi à la demande initiale de l'utilisateur.

Remarque: Pour des raisons pratiques, une dernière traduction du langage algébrique en langage machine est nécessaire pour que le système automatise le traitement des requêtes. Nous reviendrons plus en détails sur cet aspect dans le dernier chapitre de ce mémoire consacré à l'implémentation de notre approche.

6.2.2 Algorithmes de vérification des requêtes ASK

Un W^{3}Graph est une structure de graphe dont les sommets contiennent l'information du Web, sous la forme d'un WPGraph (nous rappelons qu'un WPGraph contient une représentation enrichie du contenu d'une page Web), et les arêtes représentent un lien sémantique entre les sommets (i.e. il existe un lien plus ou moins fort entre deux sommets si ceux-ci sont proches du point de vue de la sémantique de leur contenu). La vérification des requêtes ASK qui détermine la sélection des pages pertinentes à partir des WPGraphs et W^{3}Graphs est faite à deux niveaux différents :

1. A un niveau «local», c'est-à-dire au niveau d'un sommet du W^{3}Graph. On vérifie que l'information contenue dans un WPGraph permet de valider la requête ASK.
2. A un niveau «global», c'est-à-dire sur l'ensemble des sommets du W^{3}Graph. Les pages Web dont les WPGraph associés vérifient la requête sont définies comme pertinentes. La liste des URL correspondants à ces pages Web est alors classée puis retournée à l'utilisateur.

Les deux algorithmes définis ci-après décrivent les opérations effectuées à chacun des deux niveaux. L'algorithme 5 illustre le processus de vérification d'une requête ASK sur un WPGraph (i.e. au niveau local) alors que l'algorithme 6 s'applique au niveau global. Par conséquent, il traite de l'évaluation de la requête enrichie sur l'ensemble des WPGraphs composant le W^{3}Graph.

Algorithme de vérification d'une requête ASK sur la structure logique d'un WPGraph

Pour vérifier la requête, nous nous appuyons sur la structure logique d'un WPGraph (voir section 4.4.1). Dans la mesure où la formalisation du langage ASK et celle d'un WPGraph sont homogènes, l'algorithme que nous proposons consiste en l'interprétation de la formule logique représentant la requête ASK sur la structure logique d'un WPGraph. Une page Web sera considérée comme pertinente si la structure logique du WPGraph qui lui est associée vérifie complètement la formule de logique correspondant à la requête posée.

Algorithme 5 Algorithme de vérification d'une requête ASK sur un WPGraph

Entrées: R_e Une requête ASK enrichie
WP Un WPGraph
Sorties: Booléen
$L1[] \longleftarrow SplitQuery(et, R_e)$
pour i=1 à L1.longueur **faire**
$L2[] \longleftarrow SplitQuery(ou, L1[i])$
pour j=1 à L2.longueur **faire**
si Verifier(L2[j],WP)=Vrai **alors**
break
finsi
fin pour
si j=(L2.longueur + 1) **alors**
Retourner Faux
finsi
fin pour
Retourner Vrai.

Dans l'algorithme 5 ci-dessus, les trois fonctions et procédures utilisées produisent les résultats suivants :

- *SplitQuery(connecteur, Requête ASK)* : Cette fonction découpe la formule de logique correspondant à la requête ASK par rapport au connecteur logique passé en paramètre. Le résultat est un tableau de termes.
- *longueur* retourne la longueur du tableau auquel elle s'applique.
- *Verifier(formule,WPGraph)* : Cette fonction parcourt la structure logique du WPGraph pour vérifier la formule de logique passée en paramètre. La fonction retourne vrai si elle parvient à vérifier la formule et faux dans le cas contraire.

L'algorithme que nous proposons s'exécute au pire des cas en temps en $O(nbterme \times nbsommet)$ où *nbterme* représente le nombre de termes de la formule à vérifier et *nbsommet* représente le nombre de sommets composant le WPGraph contenant l'information.

Remarque: La définition de cet algorithme dépend fortement du langage dans lequel il sera implémenté. Pour notre part, nous utilisons le langage PROLOG et nous discutons de l'implémentation de cet algorithme dans ce langage au chapitre 7.

Algorithme de vérification d'une requête ASK sur un W^{3}Graphs

Dans sa quête d'information sur le Web, l'utilisateur attend principalement deux choses. Premièrement, il attend de se voir proposer des résultats pertinents par rapport à ses besoins initiaux et deuxièmement les résultats de la recherche doivent lui parvenir dans un temps acceptable. Ce second point nous oblige à proposer un algorithme qui s'exécute en temps acceptable afin de répondre efficacement à la requête de l'utilisateur. L'algorithme que nous proposons exploite la structure de graphe d'un W^{3}Graph et principalement le poids attribué à chacune de ses arêtes.

Le poids attribué à chaque arête du W^{3}Graph représente la force du lien sémantique reliant deux sommets du graphe. En conséquence, plus le poids est élevé, plus les deux sommets formant l'arête sont proches du point de vue de la sémantique de leur contenu. C'est donc sur cette propriété que nous allons axer le parcours du W^{3}Graph. L'idée est de parcourir les voisins les plus proches d'un sommet vérifiant la requête car ces derniers ont une probabilité plus grande de vérifier la requête et de parcourir les sommets éloignés ou isolés en dernier.

L'algorithme 6 met en œuvre six fonctions et procédures dont l'exécution se détaille comme suit :

- *ChoisirSommet(Ensemble)* : Cette fonction permet de choisir aléatoirement un sommet du W^{3}Graph. Le choix est fait en priorité dans les parties connexes du W^{3}Graph.
- *Enlever(Sommet)* : Cette procédure permet de retirer le sommet passé en paramètre de l'ensemble des sommets du graphe restant à visiter.
- *Verifier(Requête ASK ,WPGraph)* : Cette fonction correspond à l'algorithme 5 de vérification d'une requête ASK sur un WPGraph présenté ci-dessus.
- *Classer(WPGraph ,Liste)* : Cette procédure permet de classer l'URL de la page Web associé au WPGraph passé en paramètre dans la liste d'URL des pages considérées comme pertinentes. La définition de cette procédure est discutée plus largement à la section 6.3.
- *ChoisirSommetProche(WPGraph ,Ensemble)* : Cette fonction a pour objectif de déterminer le sommet de l'ensemble le plus proche (du point de vue de la sémantique du contenu) du WPGraph passé en paramètre. Ce choix est effectué suivant le poids attribué à chaque arête du *W^{3}Graph*.
- *ChoisirSommetEloigne(WPGraph ,Ensemble)* : Cette fonction a pour objectif de détermi-

Algorithme 6 Algorithme d'extraction de l'information d'un W^{3}Graph

Entrées: R_e Une requête ASK enrichie
$WWW = (S, A)$ un W^{3}Graph
Sorties: URL Une liste de d'url

```
URL ⟵ ∅
σ ⟵ ChoisirSommet(A)
tantque S ≠ ∅ faire
  Enlever(σ, A)
  si Verifier(R_e, σ) = true alors
    Classer(σ, URL)
    σ ⟵ ChoisirSommetProche(σ, A)
  sinon
    σ ⟵ ChoisirSommetEloigne(σ, A)
  finsi
  si σ = null alors
    σ ⟵ ChoisirSommet(A)
    si σ = null alors
      σ ⟵ ChoisirSommet(S)
    finsi
  finsi
fin tantque
Retourner URL.
```

ner le sommet de l'ensemble le plus éloigné (du point de vue de la sémantique du contenu) du WPGraph passé en paramètre.

Du point de vue de la complexité en temps, la suite d'instruction de l'algorithme consiste en un simple parcours de graphe. En conséquence, l'algorithme ne visite qu'une seule fois chaque sommet du graphe d'où une complexité en $O(|S|)$.

Dans cette section, nous avons présenté les algorithmes mis en œuvre dans l'évaluation des requêtes ASK. Il nous est alors nécessaire de discuter de la façon dont les résultats pertinents sont classés pour satisfaire pleinement les utilisateurs afin qu'ils perdent le moins de temps possible lors de la consultation des résultats renvoyés par l'application.

6.3 Classement des résultats

Le classement des résultats est un des aspects les plus importants dans les approches pour la recherche d'information. L'algorithme pour le classement des résultats est même à l'origine de la réussite de bon nombre d'applications. Dans cette section, nous discutons de la méthode utilisée dans notre approche pour classer les résultats d'une recherche documentaire par ordre de pertinence.

A l'instar du PageRank, les principaux algorithmes de classement des résultats d'une recherche d'information sur le Web s'appuient sur la structure du Web et principalement sur les hyperliens pointant d'une page vers une autre. Le choix d'utiliser la structure du Web est la conséquence directe de la popularité de l'algorithme du PageRank exploité par Google. Cet algorithme

comporte néanmoins quelques désavantages ayant un impact non négligeable sur la pertinence des pages Web retournées. Parmi les principaux inconvénients du PageRank on retrouve :

- Le caractère facilement influençable du calcul du PageRank associé à chaque page Web. Comme l'algorithme compte les liens pointant vers une page il suffit de construire un certain nombre de pages contenant un hyperlien pointant sur la page dont on souhaite augmenter le PageRank. Les sites Web a caractère commercial exploite cette faille afin que leurs applications soient référencées en priorité par le moteur de recherche.
- L'algorithme ne favorise pas l'évolution du Web et de son contenu. Les pages les plus anciennes du Web sont celles qui sont les plus référencées par les autres pages existantes. En conséquence, elles auront un PageRank élevé contrairement aux pages fraîchement publiées. Pour corriger ce problème, Google met en œuvre, en plus du PageRank, d'autres algorithmes plus complexes tenant compte notamment de la date de création et du contenu des pages Web.
- L'impact des pages les plus intéressantes sur le PageRank des autres pages qui lui sont connectées est limité par la longueur et le nombre d'hyperliens contenu dans la page intéressante.

L'ensemble des inconvénients que nous venons d'évoquer peut partiellement être corrigé par l'utilisation de la sémantique du contenu des pages Web pour les classer après une recherche documentaire sur le Web. En conséquence, notre méthode de classement des résultats est basée sur le contenu des pages Web et non sur la structure du Web. La seule structure sur laquelle nous nous appuyons est celle des WPGraphs qui, rappelons le, sont construits à partir des ontologies adaptatives et de la sémantique du contenu des pages.

Comme tous les algorithmes de classement des résultats, notre méthode consiste à ordonner les pages Web validant la requête enrichie selon leur pertinence. Cette dernière est, dans notre approche, définie par rapport au domaine de recherche. Les pages considérées comme les plus pertinentes sont celles les plus en rapport avec le domaine dans lequel a été émise la requête initiale. Nous mesurons le degré de pertinence d'une page Web grâce aux nombres d'arêtes composant le WPGraph associé à la page Web.

Définition 6.1 *Soit $WP = (V, E, T, \varphi, \rho_v, \rho_e)$ un WPGraph associé à une page Web Ω. On définie le degré de pertinence de Ω et on note $deg(\Omega)$ la quantité telle que :*

$$deg(\Omega) = |E|$$

Suivant cette définition, un degré de pertinence d'une page Web élevé représente un nombre d'arêtes important dans le WPGraph qui lui est associé. Comme cet ensemble est construit suivant l'ontologie adaptative du domaine, les arêtes du WPGraph représentent les liens sémantiques entre les termes de la page Web et sont donc significatives de l'appartenance de la page Web au domaine représenté par l'ontologie adaptative. En conséquence, nous proposons de classer les pages Web par ordre décroissant de leur degré de pertinence.

Notre algorithme de classement des résultats (algorithme 7) procède à un tri des pages Web vérifiant la requête enrichie suivant leur degré de pertinence. Il correspond à la procédure *Classer(WPGraph, liste)* utilisée dans l'algorithme 6 et permet de classer au fur et à mesure les pages Web définies comme pertinentes.

Dans l'algorithme 7 des procédures et fonctions sont utilisées chacune d'entre elles produit les résultats suivants :

Algorithme 7 Classement des résultats pertinents

Entrées: $WP = (V, E, T, \varphi, \rho_v, \rho_e)$ Un WPGraph
URL Une liste d'URL
Sorties: URL Une liste de d'url
si $URL = \emptyset$ **alors**
$URL \longleftarrow Inserer(Adresse(WP), 1)$
sinon
$i \longleftarrow 1$
tantque $i \leq |URL|$ **faire**
si $|E| \geq NbAretesWPGraph(getWPGraph(URL(i)))$ **alors**
Inserer(Adresse(WP),i)
Retourner URL
sinon
$i \longleftarrow i + 1$
finsi
fin tantque
finsi
Inserer(Adresse(WP),i)
Retourner URL.

- *Inserer(url,entier)* : Cette fonction insère l'adresse *url* à la position donnée en paramètre. La méthode d'insertion est fonction de la structure de donnée *URL* et du langage dans lequel est implémenté l'algorithme.
- *Adresse(WPGraph)* : Cette fonction retourne l'adresse url du WPGraph passé en paramètre.
- *NbAretesWPGraph(WPGraph)* : Cette fonction calcule le nombre d'arête du WPGraph passé en paramètre.
- *getWPGraph(url)* : Cette fonction est la fonction inverse de la fonction *Adress(WPGraph)*. En conséquence, elle retourne le WPGraph associé à la page Web dont l'adresse est passée en paramètre.

6.4 Exemple

Le fonctionnement des différents algorithmes introduits dans ce chapitre nécessite d'être illustré sur un exemple.

Dans un premier temps, nous allons illustrer le principe de l'algorithme 6 (parcours d'un W^{3}Graph). Pour cela nous nous basons sur le W^{3}Graph de la figure 6.1 et nous supposons que la requête enrichie est : $A\&(B|C)$

Au cours de la première étape, l'algorithme procède au choix du premier sommet du graphe à évaluer. Le choix est fait aléatoirement en priorité parmi les sommets des parties connexes du W^{3}Graph. Supposons que le choix est porté sur le sommet WP_1.

La seconde étape consiste à vérifier la requête enrichie sur le WPGraph choisie selon l'algorithme 5. Nous allons détailler cette phase sur le WPgraph représenté sur la figure 6.2. Tout d'abord, la requête est découpée par rapport à l'opérateur de conjonction. Dans notre exemple, nous obtenons deux parties. La première partie contenant A et la seconde partie $B|C$. Ensuite une deuxième division des parties de la requête intervient selon l'opérateur de disjonction. Dans

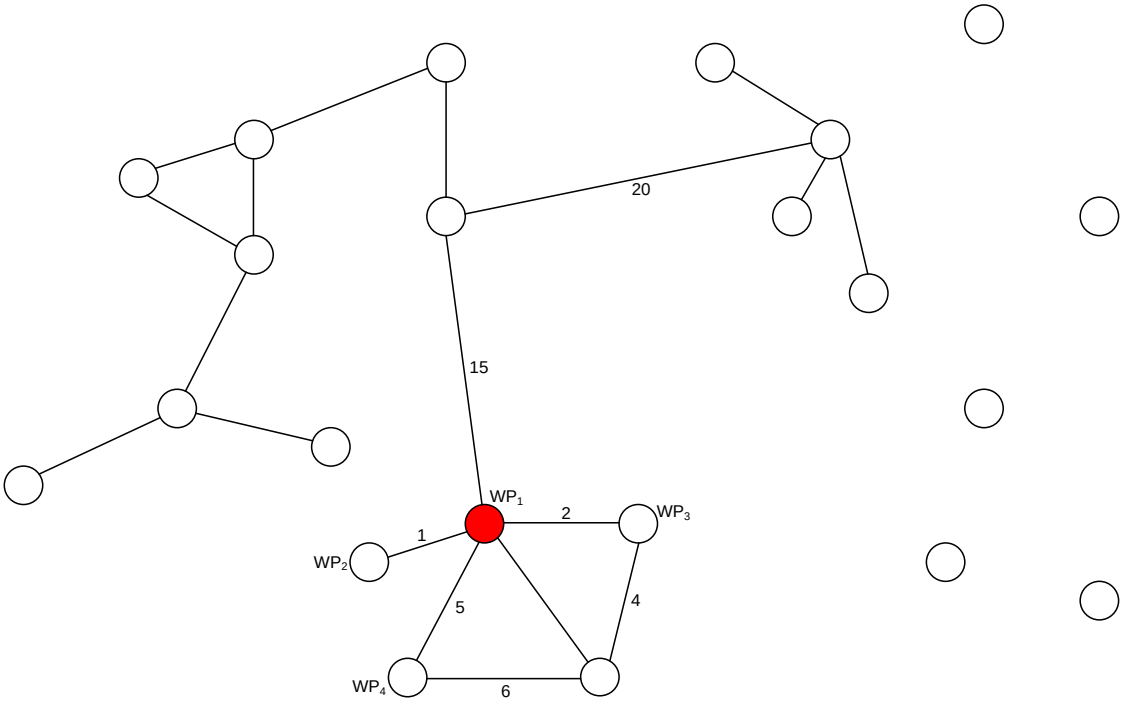


FIGURE 6.1 – Exemple de W^{3}Graph

cet exemple, la partie A ne peut pas être divisée par contre $B|C$ est divisé en B et C. Les différentes parties sont ensuite vérifiées sur la structure du WPGraph WP_1. La partie A est bien vérifiée tout comme les parties B et C, en conséquence, le WPGraph WP_1 est considéré comme pertinent et l'URL de la page Web qui lui associé est classée. Comme dans cet première passe aucun sommet n'a encore été parcouru, cette URL est la seule pour l'instant.

La troisième partie étape consiste à choisir le sommet du W^{3}Graph à vérifier. Cette sélection est fonction de la proximité sémantique entre les sommets (i.e. le poids attribué à chaque arête). Dans notre exemple, le sommet le plus proche directement connecté à WP_1 est WP_2 (voir figure 6.3). La requête enrichie est ensuite évaluée de la façon expliquée au paragraphe précédent. WP_2 valide également la requête il s'agit alors de classer ce sommet. Comme dans cet exemple, $deg(W_{WP_1}) \leq deg(W_{WP_2})$, la page W_{WP_2} associée à WP_2 est plus pertinente que W_{WP_1}. L'URL de W_{WP_2} est donc insérée devant celle de W_{WP_1} dans la liste d'URL des pages pertinentes. L'algorithme retourne à nouveau à la première étape et choisit le sommet suivant du W^{3}Graph à visiter.

6.5 Conclusion

L'évaluation des requêtes et le classement des résultats pertinents sont deux des point clés de tous les systèmes de recherche d'information. Dans ce chapitre nous avons discuté de ces aspects. Nous avons tout d'abord présenté la façon d'évaluer les requêtes ASK puis nous avons introduit une nouvelle technique de classement des résultats pertinents en fonction de la sémantique du contenu des pages Web.

L'évaluation des requêtes ASK posées par les utilisateurs se déroule en plusieurs étapes. La première étape consiste en la traduction des requêtes dans une algèbre pour dans un second temps

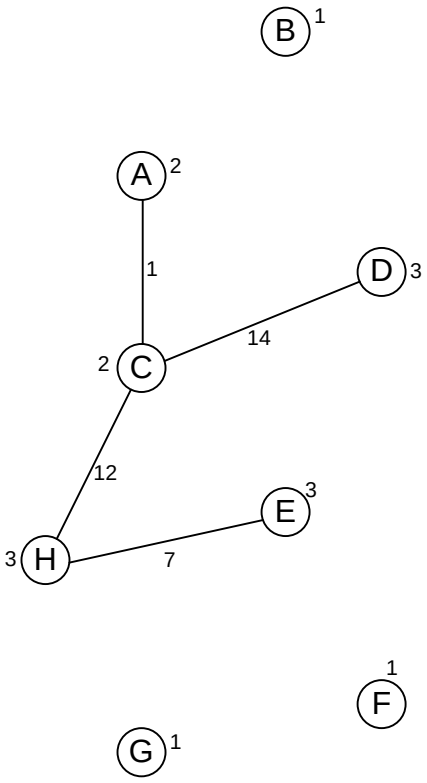


FIGURE 6.2 – Le WPGraph WP_1

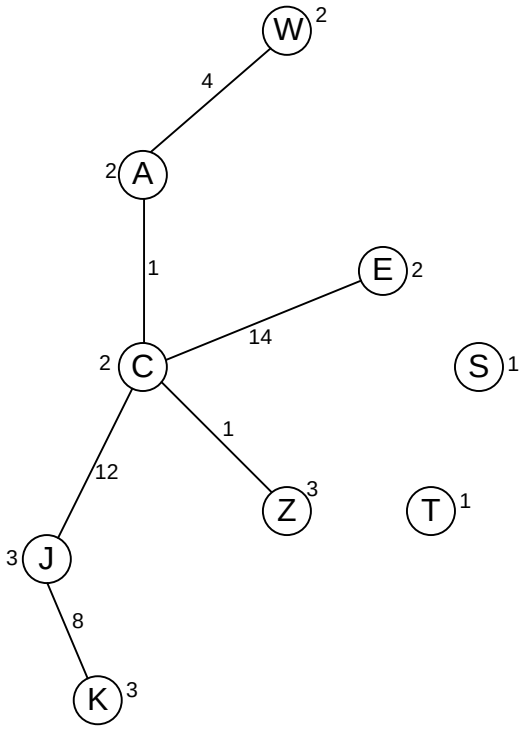


FIGURE 6.3 – Le WPGraph WP_2

vérifier les expressions algébriques correspondantes sur la structure logique des WPGraphs et W^3Graphs grâce à des algorithmes adaptés afin de déterminer quelles pages Web sont pertinentes par rapport à la requête enrichie.

Ensuite, la méthode de classement des résultats pertinents que nous proposons s'appuie sur la sémantique du contenu des pages Web et non sur la structure d'hyperlien du Web comme dans la plupart des approches existantes. Cette méthode permet de résoudre certains des problèmes posés par les applications s'appuyant sur la structure du Web et notamment le fait de privilégier les pages Web à caractère commercial. La pertinence des pages Web est définie en fonction de son appartenance au domaine de recherche visé par la requête et représenté par une ontologie adaptative.

L'ensemble des concepts introduits tout au long de ce mémoire nécessite alors d'être validés. Dans le chapitre suivant, nous présentons la validation expérimentale de notre approche à travers un cas d'étude consacré à la recherche d'articles scientifiques.

Chapitre 7

Etude de cas et validation expérimentale de l'approche

Sommaire

Nous avons, jusqu'à présent, introduit un ensemble de concepts pour l'évolution d'ontologies et la recherche d'information sur le Web. Cependant, nous n'avons pas encore démontré leur validité et leur utilité pour l'amélioration de la recherche d'informations pertinentes sur le Web, objectif principal de nos travaux. Par conséquent, ce dernier chapitre porte sur la validation expérimentale de ces concepts pour la recherche d'information pertinente sur le Web.

La réalisation d'une validation expérimentale nécessite la mise en application d'un cadre spécifique présentant un ensemble de propriétés caractéristiques des différents éléments à valider. Le cas d'étude synthétique que nous proposons pour la validation de nos travaux et que nous présentons dans la première partie de ce chapitre, représente une situation réelle et assimilée à une recherche d'information sur le Web telle que les utilisateurs ont l'habitude de mener. Le cas d'étude que nous avons utilisé porte sur la recherche d'articles scientifiques publiés à la conférence internationale *World Wide Web*. Il a été conçu dans le but de permettre l'évaluation qualitative et quantitative des concepts introduits dans les différents chapitre de ce mémoire et qui forment la méthode TARGET.

Les ontologies adaptatives définies au chapitre 2, ainsi que les mécanismes d'adaptation qui leurs sont associées, constituent les rouages principaux de l'approche TARGET mais aussi son originalité. Par conséquent, la création et la gestion de ce type d'ontologies à travers le temps jouent un rôle essentiel dans la qualité du processus décrivant la méthode TARGET. Pour faciliter la construction et surtout pour gérer efficacement l'évolution des ontologies adaptatives à travers le temps, un outil assistant l'utilisateur pour la réalisation de ces tâches est nécessaire. La présentation de l'outil développé est l'objet de la seconde partie de ce chapitre. Les fonctionnalités de l'outil pour la gestion des ontologies adaptatives sont construites sur les bases des caractéristiques des ontologies adaptatives et de celles propres à l'évolution d'ontologies détaillées dans les trois premiers chapitres de ce mémoire. L'outil que nous avons développé, présente également des fonctionnalités mettant en œuvre les ontologies adaptatives, les structures de données du Web (les WPGraphs et W^3Graphs) et le langage de requête ASK pour la recherche d'informations pertinentes sur le Web.

Cet outil est donc à la base de la validation expérimentale de notre approche car il a permis, d'une part, la mise en place d'un ensemble d'expérimentations et, d'autre part, l'obtention rapide des résultats escomptés. Les expérimentations que nous proposons répondent à un ensemble d'objectifs de validation et décrivent un ensemble de scénarios visant un concept particulier de l'approche à valider. Les résultats obtenus sont discutés afin de faire ressortir la contribution de chaque concept introduit dans ce manuscrit pour l'amélioration de la recherche documentaire sur le Web du point de vue de la pertinence des résultats.

Dans ce chapitre, nous introduisons tout d'abord le cas d'étude de travail sur lequel repose la validation expérimentale de notre approche. Nous présentons ensuite l'outil TARGET pour la gestion des ontologies adaptatives et la recherche d'information pertinente sur le Web. Nous montrons, dans une troisième partie, l'efficacité de l'outil et son exploitation afin de valider expérimentalement les concepts présentés dans ce mémoire. Dans ce sens, nous discutons des différents objectifs et scénarios mis en œuvre puis les nombreux résultats expérimentaux obtenus.

7.1 Motivations

La définition d'un cas d'étude est fondamentale pour la validation expérimentale d'un ensemble de concepts. Le niveau de réalisme du cadre expérimental proposé doit permettre une validation qualitative des concepts définis. Par conséquent, la motivation principale dans la construction d'un tel cadre expérimental reste la démonstration du bien-fondé de ces concepts et la mise en évidence de leur contribution dans l'objectif pour lequel ils sont définies à savoir dans notre cas l'amélioration de la recherche d'informations pertinentes sur le Web.

L'étude entreprise au chapitre premier de ce mémoire sur les approches existantes pour l'évolution d'ontologies a souligné un manque évident en matière d'outils efficaces pour la gestion de l'évolution des ontologies. La construction d'un outil implémentant les concepts et mécanismes pour la construction et la gestion de l'évolution des ontologies adaptatives permet de combler ce manque. De plus, l'outil doit permettre l'utilisation de ce type d'ontologies pour la recherche d'information pertinente sur le Web et ainsi constituer le fondement des expérimentations pour la validation de nos concepts.

Le troisième aspect qui a motivé ce chapitre concerne la description des différentes expé-

rimentations mises en œuvre dans le cadre de la validation des concepts. Le protocole expérimental ainsi que les différents scénarios que nous proposons vont permettre de valider un à un les concepts introduits dans ces travaux. De plus, la qualité de ces éléments va renforcer la qualité et la validité de l'approche TARGET.

Dans la suite de ce chapitre, nous allons détailler successivement notre cas d'étude, puis l'outil implémentant l'approche TARGET et, pour finir, les expérimentations ainsi que les résultats obtenus.

7.2 Cas d'étude : La conférence internationale World Wide Web

Dans cette section, nous présentons le cas d'étude mis en place pour la validation expérimentale des concepts. Dans un premier temps, nous décrivons le cas d'étude et ses caractéristiques, puis, dans un deuxième temps, nous détaillons de manière plus technique la mise en place des composants de ce cadre expérimental.

7.2.1 Définition du cas d'étude

Le cas d'étude mis en œuvre comporte plusieurs propriétés fondamentales liées aux différents concepts introduits dans ce mémoire. Comme évoqué au cours de l'introduction de ce chapitre, le cas d'étude doit nous permettre de valider à la fois les concepts relatifs aux ontologies adaptatives et aussi à l'exploitation de ce type d'ontologies à travers les mécanismes d'expansion de requêtes visant à améliorer la recherche d'information sur le Web.

La validation des éléments caractéristiques des ontologies adaptatives et des règles d'adaptation présentées au chapitre 3 requiert des caractéristiques particulières. Rappelons que dans notre approche, nous utilisons les ontologies adaptatives pour représenter les connaissances d'un domaine, les documents qui le composent et les vues des différentes catégories d'utilisateurs sur ce domaine. Ainsi, le domaine doit présenter les caractéristiques suivantes :

- Le niveau de réalisme du cas d'étude est important. Il doit permettre une validation qualitative et quantitative des concepts liés aux ontologies adaptatives et au processus d'adaptation qui leur est associé.
- Les connaissances du domaine doivent pouvoir être facilement identifiées pour permettre la construction de l'ontologie adaptative qui lui est associée. Les concepts du domaine tout comme les relations qui les relient doivent bien évidemment être caractéristiques du domaine.
- Les utilisateurs du domaine doivent avoir des caractéristiques distinctes en vertu de l'ontologie adaptative qui leur est associée. Ces caractéristiques doivent respecter les éléments avancés à la section 4.3.2 pour des raisons d'exploitation. Les informations des profils des utilisateurs comportant trop de similitudes ne sera pas suffisamment discriminante au moment de l'exploitation de ces données lors de la phase d'enrichissement des requêtes ASK pour le Web.
- Les documents du domaine sélectionné doivent préférablement être homogènes du point de vue de leur structure afin de pouvoir d'une part construire l'ontologie adaptative des documents du domaine (voir section 4.2.2) et d'autre part exploiter cette ontologie dans le cadre de l'enrichissement sémantique du contenu des pages Web au travers de la construction des WPGraphs et W^3Graphs (voir chapitre 4).

- Les aptitudes de ce domaine à évoluer est une des caractéristiques fondamentales que ce dernier doit présenter. Celle-ci doit être régulière et à intervalles de temps raisonnables afin que les modifications des connaissances du domaine soient observables. Par ailleurs, le domaine sélectionné doit être suffisamment mature et rassembler un nombre significatif d'utilisateurs afin que l'évolution des connaissances de ce domaine se fasse de manière consensuelle.

La validation des règles d'enrichissement des requêtes et la qualité de la recherche documentaire doivent pouvoir être mesurés. Par conséquent, le cadre expérimental doit fournir en plus des caractéristiques pour les ontologies adaptatives d'autres caractéristiques parmi lesquelles :

- La taille du domaine et principalement le nombre de documents le composant doit être bien définie. Si ce domaine est trop volumineux, il sera difficile de déterminer avec précision le rappel et la précision des résultats de la recherche. Par conséquent, il sera difficile de montrer l'efficacité des ontologies adaptatives, du processus d'adaptation et des règles d'enrichissement des requêtes ASK pour la recherche de documents pertinents. La taille du domaine a également un impact sur le phénomène d'évolution. Il sera difficile de suivre et de détecter toutes les évolutions d'un domaine trop important alors qu'un domaine trop petit ne fera apparaître aucune évolution.
- Le contenu des documents et notamment la sémantique ne doit pas présenter d'ambiguïté toujours dans le but de mesurer la précision et le rappel des résultats en réponse à une requête.
- En vue de mesurer la qualité de l'évolution des ontologies adaptatives, le contenu des documents doit représenter l'état de la connaissance du domaine à plusieurs moments distincts dans le temps.

En vertu de tous les arguments énoncés plus haut, le cas d'étude que nous proposons pour la validation de nos concepts correspond au domaine décrit par les documents relatifs à la conférence internationale *World Wide Web* publiés sur une période de dix ans. Ce choix a été influencé par notre expérience personnelle de chercheur en informatique et aussi car ce domaine présente les caractéristiques nécessaires et suffisantes pour la validation des concepts introduits tout au long de ce mémoire.

En effet, le domaine associé au cas d'étude présente les propriétés suivantes :

- Le **niveau de réalisme** : Ce cas d'étude présente un niveau de réalisme élevé principalement de part sa taille et son aptitude à évoluer au cours du temps.
- La **taille** : Le volume du domaine est essentiellement défini par la quantité de documents relatifs à la conférence *World Wide Web*. En conséquence, les documents que nous considérons pour la description du domaine sont, d'une part, les différents *appels à communication* diffusés chaque année en prévision de l'événement et, d'autre part, les différents *articles scientifiques acceptés pour publication* chaque année. L'idée de considérer les différentes conférences WWW organisées sur une période de dix ans nous permet de rassembler 633 documents (10 appels à contribution et 623 articles scientifiques).
- Les **connaissances** du domaine sont d'une part facilement identifiables car ces dernières sont synthétisées au niveau des appels à contribution de chaque événement. D'autre part, ces appels sont le fruit d'une concertation de l'ensemble du comité de programme de la conférence. Ce groupe est constitué des personnes considérées comme les plus influentes et compétentes du domaine, les connaissances sont donc **significatives** du domaine et font l'objet d'un **consensus**.
- Les connaissances du domaine sont **évolutives**. Le dynamisme affectant les connaissances

du domaine découle principalement de celui de la communauté scientifique fédérée autour de la conférence WWW et du Web en général. De plus, le rythme de l'évolution est bien définie car la conférence a lieu tout les ans à la même période de l'année ce qui permet de mettre en place de manière plus souple le processus d'adaptation des ontologies et de surveiller et comprendre les modifications des connaissances du domaine de façon plus rigoureuse.
- Les **documents du domaine sont homogènes au niveau de leur structure** et **la sémantique de leur contenu est bien définie**. Les grande majorité des documents du domaine sont des articles scientifiques. Par conséquent, leur structure est complètement homogène et respecte un plan bien précis (titre, auteurs, affiliations, sections, etc). De plus la sémantique de leur contenu est non ambiguë car la réputation de la conférence implique une sélection extrêmement rigoureuse des articles ainsi un article ambigu est systématiquement rejeté.
- Notre connaissance du domaine nous permet d'identifier facilement les **différentes catégories d'utilisateurs** et d'en construire leur profil. Deux catégories d'utilisateurs s'opposent. D'un coté, la catégorie des utilisateurs dont le profil représente les vues des chercheurs fondamentaux sur le domaine. Cette catégorie regroupe principalement les universitaires. De l'autre, celle des chercheurs appliqués regroupant majoritairement des industriels. On peut observer que les appels à communication s'adressent principalement à ces deux types de personnes (voir les en-têtes de ces documents).
- Les documents du domaine sont **accessibles via le Web**. Bien que les articles acceptés pour publication de chaque conférence sont regroupés dans les actes de la conférence (en version papier), les papiers acceptés à la conférence *World Wide Web* sont quant à eux également accessibles sur Internet (en version électronique). Deux types d'accès sont généralement proposés, un premier mode d'accès protégé par des mécanismes d'authentification et un autre complètement libre à travers le site Web de la conférence.

Les arguments avancés dans cette section justifient notre choix du domaine de la conférence internationale *World Wide Web* comme cas d'étude pour la validation expérimentale de la qualité de l'ensemble des concepts présentés dans ces travaux.

7.2.2 Construction du cadre expérimental

Le cadre expérimental défini à la section précédente sous-entend la construction et l'utilisation d'un ensemble d'éléments. Tout d'abord, la constitution d'un corpus de documents sur lequel seront axés à la fois le processus de construction et d'évolution des ontologies adaptatives ainsi que la validation de la qualité de la recherche documentaire. D'autre part, des ontologies adaptatives du domaine, des documents du domaine et des vues des utilisateurs sur le domaine sont nécessaires afin de démontrer leur efficacité dans la recherche documentaire. Dans cette section, nous allons discuter et illustrer la construction du cadre expérimental de notre étude.

Le corpus de documents

Avant de détailler la construction des ontologies adaptatives, nous avons besoin de constituer un corpus de documents en vertu de la méthodologie des ontologies différentielles de Bachimont à la base de la construction des ontologies adaptatives. Nous avons énoncé, à la section 3.3.1, les caractéristiques indispensables dont un bon corpus doit faire preuve. Par conséquent, notre corpus est composé des différents articles scientifiques acceptés pour publication à toutes les

conférences WWW depuis 1998 ainsi que des appels à communication pour chaque événement depuis cette même date.

Année de la conférence WWW	**Nombre de documents**
1998	55
1999	49
2000	56
2001	78
2002	71
2003	77
2004	77
2005	78
2006	84
Total	633

TABLE 7.1: Composition du corpus

Nous disposons donc d'un corpus de 633 documents sur lequel nous pouvons nous appuyer pour construire les ontologies adaptatives nécessaires et pour réaliser un ensemble d'expérimentation pour valider l'approche proposée. Les documents sont tous écrits dans la même langue (l'anglais), et le contenu est homogène du point de vue de la sémantique car ce sont tous des articles scientifiques.

L'ensemble de ces documents est regroupé dans une base de donnée relationnelle dont le schéma est représenté sur la figure 7.1. L'utilisation d'une base de données permet de stocker l'ensemble des documents, de structurer davantage le corpus mais également de faciliter les accès aux documents qu'elle contient en vue d'une future exploitation.

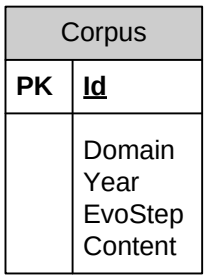


FIGURE 7.1 – Schéma de la base de données

Comme le montre la figure 7.1, la base de données est très simple. Les champs qui composent l'unique table se détaillent de la manière suivante :

- **Id :** La clef primaire permettant d'identifier chaque enregistrement. Par conséquent, à chaque document correspond un unique Id.
- **Domain :** Ce champ contient le nom du domaine auquel appartient le document correspondant. Dans notre cas, *domain* contient la valeur WWW.
- **Year :** Ce champ contient un entier correspondant à l'année où le document est apparu dans le domaine. Dans notre cas, la date en question correspond l'année de la conférence où l'article a été publié.

- **EvoStep :** Ce champ contient un entier naturel dont la valeur représente le nombre de pas d'évolution ayant eu lieu jusqu'à l'apparition du document dans le domaine. La valeur de ce champ est utilisée dans le processus d'adaptation des ontologies.
- **Content :** Ce dernier champ contient le contenu de chaque document. Dans notre cas, le contenu d'un article ou d'un appel à communication.

Le corpus ainsi constitué va nous permettre de construire les différentes ontologies adaptatives nécessaires au bon déroulement de l'approche TARGET.

Les ontologies adaptatives

La méthodologie de construction des ontologies adaptatives est décrite au chapitre 4. Cette méthodologie est basée sur celle des ontologies différentielles de Bachimont [Bouaud et al., 1994]. Les ontologies adaptatives utilisées pour la validation expérimentale de nos concepts sont au nombre de trois. Nous illustrons tout d'abord la construction de l'ontologie du domaine, puis celle des documents du domaine pour finir celles représentant les vues des différentes catégories d'utilisateurs sur le domaine. La présentation du processus de construction respecte les différentes étapes de la méthode des ontologies différentielles.

L'ontologie adaptative du domaine

L'ontologie adaptative du domaine représente les connaissances du domaine définies par la conférence internationale *World Wide Web*. L'ontologie de départ est construite suivant la méthodologie décrite à la section 4.2.3 fondée sur les idées de Bachimont et sur l'état de la connaissance du domaine en 1998. Le processus de construction que nous avons suivi s'organise suivant quatre étapes :

- La **construction d'un corpus de documents** et son analyse. Cette étape a été discutée au paragraphe précédent. Cependant, pour construire cette ontologie adaptative nous allons privilégier l'utilisation des appels à communication pour déterminer l'ensemble des concepts de l'ontologie et le corps des articles scientifiques pour définir les relations entre ces concepts. Lors de l'analyse du corpus, nous avons retenu les «topics» proposés dans les appels à communication comme candidats pour la définition des concepts de l'ontologie. Ces termes sont définis en résultat d'une concertation entre les membres du comité de programme et reflètent bien l'état de la connaissance du domaine à un moment précis du temps. Par conséquent, chaque topic donne lieu à un concept dans l'ontologie. L'analyse sémantique du contenu des papiers, et principalement de la partie *résumé* permet de définir les relations entre les concepts de l'ontologie. Pour cela, nous avons procédé à une analyse manuelle du corpus afin de relier les concepts entre eux.
- La **normalisation sémantique** au cours de laquelle les concepts retenus lors de l'étape précédente sont désambiguïsés. Dans notre cas, ce processus est rapide car les candidats retenus sont, par définition de l'appel à communication, non ambigus. Au cours de cette étape, nous définissons également la valeur des éléments propres au modèle des ontologies adaptatives (poids sémantique, durée de validation et date d'émergence) toujours suivant l'analyse du corpus et les métriques décrites au chapitre 3.
- **L'engagement ontologique** permet de construire le treillis de l'ontologie (i.e. la hiérarchie de concepts). Dans notre cas, nous utilisons le résultat de l'analyse du corpus pour définir les relations entre les concepts y compris les relations de subsumption, éléments de base du treillis. Lors de la définition des relations entre les concepts, nous déterminons également

les valeurs associées à la distance sémantique et à la résistance aux changements propres à chaque relation ontologique.
- **L'opérationalisation** dans un langage de représentation des connaissances. Dans notre cas, nous utilisons OWL. Le code OWL de l'ontologie résultante est celui de l'annexe B.

Cette méthodologie nous permet d'obtenir une ontologie adaptative qui va servir de base au processus d'adaptation décrit au chapitre 3. La construction incrémentale du corpus (année après année) va permettre de distinguer les documents par rapport à leur date d'émergence dans le domaine ce qui permet une application des métriques introduites précédemment.

L'ontologie adaptative des documents du domaine

Les documents du domaine respectent également une structure particulière et bien définie exploitable dans notre approche pour la sélection des données lors de la construction des WP-Graphs et W^3Graphs représentant le contenu annoté d'une page Web. L'ontologie des documents du domaine est construite suivant la structure des documents de telle façon à faire ressortir l'importance du contenu de chaque partie composant les documents. L'ontologie que nous avons construite est celle présentée à la section 4.2.2.

Les catégories d'utilisateurs

Enfin, nous construisons les ontologies adaptatives représentant les vues des différentes catégories d'utilisateurs sur le domaine. Notre expérience personnelle du monde de la recherche scientifique auquel appartient celui de la conférence WWW, nous permet de distinguer deux types d'utilisateurs, les *universitaires* et les *industriels*. Cette distinction est renforcée par les informations contenues dans les appels à communication faisant explicitement référence à ces deux catégories de personnes. Les deux ontologies adaptatives que nous utilisons dans ce cas d'étude sont celles détaillées à la section 4.3.2 représentant les caractéristiques des chercheurs fondamentaux (i.e. des universitaires) et des chercheurs appliqués (i.e. des industriels) (voir annexe C).

Pour résumé, notre cas d'étude porte sur le domaine défini par la conférence internationale World Wide Web. Il est composé d'un corpus contenant 633 documents essentiellement d'articles scientifiques et d'appels à communication pour la conférence WWW, et de quatre ontologies adaptatives représentant les connaissances du domaine, les documents du domaine ainsi que deux catégories d'utilisateurs : les chercheurs fondamentaux et appliqués. Ce cadre expérimental va nous permettre de mettre en place un ensemble d'expérimentations pour la validation de notre approche. Cependant, dans le but d'automatiser le déroulement des expérimentations nous avons développé un outil implémentant l'approche.

7.3 L'outil TARGET pour la gestion des ontologies adaptatives et la recherche d'information pertinente sur le Web

L'approche TARGET présentée dans ces travaux intègre à la fois l'évolution des ontologies et leur utilisation pour améliorer la recherche documentaire sur le Web du point de vue de la

pertinence des résultats. Le développement d'un outil doit donc tenir compte de ces deux aspects. L'étude que nous avons menée sur les outils pour la gestion de l'évolution des ontologies a montré des manques importants, c'est pourquoi la proposition d'un outil pour la gestion de l'adaptation des ontologies contribue à combler quelque peu ces lacunes. Dans cette section, nous discutons du développement de l'outil implémentant l'approche TARGET. Cette section est plus technique que celles présentées jusque la dans la mesure où nous détaillons les fonctionnalités offertes par l'outil ainsi que les différents choix d'implémentation.

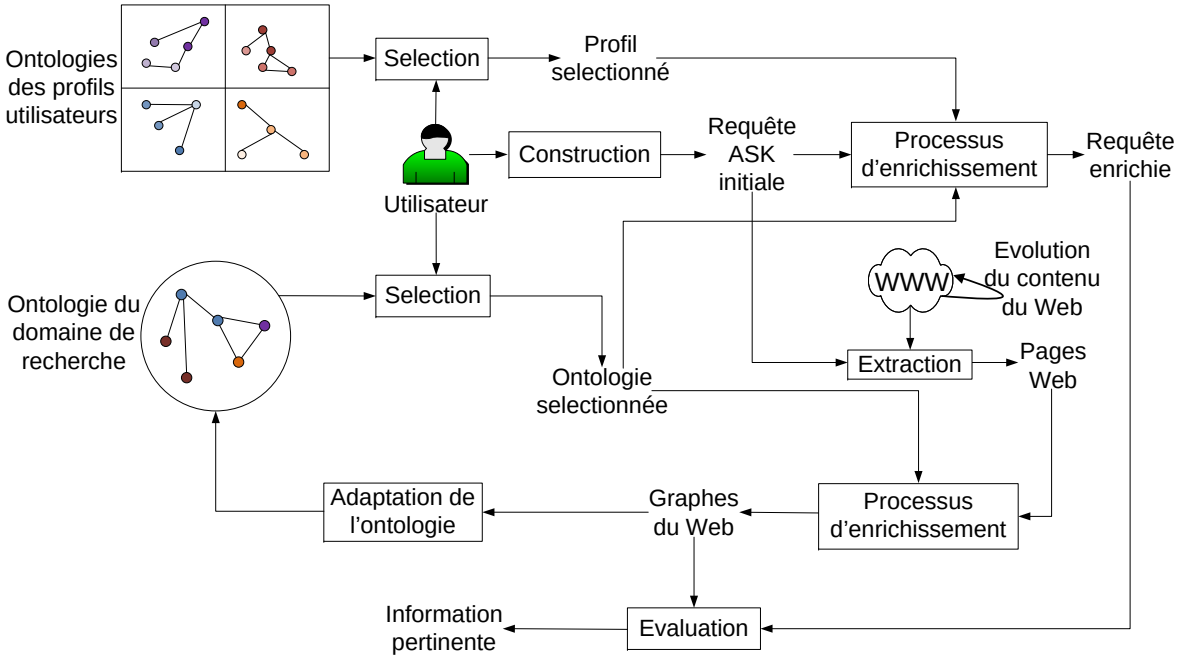


FIGURE 7.2 – Schéma fonctionnel de l'approche

L'approche TARGET figurant sur la figure 7.2 suit sept étapes dans lesquelles l'utilisateur est assisté au maximum afin d'obtenir les réponses les plus pertinentes à ses requêtes. Les sept étapes de l'approche se déclinent comme suit :

1. L'utilisateur saisit une requête ASK initiale en prenant soin de bien préciser le domaine visé par la requête en utilisant l'opérateur « :».
2. Il sélectionne ensuite l'ontologie représentant le mieux les vues qu'il a sur le domaine (i.e. la catégorie d'utilisateurs le caractérisant le mieux).
3. A partir de la requête saisie par l'utilisateur, le système interroge un moteur de recherche usuel pour extraire un ensemble de pages Web. Dans notre cas nous utilisons l'application Google.
4. En parallèle, le système procède à l'enrichissement de la requête initiale grâce aux règles d'enrichissements des requêtes et aux deux ontologies (domaine et utilisateur) sélectionnées par l'utilisateur.
5. L'ensemble des pages Web retournées par Google est transformé en WPGraphs et W^3Graphs grâce à l'utilisation des ontologies du domaine et des documents du domaine.
6. La requête enrichie est vérifiée suivant les algorithmes présentés au chapitre 6 sur la structure logique des WPGraphs et W^3Graphs pour en extraire l'information la plus pertinente.

La liste des URL des pages Web pertinentes est classée puis retournée à l'utilisateur afin d'être consultée.

7. Les pages pertinentes sont stockées dans une base de données puis analysées par le système. Le résultat de cette analyse est réutilisé pour adapter les connaissances de l'ontologie du domaine à celles du monde réel décrites par ces documents.

Pour être cohérent avec le processus régissant l'approche TARGET décrit ci-dessus, l'outil l'implémentant doit répondre à plusieurs objectifs et par conséquent offrir plusieurs fonctionnalités. L'outil doit tout d'abord permettre à l'utilisateur de gérer les ontologies adaptatives. Pour cette raison, le système doit offrir :

- La possibilité de modifier manuellement la valeur des éléments caractéristiques des ontologies adaptatives (i.e. la date d'émergence, le poids sémantique, la date de validité, la distance sémantique et la résistance aux changements).
- La possibilité de construire dynamiquement un corpus de documents, de l'analyser et d'adapter les connaissances représentées dans l'ontologie. Ceci implique :
  - Le moyen d'intégrer les nouveaux concepts, détectés lors de l'analyse du corpus, à l'ontologie.
  - La modification des valeurs des coefficients en fonction des résultats de l'analyse du corpus.
  - La suppression éventuelle de certain éléments ontologiques.
- La garantie de la cohérence (au sens complétude du raisonnement) de l'ontologie adaptative après adaptation.

Remarque: La construction d'une ontologie adaptative est faite à l'aide d'un éditeur d'ontologie existant pouvant gérer le langage OWL. Dans nos travaux, nous avons utilisé Protégé.

Par ailleurs, l'outil doit permettre d'effectuer une recherche documentaire sur le Web. Les fonctionnalités du système répondant à cet objectif sont alors :

- La possibilité pour les utilisateurs de saisir les requêtes ASK.
- La possibilité pour les utilisateurs de sélectionner les ontologies adaptatives du domaine et de la catégorie des utilisateurs à laquelle ils appartiennent.
- Un interfaçage avec un moteur de recherche usuel donnant un accès aux documents du Web.
- Le moyen de construire les WPgraphs et W^3Graphs associés aux pages Web retournées par le moteur de recherche usuel suivant l'ontologie du domaine sélectionnée, d'enrichir la requête initiale et de l'évaluer sur la structure logique des graphes.
- La possibilité pour les utilisateurs de consulter les résultats de la recherche.

La description des différentes fonctionnalités offertes par notre outil nous permet de détailler à présent l'architecture logicielle puis les choix d'implémentation.

7.3.1 Architecture de l'outil

L'architecture générale de l'outil TARGET respecte le schéma de la figure 7.3 ci-dessous. Cette architecture se décompose en deux parties :

1. Une partie pour la gestion des ontologies adaptatives.
2. Une partie pour la recherche documentaire sur le Web.

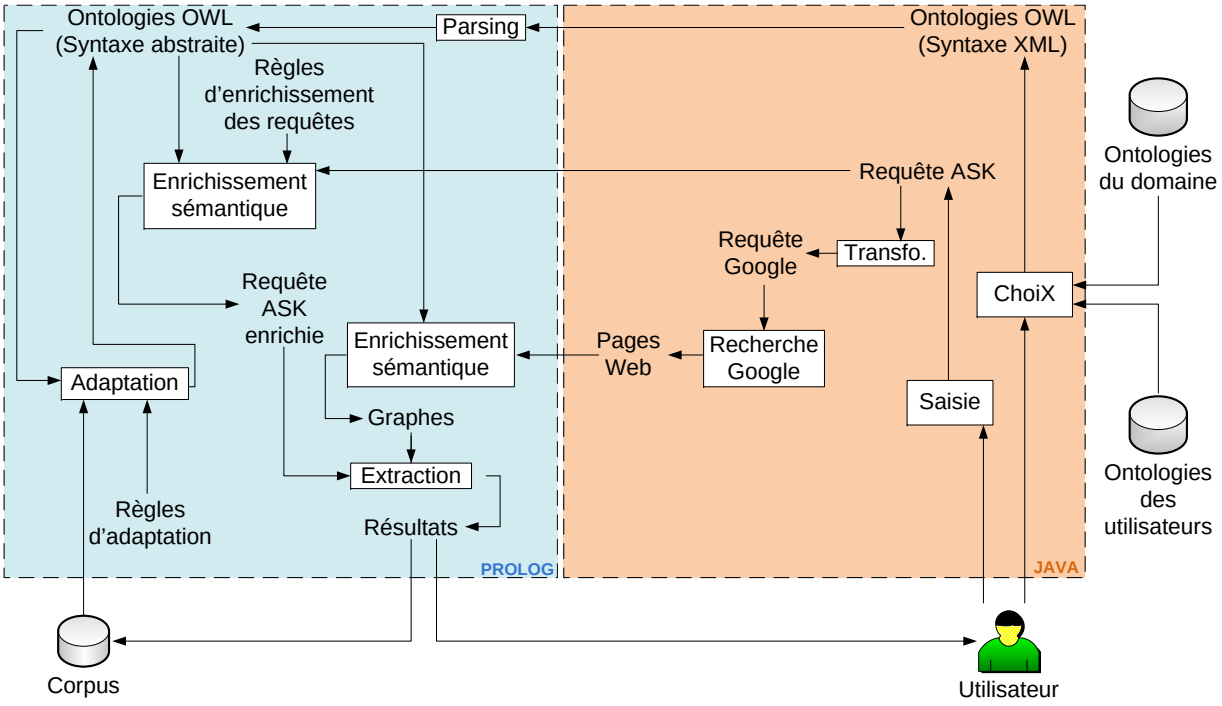


FIGURE 7.3 – Schéma fonctionnel de l'outil

Dans les sous-sections suivantes, nous allons discuter plus en détail de l'architecture logicielle de ces deux parties ce qui nous permettra de justifier par la suite nos choix d'implémentation de l'outil.

Module pour la gestion des ontologies adaptatives

Le module pour la gestion des ontologies adaptatives doit répondre aux objectifs énoncés à la section 7.3. La vue logique de l'architecture de ce composant logiciel est représentée figure 7.4 ci-après.

Le diagramme de la figure 7.4 montre les différents *packages* composant le module pour la gestion des ontologies adaptatives. Parmi les 5 packages formant le module, un est spécifique à la gestion des ontologies adaptatives et quatre sont communs à ce module et à celui consacré à la recherche documentaire sur le Web.

Le package *Evo* est spécifique au module pour la gestion des ontologies adaptatives. Il fournit les éléments nécessaires à l'interaction avec les packages *corpus*, *PROLOG* et *GUI*. Il contient notamment les éléments logiciels concernant les règles d'adaptation des ontologies et leur application.

Les packages *corpus*, *GUI*, *PROLOG* et *Proto* sont communs à tous les modules de l'outil. Ils s'articulent et ont les rôles suivants :

- Le package *corpus* permet la gestion du corpus de documents nécessaires au bon fonctionnement de l'approche et principalement l'ajout de documents au corpus et leur analyse (i.e. détection des nouveaux concepts et application des métriques introduites au chapitre 3). Il offre également les fonctionnalités pour la gestion de la base de données servant à stocker les documents du corpus.
- Le package *GUI* gère toutes les fonctionnalités relatives à l'interface graphique de l'outil.

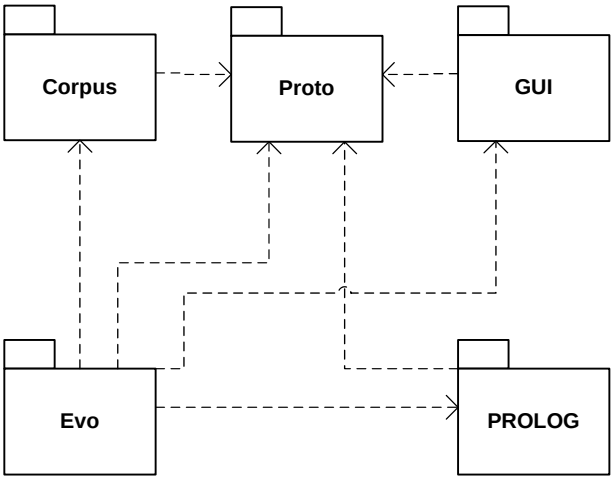


FIGURE 7.4 – Vue logique de la partie pour la gestion des ontologies adaptatives

Ces fonctionnalités sont celles qui permettent à l'utilisateur d'interagir avec l'outil comme la saisie de la requête, la sélection des ontologies, la saisie des éléments des ontologies adaptatives à modifier et la visualisation des résultats d'une recherche documentaire sur le Web.

- Le package *PROLOG* permet l'application des règles d'enrichissement des requêtes ASK et celles pour l'adaptation des ontologies. Il gère aussi les fonctionnalités pour garantir la bonne évolution de l'ontologie du domaine et des catégories d'utilisateurs.
- Le package *Proto* fournit des éléments de base au bon déroulement de l'approche. Nous entendons par la le chargement des ontologies, le parsing d'ontologie, etc. Comme le montre les figures 7.4 et 7.5 ce package permet de gérer les dépendances entre les différents modules de l'outil.

Les différents package que nous venons de présenter constituent le module logiciel pour la gestion des ontologies adaptatives. Nous discutons à présent des caractéristiques de l'autre module de l'outil.

Module pour la recherche d'information sur le Web

Le module pour la recherche documentaire sur le Web est formé de huit packages (voir figure 7.5), quatre d'entre eux ont été présentés à la section précédente et les quatre autres sont les suivants :

- Le package *Google* : il permet à l'outil de s'interfacer avec le moteur de recherche Google à travers l'API offerte par ce dernier.
- Le package *Query* : il permet la gestion des requêtes ASK. Par conséquent, la vérification syntaxique, l'enrichissement ainsi que la transformation de la requête se font à travers les classes de ce package.
- Le package *Webgraphs* : il contient les classes et méthodes nécessaires à la transformation du code HTML d'une page Web en WPGraph et par la suite leur regroupement dans un W^3Graph.
- Le package *Websearch* : il met en œuvre tous les éléments impliqués dans la recherche documentaire y compris le classement des résultats et l'interaction avec le package *GUI*

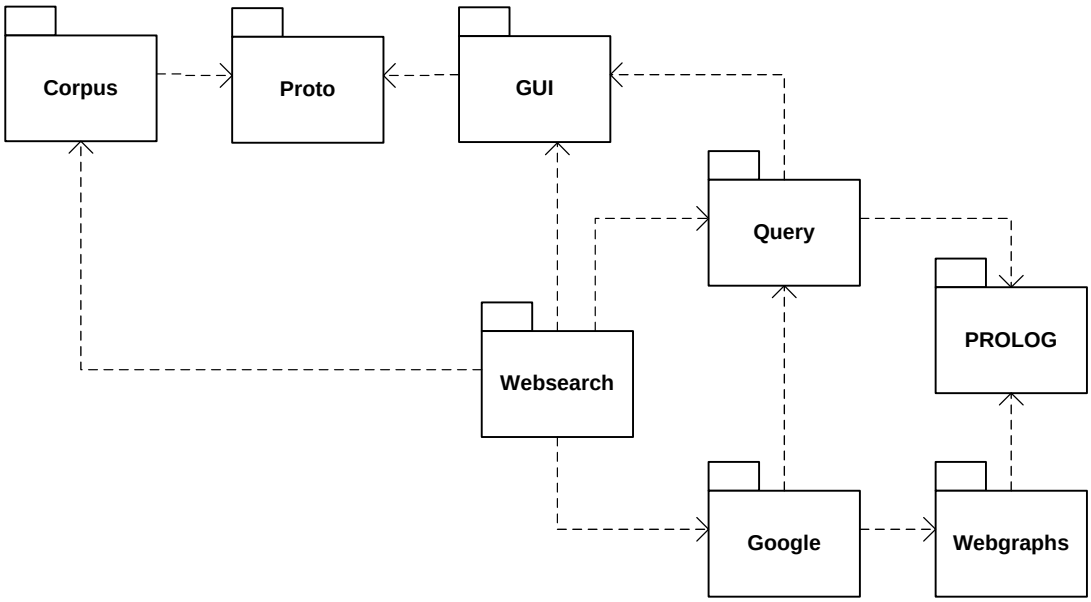


FIGURE 7.5 – Vue logique de la partie pour la recherche d'information sur le Web

pour l'affichage des résultats.

L'architecture de l'outil TARGET que nous proposons se prête à l'utilisation de diverses technologies pour le développement du logiciel correspondant. Nous allons discuter de ces choix dans la section suivante.

7.3.2 Implémentation

Les choix des technologies qui ont permis le développement de l'outil TARGET ont été influencés par plusieurs facteurs parmi lesquels :

- Le **temps de développement** doit être le plus court possible. L'objectif pour lequel nous avons développé l'outil visait à améliorer la qualité et la quantité des expérimentations permettant de valider les concepts introduits tout au long de ce mémoire. Pour cette raison nous avons décidé de réduire autant que possible le temps de développement en favorisant au maximum l'utilisation d'outils existants.
- La possibilité de **combiner différents composants** logiciels. L'architecture que nous proposons nécessite l'utilisation de composants tels que des bases de données. Il faut donc que les technologies sélectionnées puissent intégrer tous ces composants.
- La **réutilisation** et **l'extension** de l'outil. Les technologies doivent permettre d'étendre et de réutiliser le code facilement.
- Le **résultat final** a également son importance. L'outil TARGET doit pouvoir être réutilisé facilement par des utilisateurs ayant quelques familiarités avec l'utilisation des outils conventionnels pour la recherche d'information.

Ces différents points ont favorisé l'utilisation des technologies Java [1] et XSB PROLOG [2] pour le développement de l'outil TARGET.

Les technologies Java permettent de répondre à la plupart de ces objectifs. Tout d'abord, ces technologies sont suffisamment matures et bien outillées (comme par exemple l'environnement de développement intégré Eclipse) pour offrir des moyens de développement riches pouvant réduire le temps de développement d'un logiciel.

Ensuite, l'engouement suscité par Java a permis de fédérer une grosse communauté de chercheurs, d'ingénieurs ou plus généralement de développeurs pour la création de bibliothèques et de composants réutilisables pour faciliter le développement de logiciel Java. Par conséquent, nous avons pu en bénéficier notamment à travers l'utilisation de bases de données MySQL [3] (pour la construction du corpus de documents), de l'API Java développée par Google (pour interroger le moteur de recherche) ainsi que celles permettant d'accéder directement à des documents du Web et de traiter facilement du code HTML (facilitant ainsi la construction des graphes) et enfin le générateur de parser CUP [4] pour le traitement efficace du langage de requêtes ASK (parsing des requêtes et transformation en questions PROLOG).

De plus, Java permet le développement rapide de composants graphiques, aspect non négligeable pour la construction d'interface adaptée à la saisie des requêtes et à l'affichage des résultats d'une recherche. Un dernier point important, le paradigme orienté objet sur lequel s'appuie Java permet la réutilisation et l'extension efficace des objets conçus.

Nous avons également pu tirer parti des avantages offerts par le langage PROLOG et notamment des fonctionnalités de la distribution XSB. La philosophie de PROLOG (son fonctionnement à partir d'une base de faits) se marie parfaitement avec les différentes règles (d'enrichissement des requêtes ASK et d'adaptation des ontologies) que nous utilisons dans notre approche facilitant ainsi la gestion des ontologies adaptatives et leur exploitation à travers le processus d'enrichissement des requêtes.

De plus, les fondations logiques de PROLOG facilitent la construction des WPGraphs, W^3Graphs et des requêtes ASK (reposant tous les trois sur un formalisme logique) puis automatisent complètement la vérification de ces dernières sur les graphes grâce à des algorithmes d'unification implémentés par les différentes distributions de PROLOG.

Un point primordial qui a également motivé l'utilisation de PROLOG est l'existence d'une bibliothèque (THEA [5]) de buts PROLOG pour le parsing et l'utilisation d'ontologies OWL (Lite, DL ou Full). Cet ensemble de buts permet de transformer une ontologie OWL, décrite dans une syntaxe à la XML, dans un ensemble de faits PROLOG dont la syntaxe est proche de la syntaxe abstraite de OWL. Certains buts permettent également de raisonner sur les ontologies OWL et notamment de tester leur cohérence si ces dernières ont été modifiées.

Enfin, l'utilisation de l'API Java Interprolog [6] a permis de lier les composants Java et ceux de PROLOG afin de combiner les avantages des deux technologies. Ceci est nécessaire principalement pour l'enrichissement des requêtes et leur vérification sur les graphes comme le montre la figure 7.3.

1. http ://java.sun.com
2. http ://xsb.sourceforge.net/
3. www.mysql.com
4. http ://www.cs.princeton.edu/∼appel/modern/java/CUP/
5. www.semanticweb.gr/TheaOWLLib
6. http ://www.declarativa.com/interprolog/

L'outil TARGET présente, dans sa conception, deux parties distinctes : un *front end* Java afin de permettre aux utilisateurs d'interagir avec l'outil et un moteur PROLOG pour la gestion et l'exploitation des ontologies adaptatives en fonction des données saisies par l'utilisateur. Jusqu'à présent, nous avons présenté certains des éléments utiles pour la validation expérimentale de nos concepts à savoir le cas d'étude ainsi que l'outil TARGET nécessaire à l'automatisation de l'approche. Dans les sections suivantes, nous allons nous concentrer sur le détail des expérimentations mises en place pour la validation des concepts.

7.4 Validation expérimentale de l'approche

Dans cette section, nous discutons des expérimentations permettant la validation des concepts mises en œuvre dans le cadre de l'approche TARGET. Dans ce but, nous introduisons en premier lieu les objectifs de validation que nous nous sommes fixés. La seconde partie de la section est consacrée aux expérimentations. Par conséquent, nous présentons le protocole expérimental ainsi que les différents scénarios mis en place pour atteindre les objectifs définis. Pour finir, nous détaillons et commentons l'ensemble des résultats expérimentaux obtenus grâce à l'outil TARGET introduit à la section précédente.

7.4.1 Objectifs

L'objectif principal pour lequel nous avons défini un tel cadre expérimental consiste en la démonstration de l'utilité et du bien-fondé de nos concepts et principalement du modèle des ontologies adaptatives et de leur utilité dans la recherche d'information sur le Web dans l'amélioration de la pertinence des résultats. Pour atteindre cet objectif principal, nous proposons de réduire la complexité de la tâche en définissant quatre sous-objectifs. Ces quatre objectifs secondaires se définissent de la façon suivante :

- La validation du modèle des ontologies adaptatives. L'objectif est de démontrer la validité des concepts dénotant l'émergence, la période de validité, le poids sémantique, la distance sémantique et la résistance aux changements associés aux différents éléments d'une ontologie.
- La démonstration de l'apport des ontologies adaptatives dans le processus de recherche d'information. L'objectif est de montrer l'intérêt des éléments du modèle des ontologies adaptatives introduit au chapitre 2 dans la recherche d'information. Ces éléments montreront leur efficacité à la fois dans le processus d'adaptation ainsi que dans l'utilisation des ontologies pour l'enrichissement des requêtes ASK.
- La démonstration de la qualité du processus d'évolution que nous avons défini au chapitre 3. Dans l'approche TARGET, les résultats sont conditionnés par l'utilisation des ontologies adaptatives principalement dans la phase d'enrichissement des requêtes. Par conséquent, la qualité des résultats d'une recherche est fonction de la qualité de l'évolution de l'ontologie du domaine utilisée. L'ontologie ayant mal évolué aura un impact négatif sur la qualité de la recherche sur le Web. C'est pourquoi des résultats précis et un rappel élevé attesteront de la qualité du processus d'adaptation appliqué à l'ontologie du domaine ayant servi pour l'enrichissement des requêtes.
- la démonstration de l'apport de l'ontologie du domaine et de l'ontologie représentant les vues de l'utilisateur sur le domaine lors de la recherche documentaire. L'objectif est de montrer que ces deux ontologies jouent un rôle important dans la qualité des résultats retournés et notamment leur utilisation pour restreindre la taille de l'espace des documents au moment de l'interprétation des requêtes.

– La démonstration de la qualité des règles d'enrichissement des requêtes ASK. Cet aspect a un impact important sur la pertinence des pages Web retournées il nous faut alors montrer que les règles que nous proposons ont un impact positif sur la pertinence des résultats de la recherche. La validation des règles d'enrichissement sous-entend la qualité de la forme de la requête enrichie (i.e. la façon dont les mots-clés additionnels sont combinés à ceux initialement saisies par l'utilisateur) et le choix des relations ontologiques pour le choix des mots-clés pour l'expansion.

Si tous ces objectifs sont atteints, nous pourrons dire que nos concepts permettent, dans le cadre expérimental que nous avons défini, l'amélioration de la recherche documentaire du point de vue de la pertinence des résultats.

7.4.2 Protocole expérimental

Une validation expérimentale doit être rigoureuse de par la place prépondérante qu'elle occupe dans des travaux de recherche. Par conséquent, le protocole expérimental que nous avons défini doit respecter cette rigueur. En plus d'être rigoureuse, la validation doit être crédible. Dans notre cas, cela sous-entend que les expérimentations comprennent un nombre significatif de requêtes. Pour cette raison, nous avons décidé de définir un ensemble d'une centaine de requêtes ASK différentes pour nos expérimentations.

Ce nombre conséquent de requêtes va permettre d'utiliser tous les constructeurs du langage ASK. L'objectif vise à mettre en œuvre l'ensemble des règles d'enrichissement des requêtes. Pour cette raison différents constructeurs du langage doivent être utilisés ainsi que différentes ontologies ou différents mots-clés afin de solliciter toutes les relations ontologiques pour l'enrichissement des requêtes.

Le mode opératoire consiste donc à poser successivement une centaine de requêtes différentes en utilisant l'ensemble des constructeurs du langage et différentes ontologies pour l'expansion des requêtes en fonction du scénario (voir ci-dessous) puis à mesurer manuellement la précision et le rappel des résultats retournés par l'outil TARGET. Le fait de travailler dans un environnement restreint (un corpus de 633 documents) nous permet de procéder manuellement pour de telles mesures.

7.4.3 Scénarios expérimentaux pour la validation de l'approche

Les objectifs de validation définis à la section 7.4.1 ainsi que le protocole expérimental discuté précédemment sous-entend la mise en place de scénario. Les concepts à valider sont nombreux c'est pourquoi il est difficile de faire ressortir la contribution de chacun d'entre eux au cours d'une seule expérience. Pour cette raison, nous avons déterminé un ensemble de cinq scénarios différents afin de mettre en évidence tour à tour l'apport des concepts dans le cadre expérimental que nous nous sommes fixé (i.e. la recherche d'articles scientifiques publiés à la conférence World Wide Web). Dans les sous-sections à venir, nous allons présenter plus précisément cet ensemble de cinq scénarios.

Scénario 1 : Le cas de base

Le premier scénario que nous proposons représente le cas de base auquel vont être comparés les résultats obtenus au cours de nos expérimentations. Les mesures de précision et de rappel des

résultats obtenus suivant ce scénario vont permettre d'étalonner les résultats obtenus en suivant les autres scénarios (voir ci-dessous).

Au cours de ce scénario, les requêtes ASK saisies par l'utilisateur sont interprétées directement sans avoir subi un quelconque enrichissement. Ce type d'expériences correspond au cas où les utilisateurs du Web sont confrontés à chaque fois qu'ils effectuent une recherche via un moteur de recherche classique du type Google ou Yahoo. Les résultats obtenus suivant ce scénario consistent en un bon étalon et serviront de valeur de référence pour démontrer l'efficacité de nos concepts dans l'amélioration de la pertinence des résultats au cours d'une recherche documentaire sur le Web.

Remarque: Le fait de restreindre la recherche à notre corpus de documents représente un premier filtre important. Comme les moteurs de recherche usuels du type Google sont ouverts sur l'ensemble du Web, donc sur des domaines divers et variés, les résultats sont forcément plus diversifiés et moins précis. En dépit de cette restriction, nous considérons tout de même ce scénario comme une bonne référence pour la suite de la validation.

Scénario 2 : Pour la validation des règles d'enrichissement des requêtes ASK

L'objectif de cette deuxième série d'expériences concerne la validation des règles d'enrichissement des requêtes ASK contenues dans le tableau 5.4 de la section 5.2. Le processus d'enrichissement nécessite l'utilisation de l'ontologie du domaine et de l'ontologie représentant les vues de l'utilisateur sur le domaine.

Cependant, l'utilisation de cette dernière ontologie représente une des valeurs ajoutées de notre approche. Pour démontrer sa contribution dans la recherche documentaire nous avons décidé de lui consacrer un scénario à part entière (voir ci-dessous). Par conséquent, la validation du processus d'enrichissement des requêtes ASK est faite par l'utilisation de la seule ontologie du domaine.

De plus, pour faire ressortir davantage la contribution des règles d'enrichissement, l'ontologie que nous utilisons est une ontologie OWL conventionnelle et non une ontologie adaptative c'est-à-dire une ontologie ne tenant pas compte notamment du poids sémantique, de la date d'émergence et de la distance sémantique mis en œuvre dans les règles d'enrichissement . La contribution apportée par ce type d'ontologies est l'objet du scénario suivant. De plus, l'ontologie OWL utilisée est l'ontologie représentant le domaine en 1998. Elle n'a donc subit aucune évolution.

Remarque: Les expériences effectuées dans le cadre de ce scénario ont nécessité l'adaptation de l'outil TARGET pour le processus d'enrichissement. D'autres types d'adaptation de l'outil sont également nécessaires pour le bon déroulement des autres scénarios notamment sur l'utilisation des ontologies.

Scénario 3 : Pour la validation du modèle des ontologies adaptatives

Le troisième scénario mis en place a pour objectif principal la validation expérimentale du modèle des ontologies adaptatives et de l'apport des éléments de ce modèle pour la recherche d'information sur le Web. Leur utilisation pour la recherche d'information se fait principalement à travers les règles d'enrichissement des requêtes ASK et notamment l'exploitation de la distance sémantique, du poids sémantique et de la date d'émergence. C'est pourquoi nous utilisons l'ontologie adaptative du domaine dans le processus d'enrichissement. Par conséquent, ce scénario

se différencie du précédent sur ce seul point. L'ontologie adaptative du domaine utilisée est celle représentant le domaine en 1998 par conséquent, elle n'a subit aucune adaptation aux évolutions du domaine de la conférence World Wide Web au cours du temps.

Pour les expérimentations de ce scénario, aucune ontologie représentant les vues de l'utilisateur sur le domaine de la conférence n'est utilisée. Ce point a donc nécessité l'adaptation de l'outil TARGET afin que ce dernier ne tienne pas compte de cette ontologie pour l'enrichissement des requêtes ASK.

Le fait que seul le type de l'ontologie utilisée change par rapport au scénario précédent va nous permettre de bien mettre en évidence l'impact des ontologies adaptatives sur la pertinence des résultats retournés lors d'une recherche documentaire notamment au niveau de la précision et du rappel des résultats. Ceci va nous permettre de valider expérimentalement les éléments caractéristiques du modèle des ontologies adaptatives. La validation du modèle sera complétée par un ensemble d'expérimentations mettant en œuvre le processus d'adaptation (voir scénario 5).

Scénario 4 : Pour la validation de l'apport de l'ontologie adaptative des vues de l'utilisateur

Le quatrième scénario mis en œuvre a pour objectif la validation de la contribution de l'utilisation de l'ontologie représentant les vues de l'utilisateur sur le domaine sur la pertinence des résultats d'une recherche d'information.

Pour la réalisation de ces expérimentations, nous procédons à l'utilisation des ontologies adaptatives du domaine et des vues de l'utilisateur sur le domaine pour l'enrichissement des requêtes ASK. Rappelons que l'exploitation de ce type d'ontologie lors du processus d'enrichissement se fait principalement à travers la considérons du poids sémantique pour la sélection des termes les plus adaptés à l'expansion de la requêtes initialement soumis par l'utilisateur au système. Les ontologies représentant les vues des utilisateurs sont celles discutées dans la section précédente et décrivant le profil d'un chercheur fondamental et d'un chercheur appliqué (le lecteur peut retrouver ces deux ontologies dans la partie de ce manuscrit réservée aux annexes). On peut également observer qu'aucune adaptation de l'outil n'est nécessaire pour la réalisation de ce set d'expériences.

Les expériences mises en place dans le cadre de ce scénario diffèrent des précédentes dans le sens où l'ontologie représentant les vues de l'utilisateur sur le domaine est utilisée. La différence entre les résultats obtenus lors de la réalisation de ces expériences et des résultats obtenus précédemment est la conséquence de l'utilisation de cette ontologie particulière. Il sera alors facile d'en tirer les conclusions qui s'imposent sur la contribution du profil utilisateur dans la recherche d'information.

Scénario 5 : Pour la validation du processus d'adaptation

Le cinquième et dernier scénario que nous proposons a pour objectif la mise en évidence de la qualité du processus d'adaptation des ontologies que nous avons proposé au chapitre 3 ainsi que de la résistance aux changements et de la période de validité, éléments du modèle des ontologies adaptatives.

Pour démontrer ce point, les expérimentations que nous proposons sont un peu différentes de celles proposées jusqu'à présent car dans le cadre de ce scénario nous avons défini trois types

d'expérimentations utilisant tous une ontologie adaptative du domaine, les ontologies adaptatives représentant les vues des utilisateurs sur le domaine et le processus d'enrichissement des requêtes classique. Cependant, les ontologies utilisées ont toutes évolué de manière différente (i.e. les règles d'adaptation définies au chapitre 3 ont été modifiées de telle sorte que les coefficients associés à chaque élément soient différents).

Le mode d'adaptation (ou d'évolution) des trois ontologies adaptatives du domaine que nous avons utilisées dans le cadre des expérimentations de ce cinquième scénario se différencie sur les aspects suivants :

- Le premier mode d'adaptation que nous considérons dans le cadre de ce scénario est celui introduit au chapitre 3. L'ontologie résultante décrit alors l'état des connaissances du domaine de la conférence en 2006. Cette ontologie a été obtenue en partant de l'ontologie de 1998 à laquelle le processus d'adaptation a été appliqué (l'ontologie figure dans la partie annexe). Ce cas correspond à l'application de l'ensemble des concepts de l'approche TARGET.
- Le second mode d'évolution appliqué à l'ontologie du domaine respecte les modes d'évolution classique que l'on retrouve dans la littérature (voir section 1.2.2). Dans ce type de procédé, les évolutions s'appliquant à l'ontologie ne tiennent compte que des nouveaux concepts et pour des raisons indéfinies, se proposent de conserver tous les concepts dans l'ontologie. Par conséquent, du fait qu'aucun concept de l'ontologie n'est supprimé, la connaissance représentée dans l'ontologie n'est pas à jour et l'ontologie peut prendre des proportions imposantes (en nombre d'éléments).
- Le troisième et dernier mode d'évolution appliqué à l'ontologie adaptative du domaine fait abstraction des notions de distance sémantique et de résistance aux changements. Nous entendons par là que dans le processus d'adaptation présenté au chapitre 3, les règles 3, 4 et 5 d'une part ne tiennent pas compte de la résistance aux changements et d'autre part, la distance sémantique existant entre les concepts est définie comme constante et de même valeur quelque soit la relation ontologique considérée.

Après avoir présenté les différents scénarios selon lesquels nous proposons de valider la valeur ajoutée de l'ensemble des concepts de l'approche TARGET, nous nous concentrons sur la présentation et la discussion des résultats expérimentaux obtenus.

7.4.4 Résultats expérimentaux : présentation et discussion

Les résultats expérimentaux ont été obtenus grâce à l'outil TARGET. Ils sont de plusieurs nature, chacune respectant un des scénarios présentés plus haut et concernent deux aspects majeurs de la recherche d'information, la pertinence des résultats et la performance du système. Le premier aspect est le plus important du fait que l'approche TARGET tend à améliorer la qualité des résultats en terme de pertinence. Le second aspect est moins important dans notre approche, néanmoins il est nécessaire de le discuter car il fait partie des critères sur lesquels sont évalués les systèmes pour la recherche d'information. Dans cette partie, nous allons tout d'abord présenter les résultats obtenus, puis nous allons les commenter plus largement. La présentation des résultats est un passage nécessaire avant de les commenter et de tirer les conclusions sur la valeur ajoutée des concepts testés pour l'amélioration de la recherche documentaire. Nous

allons dans un premier temps présenter les résultats relatifs à la pertinence des résultats et dans un second temps, nous décrivons brièvement ceux permettant de discuter de la performance du système.

Pertinence des résultats

La pertinence des résultats d'une recherche documentaire est l'aspect auquel nous nous sommes intéressé dans ces travaux. Les résultats expérimentaux obtenus à l'aide de l'outil TARGET concernent à la fois la précision et le rappel des résultats retournés.

Précision des résultats retournés

Les résultats obtenus lors de la réalisation des expérimentations des quatre premier scénarios et de la mesure de la précision des résultats retournés par l'outil sont présentés sur le graphique de la figure 7.6.

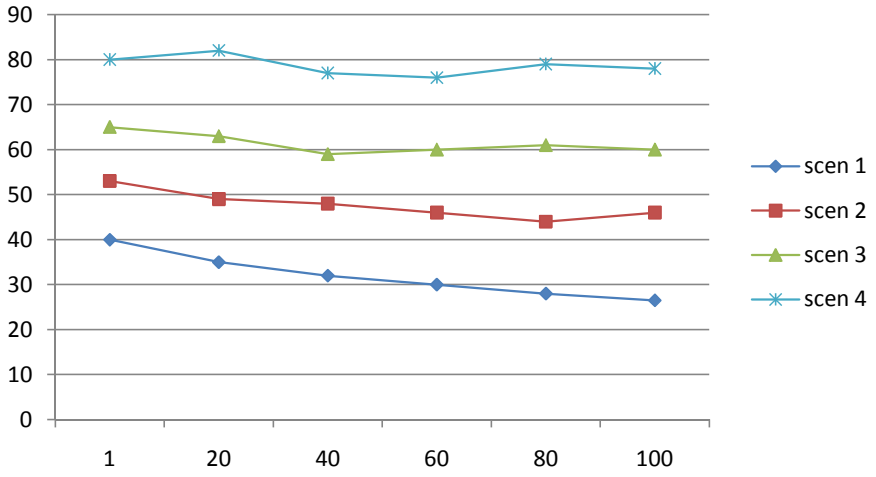


FIGURE 7.6 – Précision des résultats

Sur le graphique ci-dessus, nous observons que les résultats obtenus montrent une amélioration sensible de la pertinence des information retournées au fur et à mesure que les concepts de l'approche sont utilisés dans les expérimentations. Ceci démontre que les concepts mis en œuvre contribuent sensiblement à l'amélioration de la pertinence des résultats d'une recherche d'information.

De façon plus particulière, les résultats obtenus dans le cadre du premier scénario (scen. 1 sur le graphique) montre une précision de 37% en moyenne bien moins bonne que pour les autres scénarios. Ceci montre que l'approche classique de recherche d'information mise en œuvre dans les principaux moteurs de recherche est la moins précise. Nous expliquons principalement ces résultats par le caractère ambigu des requêtes (aucune expansion de requêtes n'est effectuée) pouvant viser plusieurs domaines à la fois.

Cette explication est confirmée par les résultats obtenus relativement au deuxième scénario (scen. 2 sur la figure 7.6) pour lesquels la précision atteint environ 45% en moyenne. Ces expérimentations mettaient l'accent sur l'apport des règles d'enrichissement, nous observons une amélioration de près de 10% de la précision des résultats retournés par l'outil. Ceci démontre deux choses :

1. Une certaine qualité de l'ontologie utilisée dans le processus d'enrichissement des requêtes ASK.
2. Le bien-fondé des règles d'enrichissement proposées au chapitre 5.

Cependant, les résultats montrent encore certaines insuffisances pouvant s'expliquer dans la qualité du choix des mots-clés additionnels. L'exploitation des éléments caractéristiques des ontologies adaptatives et notamment du poids sémantique, de la date d'émergence et de la distance sémantique entre les concepts permettent en partie de corriger ce manque. Les résultats obtenus dans le cadre du scénario 3 (scen 3 sur le graphique) permettent de confirmer cette affirmation car la précision des résultats atteint 63%. Ceci illustre la valeur ajoutée des caractéristiques propres aux ontologies adaptatives dans l'amélioration de la pertinence des résultats d'une recherche documentaire et permet de valider expérimentalement le poids sémantique, la distance sémantique et la date d'émergence (les autres éléments du modèle des ontologies adaptatives : la distance sémantique et la résistance aux changements feront l'objet d'une validation à travers les expériences mises en place dans le cadre du scénario 5).

Une autre amélioration de la précision des résultats est possible en contraignant davantage la requête initiale avec des informations relatives aux caractéristiques des utilisateurs. La sélection des mots-clés supplémentaires pour contraindre davantage la requête en s'appuyant sur l'ontologie représentant les vues de l'utilisateur sur le domaine permet d'augmenter de manière significative la précision des résultats comme le montre la figure 7.6 (scen 4). Ce procédé consiste à réduire le domaine de recherche visé par la requête initiale à une partie de ce domaine intéressant particulièrement l'utilisateur. Les résultats obtenus soulignent l'intérêt de considérer les profils des utilisateurs pour la recherche d'information.

L'impact positif du processus d'adaptation des ontologies se reflète complètement sur la précision des résultats d'une recherche d'information comme le montre le graphique de la figure 7.7.

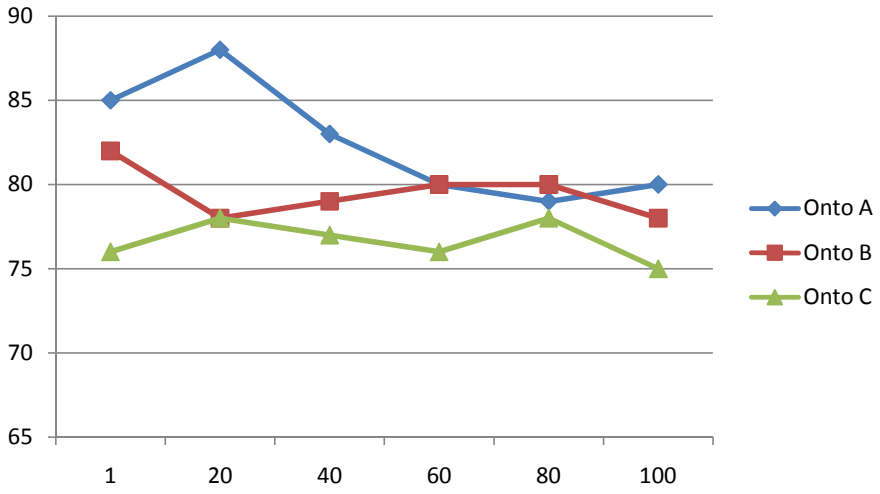


FIGURE 7.7 – Précision des résultats dans le cadre du scénario 5

Le graphique montre que les résultats obtenues en utilisant l'ontologie A, c'est-à-dire celle qui a évoluée selon le processus d'adaptation du chapitre 3, sont plus précis que ceux obtenus en utilisant les autres ontologies ayant évolué différemment (i.e. dont l'attribution des coefficients assignés aux éléments propres aux ontologies adaptatives ne respecte pas le processus décrit au chapitre 3). De plus, les résultats sont plus précis que ceux obtenus dans le cadre du scénario 4.

Ceci met l'accent sur trois critères importants :

1. L'importance de considérer des ontologies adaptatives dont les caractéristiques et les éléments représentent plus fidèlement l'état des connaissances du domaine à un moment précis du temps.
2. La qualité du processus d'adaptation que nous avons défini au chapitre 3.
3. La validité des concepts dénotant la résistance aux changements et la période de validité. Ces résultats nous permettent de compléter la validation du modèle des ontologies adaptatives entamée avec les expérimentations du scénario 3.

Les ontologies adaptatives vont bien dans le sens du Web Sémantique dont l'objectif principal est d'optimiser la compréhension des données et des ressources du Web par les machines afin de les retrouver plus facilement. Les résultats sur la précision des résultats d'une recherche documentaire nous permettent de dire que les ontologies adaptatives remplissent bien cet objectif. Le rappel des résultats est l'objet de la sous-section suivante.

Rappel des résultats retournés

Le rappel est défini par le nombre de documents pertinents retrouvés au regard du nombre de documents pertinents que possède le corpus. Les résultats concernant la mesure du rappel sont représentés sur le graphique de la figure 7.8 (pour les scénarios 1 à 4) et sur celui de la figure 7.9 (pour le scénario 5).

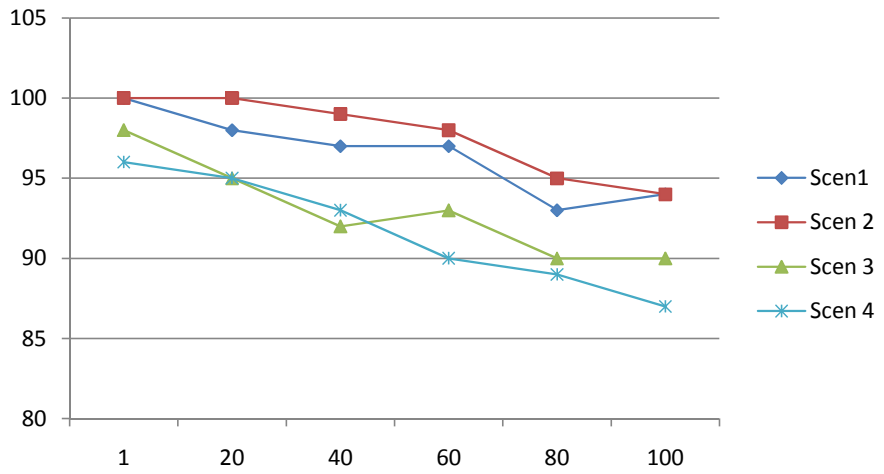


FIGURE 7.8 – Rappel des résultats

De manière générale, les résultats obtenus pour la mesure du rappel sont très élevés (supérieurs à 85%) quelque soit le scénario envisagé. Plus particulièrement, on observe que le rappel des résultats est élevé lorsque la requête est très peu contrainte (i.e. lorsqu'elle ne contient que peu de mots-clés) c'est pourquoi le rappel des résultats obtenus dans le cadre des premiers scénarios est meilleur. Les règles d'enrichissement des requêtes que nous proposons sont avant tout basées sur l'utilisation de l'opérateur de conjonction. Ainsi, les mots-clés additionnels sont vues comme des contraintes supplémentaires ayant pour effet de restreindre au maximum le domaine de recherche. Si ce procédé a pour effet d'augmenter sensiblement la précision des résultats, il a un impact plus négatif sur le rappel car les documents se situant à la frontière du domaine

avant l'enrichissement se retrouvent en dehors après l'expansion de la requête et ne sont donc pas retournés.

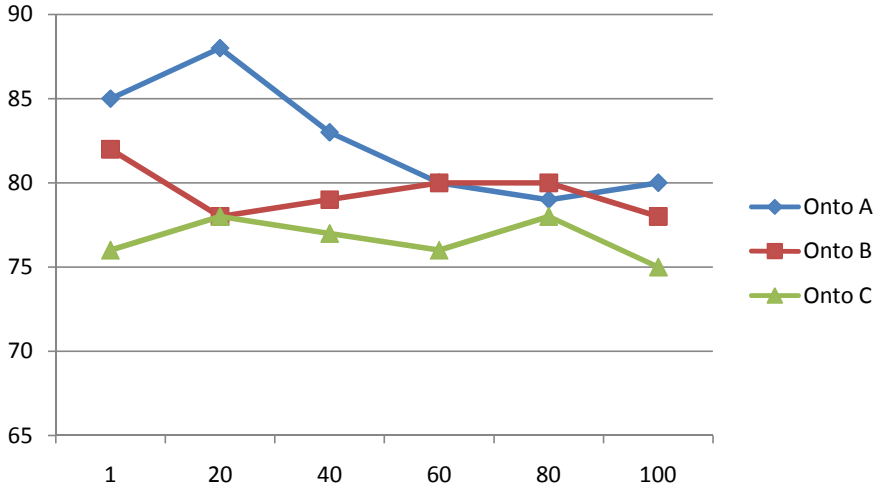


FIGURE 7.9 – Rappel des résultats dans le cadre du scénario 5

Les résultats obtenus dans le cadre des expérimentations du cinquième scénario pour la validation du processus d'adaptation confirment ceux présentés précédemment. Cependant, le rappel des résultats est un peu moins bon dans les expériences utilisant l'ontologie ayant évolué suivant le processus d'adaptation du chapitre 3. Ces chiffres peuvent s'expliquer par le fait suivant : Les concepts obsolètes supprimés de l'ontologie représentant le domaine de recherche ne peuvent plus être utilisés pour l'enrichissement des requêtes. Par conséquent, le domaine visé par ces mots-clés ne peut plus l'être du fait de leur suppression de l'ontologie. Ceci explique la différence entre le rappel obtenue avec l'utilisation de l'ontologie A (celle ayant évolué suivant notre processus d'adaptation) et celui obtenue avec l'ontologie B où aucun concept n'est supprimé. Une façon de corriger ce problème serait de rallonger suffisamment le pas d'évolution de l'ontologie afin que les concepts de l'ontologie y persistent plus longtemps.

Les résultats obtenus pour la mesure du rappel des résultats d'une recherche d'information nous réconfortent pour dire que les concepts introduits permettent d'améliorer la pertinence des résultats d'une recherche. Même si le rappel des résultats est moins élevé pour les résultats obtenus suivant l'approche TARGET (scénario 5, ontologie A) que ceux obtenus grâce aux outils usuels pour la recherche d'information, la différence est due à l'enrichissement des requêtes et principalement au choix des mots-clés additionnels. Par contre, nous observons que les résultats les plus pertinents sont à chaque fois retournés. Les documents filtrés sont ceux situés en bordure du domaine visé. Le fait d'avoir mesuré manuellement le rappel a également contribué à l'élimination de documents pouvant être considérés comme pertinents par certains utilisateurs. Ceci limite donc également la mesure du rappel.

Analyse critique de la précision et du rappel des résultats

L'analyse des résultats mesurant la précision et le rappel de la recherche a démontré la contribution des concepts introduits dans le cadre de l'approche TARGET pour l'amélioration de la pertinence de la recherche d'information au moins dans le cadre expérimental que nous avons

défini. Les résultats ont montré une précision accrue et un rappel de bonne qualité surtout par rapport aux applications usuelles de recherche de documents sur le Web.

Les résultats expérimentaux obtenus montrent que la pertinence des résultats est bonne suivant les critères que nous avons définis (des documents à jour et en rapport avec le domaine et le profil des utilisateurs). Cependant, seuls les utilisateurs sont à même de décider de la pertinence d'une information par rapport à leurs besoins initiaux. C'est pourquoi, dans le but de valider ce point et de conforter définitivement notre position, une enquête à grande échelle consistant à demander à un nombre significatif d'utilisateurs d'évaluer l'outil TARGET et plus particulièrement la qualité de la recherche documentaire aurait été nécessaire. Cette étude aurait également permis de déterminer de la validité de la méthode de classement des résultats que nous avons proposée au chapitre 6.

Le cadre expérimental que nous nous sommes donné (i.e. la conférence internationale World Wide Web) et le cas d'étude pour la recherche d'articles scientifiques n'est pas suffisant pour certifier que l'approche TARGET va donner de tels résultats dans tous les domaines du Web, mais les caractéristiques offertes par le domaine et le cas d'étude nous permettent de prédire un bon fonctionnement de l'approche dans un grand nombre de domaines du Web.

Performance du système

Les résultats obtenus sur la pertinence des résultats de la recherche présentés précédemment ont donné pleine satisfaction. Il est alors nécessaire, comme c'est la cas dans toutes les approches dédiées à la recherche d'information, de discuter de la performance (en temps de calcul) du système.

De manière générale, le temps dépensé au cours du processus de recherche est moyen. Nous entendons par la que le processus complet de recherche (celui décrit sur la figure 7.2 utilisant Google) prends en tout moins d'une minute. Ce temps d'exécution s'explique par différents facteurs :

- Le temps de construction des WPGraphs et W^{3}Graphs. Rappelons que ce temps de construction est fonction de la taille de la page Web (en nombre de termes et métadonnées) et de la taille de l'ontologie (en nombre d'éléments ontologiques) du domaine utilisée lors du processus de construction. Plus une page Web est importante et plus l'ontologie est grande, plus le temps de calcul est important (voir section 4.4.1 et 4.4.1 pour le détail sur la complexité des algorithmes de construction).
- La qualité du réseau pour télécharger les pages Web dans le but de construire les WPGraphs et W^{3}Graphs qui leurs sont associés. Un réseau lent va augmenter le temps du processus de recherche. Pour contrecarrer les effets du réseau et du temps de calcul des graphes, nous avons opté pour l'utilisation d'un moteur de recherche usuel (Google dans notre cas) et d'appliquer la méthode TARGET sur les 100 premiers résultats retournés par l'application. Ceci a pour avantage de réduire le temps d'exécution de l'approche et de bénéficier des propriétés du PageRank donnant satisfaction à la grande majorité des utilisateurs.

Le temps d'exécution moyen du processus reprenant l'approche TARGET n'était pas la préoccupation principale des travaux décrits dans ce mémoire. C'est pourquoi une optimisation de

la complexité des algorithmes mis en œuvre reste tout à fait envisageable dans les perspectives à ces travaux. Par contre, les très bons résultats obtenus lors de la mesure de la pertinence des pages Web retournées par l'approche TARGET restent un argument de poids pour satisfaire les utilisateurs dans leur quête d'information. Nous pensons qu'il est préférable de passer un peu de temps à «attendre» les résultats pertinents en sachant que l'information renvoyée va répondre aux besoins initiaux que d'obtenir immédiatement un grand nombre de résultats quelconques et de passer parfois plus de temps à les écrémer jusqu'à parvenir à l'information pertinente recherchée.

7.5 Conclusion

Dans ce chapitre, nous avons présenté successivement le cas d'étude qui a servi de base à la validation expérimentale de l'approche, l'outil TARGET grâce auquel nous avons obtenu les résultats expérimentaux ainsi que la validation expérimentale de l'approche TARGET sur la précision et le rappel des résultats d'une recherche d'information.

Le cas d'étude, fondation du processus de validation, a été défini de manière à être le plus réaliste possible pour renforcer la crédibilité de nos concepts dans la recherche d'information. Le domaine d'étude choisi est celui de la conférence internationale *World Wide Web* décrit par un ensemble de 633 documents composés essentiellement d'articles scientifiques et d'appels à communication des différentes conférences sur une décade. Les caractéristiques du domaine (de bonne taille et évolutif) a permis de mettre en évidence la contribution de tous les concepts mis en œuvre dans l'approche TARGET.

La mise en place du cas d'étude et l'obtention d'une quantité significative de résultats expérimentaux a nécessité le développement d'un outil. Les deux modules logiciels formant l'outil TARGET ont pour fonction la gestion des ontologies adaptatives et principalement l'application du processus d'adaptation ainsi que l'exécution d'une recherche documentaire sur le Web. Le composant permettant de gérer l'évolution des ontologies comble le manque d'outil adapté mis en évidence par notre étude alors que celui pour la recherche documentaire a avant tout été développé dans le but d'obtenir des résultats expérimentaux significatifs permettant de conclure quant à la validité de nos concepts.

Enfin, les résultats expérimentaux obtenus grâce à l'outil TARGET ont confirmé le bien-fondé de nos concepts et leur contribution dans l'amélioration de la pertinence des résultats d'une recherche d'information sur le Web. La précision des résultats est sensiblement meilleure dans le cadre de notre approche que celle proposée avec les approches classiques de recherche d'information sur le Web grâce notamment aux effets dus à l'enrichissement des requêtes dont les mots-clés additionnels permettent de restreindre le domaine de recherche. Cependant, l'enrichissement des requêtes montre certaines limites se reflétant sur le rappel des résultats.

Pour finir, nous pouvons dire que la combinaison du cadre expérimental et de l'outil nous a permis de valider l'ensemble des concepts composant l'approche TARGET et de confirmer leur contribution dans l'amélioration de la pertinence des résultats retournés en réponse à une requête au moins pour le domaine défini dans notre cas d'étude.

Conclusion générale et perspectives

La place incontournable que prend le Web dans notre société contemporaine oblige la grande majorité des personnes à utiliser ce média et ceci principalement dans le but de rechercher des informations pertinentes. Cependant, la nature dynamique du Web et de la connaissance qu'il renferme rend la tâche de recherche complexe pour la plupart des utilisateurs. Dans ces travaux de thèse, nous avons présenté une approche originale visant à améliorer la pertinence des résultats lors d'une recherche documentaire sur le Web. L'approche proposée s'appuie sur plusieurs éléments dont les ontologies, pierre angulaire du Web Sémantique, des structures de données pour la représentation des données du Web, un langage de requête associé à ces structures ainsi qu'un mécanisme d'enrichissement de requête basé sur certaines relations ontologiques. Une des particularités de notre approche consiste à tenir compte à la fois de l'évolution des connaissances du Web et de celles de ses utilisateurs.

La prise en compte de l'évolution des connaissances du Web dans notre approche se fait grâce aux *ontologies adaptatives*. Nous avons construit ce nouveau modèle d'ontologie sur les idées empruntées à des domaines comme la psychologie et principalement celles émises par Jean Piaget et la biologie dont la théorie de l'évolution de Charles Darwin. Ce modèle introduit les notions de **poids sémantique**, de **date d'émergence**, de **durée de validité** au niveau des concepts de l'ontologie et celles de **distance sémantique** et de **résistance aux changements** au niveau des relations ontologiques. L'ensemble de ces nouveaux éléments permet de contrôler et d'adapter les connaissances de l'ontologie à celles du domaine.

L'adaptation des ontologies adaptatives que nous avons proposée est régie par un ensemble de six règles définissant le processus d'adaptation. Ce processus d'adaptation, s'appuyant sur un corpus de documents décrivant un domaine, permet notamment de détecter les nouvelles connaissances du domaine, d'éliminer les connaissances obsolètes, et de modifier les valeurs associées aux éléments propres aux ontologies adaptatives comme le poids sémantique et la distance sémantique en fonction de la résistance aux changements et de la dynamicité du corpus. De plus, l'application de ce processus d'adaptation est réalisée de manière cohérente c'est-à-dire que des tâches de raisonnement peuvent être exécutées sur l'ontologie adaptative résultante.

Nous avons ensuite montré une exploitation des ontologies adaptatives pour la représentation des données du Web à travers la définition des structures de données que sont les WPGraphs et W^3Graphs.

La première structure de graphe que nous avons proposée, les WPGraphs, a été conçue pour la représentation des données d'une page Web. Construit à partir de l'ontologie adaptative représentant le domaine de recherche dans lequel une requête Web est émise, ce graphe consiste en une annotation des données d'une page Web. L'ensemble des sommets du graphe contient les termes d'une page Web et les arêtes du graphe représentent les liens sémantiques potentiels (déterminé grâce à l'ontologie adaptative du domaine) entre les termes. Cette représentation sous

forme de graphe des données d'une page Web et son formalisme basé sur la logique du premier ordre va favoriser la recherche d'information sur le Web.

L'autre structure de graphe, les W^{3}Graphs, consiste en la représentation enrichie d'un ensemble de pages Web. Les sommets de ce type de graphe sont des WPGraphs, donc une représentation annotée des données d'une page Web, alors que les arêtes de ce graphe également construites à partir de l'ontologie du domaine représentent les liens sémantiques entre ces pages. De plus, la pondération des arêtes de ce graphe permet de mesurer la force de ce lien entre les pages Web permettant ainsi une représentation du Web en fonction de la sémantique du contenu et non pas de sa structure basée sur les hyperliens.

Cette représentation originale des données du Web permet d'une part d'améliorer la pertinence de la recherche documentaire sur le Web et d'autre part, de définir une nouvelle méthode de classement des résultats d'une recherche en fonction de la proximité de cette page et du domaine de recherche (représenté par une ontologie adaptative) visé par une requête.

La définition des structures de données que sont les WPGraphs et W^{3}Graphs pour la représentation des données du Web a nécessité la proposition d'un langage de requête adapté pour l'extraction de l'information pertinente de ces structures. Dans ce but, nous avons proposé le langage ASK. Ce langage, inspiré principalement des langages de requête des moteurs de recherche usuels, fonctionne suivant un modèle booléen et offre un ensemble d'opérateurs pour la construction de requête. En plus des opérateurs de disjonction, de conjonction et de négation communs à tous les langages des moteurs de recherche pour le Web, ASK propose également des opérateurs pour exprimer la disjonction exclusive entre les termes ainsi que la notion de contraire et permet à l'utilisateur de spécifier directement au niveau de la requête, le domaine de recherche visé.

Le langage ASK a également été pensé afin que les requêtes exprimées dans ce langage puisse facilement être étendues. Le mécanisme d'enrichissement des requêtes ASK que nous avons défini a pour but d'étendre la requête par des mots-clés bien choisi afin de contraindre suffisamment la requête pour lever toutes ambiguïtés lors de l'interprétation de celle-ci et que le type d'information recherché par l'utilisateur soit spécifié pour satisfaire ce dernier. Les règles d'enrichissement des requêtes tiennent compte de la forme de la requête initiale et principalement des opérateurs logiques composant la requête. De plus, la sélection des mots-clés additionnels est faite parmi les labels des concepts constituant l'ontologie adaptative du domaine et de celle représentant les vues de l'utilisateur sur ce domaine. Ce choix est conditionné par les relations ontologiques reliant les mots-clés de la requête aux concepts des ontologies et des caractéristiques propres aux ontologies adaptatives (i.e. poids sémantique, distance sémantique et date d'émergence).

Nous avons également développé un outil pour la gestion des ontologies adaptatives et la recherche d'information sur le Web. L'outil développé grâce aux technologies Java et PROLOG bénéficie des avantages de ces deux technologies. Fédérant une large communauté d'intérêt, les API et plus généralement les outils Java nous ont permis de développé une interface graphique facilitant à la fois la gestion des ontologies, la saisie des requêtes, la sélection des ontologies et la lecture des résultats d'une recherche. Le module PROLOG représente le moteur permettant la modification des éléments d'une ontologie adaptative et la vérification des requêtes ASK sur la structure logique des WPGraphs et W^{3}Graphs.

L'outil que nous avons développé a servi de base à la validation expérimentale des concepts composant l'approche TARGET. Dans ce but, nous avons défini un cas d'étude portant sur la recherche d'articles scientifiques publiés à la conférence World Wide Web sur une période de dix ans pour lequel deux profils d'utilisateurs (celui des chercheurs fondamentaux et des chercheurs

appliqués) ont été définis. Les résultats expérimentaux obtenus ont montré une réelle valeur ajoutée de chaque concept (ontologies adaptatives, profils utilisateurs et règles d'enrichissement des requêtes) dans l'amélioration de la pertinence des résultats d'une recherche documentaire sur le Web.

Perspectives

Les travaux présentés dans ce mémoire ont donné satisfaction sur bien des points et notamment par rapport à l'objectif initial qui était l'amélioration de la pertinence des résultats d'une recherche documentaire. Cependant, l'approche TARGET a également montré certaines limites et offre des perspectives intéressantes pour l'extension de ses travaux.

Une des principales limites de notre approche est le temps d'exécution du processus de recherche. Celui-ci est fonction du temps de construction des WPGraphs et W^3Graphs, donc la taille (en quantité d'information) des pages Web qui leur sont associées et de la vitesse du réseau pour télécharger les pages Web. En conséquence, une manière d'améliorer l'approche consiste à optimiser les algorithmes de construction des WPGraphs et W^3Graphs. L'utilisation du moteur Google a déjà permis de réduire ce temps de construction tout comme l'utilisation de l'ontologie adaptative du domaine et la restriction du graphe aux parties intéressantes de la page Web. Cependant, nous pensons qu'il est possible d'une part d'optimiser la complexité des algorithmes de construction des graphes ou encore de travailler sur l'implémentation de ces algorithmes en exploitant par exemple une architecture parallèle afin de traiter plusieurs pages Web à la fois. Le stockage des graphes après leur construction est également une option vers l'optimisation du temps d'exécution du processus de recherche documentaire sur le Web. L'idée est de mettre en cache les graphes et de vérifier à intervalles de temps réguliers si les ontologies nécessaires à leur construction ainsi que la page Web elle même ont évolué auquel cas une reconstruction des graphes sera nécessaire et pourra être effectuée en dehors du processus de recherche déclenché au moment où un utilisateur pose sa requête.

L'idée d'automatiser complètement le processus d'adaptation des ontologies représente également une perspective intéressante aux travaux présentés dans ce mémoire. L'ajout des nouveaux concepts à une ontologie adaptative, réalisé de manière semi-automatique dans notre approche, représente la partie du processus pouvant bénéficier des travaux existants dans le domaine de l'ingénierie des connaissances en vue d'une complète automatisation. Les avancées significatives dans le domaine de l'alignement [Noy and Stuckenschmidt, 2005] ou l'appariement [Euzenat and Shvaiko, 2007] d'ontologies privilégiant l'aspect distribué des ontologies serait une piste de recherche à investiguer en vue de l'intégration automatique des nouveaux concepts dans une ontologie. La *sémantique émergente* [Cudré-Mauroux et al., 2006] (emergent semantic en anglais) est un nouveau domaine de recherche dans lequel la sémantique d'un domaine est acquise à travers l'interaction d'éléments logiciels entre eux et de la réponse à leurs interactions. Les premiers résultats obtenus dans ce domaine montrent également des perspectives intéressantes en vue de l'automatisation de l'évolution d'ontologies et l'ajout de concepts en particulier. Il en va de même pour les autres parties du processus d'adaptation nécessitant l'intervention d'un expert du domaine. En effet, le phénomène de sémantique émergente permettrait de déterminer automatiquement la valeur attribuée à chaque élément spécifique du modèle des ontologies adaptatives ce qui augmenterait davantage la fiabilité de leur adaptation à travers le temps ce qui, par conséquent, accroîtrait les possibilités d'utilisation des ontologies adaptatives.

Les ontologies adaptatives et principalement leur expression dans la famille de langage OWL peuvent donner lieu à une réflexion plus approfondie au niveau du langage lui-même. Nous avons montré que les constructeurs du langage permettaient d'exprimer les éléments du modèle des ontologies adaptatives à travers l'utilisation du constructeur *owl :AnnotationProperty*. Cependant, la méthode proposée n'était pas complètement adaptée notamment pour exprimer la distance sémantique entre des concepts liés par une relation de subsumption ou d'équivalence. Par conséquent, une réflexion est nécessaire afin, d'une part, de proposer de nouveaux constructeurs pour l'expression de ces éléments et, d'autre part, de l'impact de ces opérateurs sur la complexité du langage et notamment sur le processus de raisonnement utilisant ce type d'ontologies.

De même, d'un point de vue de l'enrichissement des requêtes et plus précisément par rapport à l'exploitation des relations ontologiques de méronymie et d'opposé que nous avons proposées, le langage OWL mérite d'être étendu. Les axiomes logiques que nous avons établis pour la formalisation de ces relations est un premier pas vers la définition de nouveaux constructeurs OWL pour exprimer ces notions. Néanmoins, comme pour l'intégration des éléments pour l'adaptation des ontologies, une réflexion plus profonde est nécessaire sur la nature des opérateurs du langage à définir et de leur impact sur la qualité du raisonnement utilisant ce type d'ontologies.

Enfin, la validation de l'approche peut être renforcée par une enquête à plus grande échelle sur la qualité du système à mener auprès des utilisateurs. Une telle enquête pourrait, d'une part, confirmer nos résultats expérimentaux sur la qualité des concepts de l'approche TARGET et leur impact sur la pertinence des résultats d'une recherche documentaire. D'autre part, l'avis des utilisateurs sur la qualité du système permettrait de perfectionner l'outil afin d'en proposer une interface plus conviviale qui faciliterait d'autant plus la saisie des requêtes, la sélection des ontologies et surtout la visualisation des résultats d'une recherche à l'écran.

Bibliographie

[Afsharchi and Far, 2006] Afsharchi, M. and Far, B. H. (2006). Automated ontology evolution in a multi-agent system. In *InfoScale '06 : Proceedings of the 1st international conference on Scalable information systems*. ACM Press.

[Alani et al., 2003] Alani, H., Kim, S., Millard, D. E., Weal, M. J., Hall, W., Lewis, P. H., and Shadbolt, N. R. (2003). Automatic ontology-based knowledge extraction from web documents. *IEEE Intelligent Systems*, 18(1) :14–21.

[Alchourron et al., 1985] Alchourron, C. E., Gardenfors, P., and Makinson, D. (1985). On the logic of theory change : Partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50(2) :510–530.

[Alferes et al., 2004] Alferes, J. J., Bailey, J., Berndtsson, M., Bry, F., Dietrich, J., Kozlenkov, A., May, W., Patrânjan, P. L., Pinto, A., Schroeder, M., and Wagner, G. (2004). State-of-the-art on evolution and reactivity. deliverable I5-D1, Centro de Inteligência Artificial - CENTRIA, Universidade Nova de Lisboa.

[Arocena et al., 1997] Arocena, G. O., Mendelzon, A. O., and Mihaila, G. A. (1997). Applications of a web query language. In *Proc. 6th International World Wide Web Conference*, pages 587–595, Santa Clara, California, USA.

[Auer and Herre, 2006] Auer, S. and Herre, H. (2006). A Versioning and Evolution Framework for RDF Knowledge Bases. In Virbitskaite, I. and Voronkov, A., editors, *Ershov Memorial Conference*, volume 4378 of *LNCS*, pages 55–69.

[Aussenac-Gilles et al., 2000] Aussenac-Gilles, N., Biebow, B., and Szulman, N. (2000). Revisiting ontology design : a method based on corpus analysis. In Dieng, R. and Corby, O., editors, *Knowledge engineering and knowledge management : methods, models and tools, Proc. of the 12th International Conference on Knowledge Engineering and Knowledge Management*, volume 1937 of *LNCS*, pages 172–188, Antibes, France. Springer Verlag.

[Avery and Yearwood, 2003] Avery, J. and Yearwood, J. (2003). Dowl : A dynamic ontology language. In *Proceedings of the IADIS International Conference WWW/Internet 2003*, pages 985–988, Algarve, Portugal. IADIS.

[Baader et al., 2003] Baader, F., Calvanese, D., McGuinness, D., Nardi, D., and Patel-Schneider, P. F. (2003). *The Description Logic Handbook : Theory, Implementation, and Applications*. Cambridge University Press.

[Bachimont et al., 2002] Bachimont, B., Isaac, A., and Troncy, R. (2002). Semantic commitment for designing ontologies : A proposal. In *13th International Conference on Knowledge Engineering and Knowledge Management (EKAW'02)*, volume LNAI 2473, pages 114–121, Sigüenza, Spain.

[Banerjee et al., 1987] Banerjee, J., Kim, W., Kim, H.-J., and Korth, H. F. (1987). Semantics and implementation of schema evolution in object-oriented databases. *SIGMOD Rec.*, 16(3) :311–322.

[Barbier et al., 2003] Barbier, F., Henderson-Sellers, B., Parc-Lacayrelle, A. L., and Bruel, J.-M. (2003). Formalization of the whole-part relationship in the unified modeling language. *IEEE Transactions on Software Engineering*, 29(5) :459–470.

[Bechhofer et al., 2001] Bechhofer, S., Horrocks, I., Goble, C., and Stevens, R. (2001). OilEd : A Reason-able Ontology Editor for the Semantic Web. In *KI '01 : Proceedings of the Joint German/Austrian Conference on AI*, pages 396–408. Springer-Verlag.

[Bender et al., 2008] Bender, M., Crecelius, T., Kacimi, M., Michel, S., Neumann, T., Parreira, J., Schenkel, R., and Weikum, G. (2008). Exploiting social relations for query expansion and result ranking. In *Proceedings of the 24th International Conference on Data Engineering Workshop*, pages 501–506.

[Berners-Lee et al., 1992] Berners-Lee, T., Cailliau, R., Groff, J.-F., and Pollermann, B. (1992). World-Wide Web : The Information Universe. *Electronic Networking : Research, Applications and Policy*, 1(2) :74–82.

[Berners-Lee et al., 2001] Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The Semantic Web. *Scientific American*, 284(5) :34–43.

[Bollen et al., 2005] Bollen, J., de Sompel, H. V., Smith, J. A., and Luce, R. (2005). Toward alternative metrics of journal impact : a comparison of download and citation data. *Inf. Process. Manage.*, 41(6) :1419–1440.

[Bouaud et al., 1994] Bouaud, J., Bachimont, B., Charlet, J., and Zweigenbaum, P. (1994). Acquisition and structuring of an ontology within conceptual graphs. In *Proceedings of ICCS'94 Workshop on Knowledge Acquisition using Conceptual Graph Theory*, pages 1–25, University of Maryland, College Park, MD.

[Brin, 1998] Brin, S. (1998). Extracting patterns and relations from the world wide web. In *WebDB Workshop at EDBT 98*.

[Broder et al., 2000] Broder, A. Z., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A., and Wiener, J. L. (2000). Graph structure in the web. *Computer Networks*, 33(1-6) :309–320.

[Brusilovsky, 1996] Brusilovsky, P. (1996). Methods and techniques of adaptive hypermedia. *User modeling and user-adapted interaction*, 6(2-3) :87–129.

[Bry and Ecker, 2006] Bry, F. and Ecker, M. (2006). Twelve theses on reactive rules for the web. In *Workshop "Reactivity on the Web" at the International Conference on Extending Database Technology*, Munich, Germany.

[Bry et al., 2005] Bry, F., Koch, C., Furche, T., Schaffert, S., Badea, L., and Berger, S. (2005). Querying the web reconsidered : Design principles for versatile web query languages. *International Journal on Semantic Web and Information Systems*, 1(2) :1–21.

[Bry and Patrânjan, 2005] Bry, F. and Patrânjan, P.-L. (2005). Reactivity on the web : paradigms and applications of the language xchange. In *SAC '05 : Proceedings of the 2005 ACM symposium on Applied computing*, pages 1645–1649, Santa Fe, New Mexico. ACM Press.

[Buitelaar et al., 2005] Buitelaar, P., Cimiano, P., and Magnini, B. (2005). Ontology learning from text : An overview. In *Ontology Learning from Text : Methods, Evaluation and Applications Frontiers in Artificial Intelligence and Applications Series*, volume 123. IOS Press.

[Castano et al., 2006] Castano, S., Ferrara, A., and Montanelli, S. (2006). A matchmaking-based ontology evolution methodology. In *CAiSE INTEROP Workshop on Enterprise Modelling and Ontologies for Interoperability (EMOI - INTEROP 2006)*, pages 547–558, Luxembourg.

[Chakrabarti et al., 1999] Chakrabarti, S., van den Berg, M., and Dom, B. (1999). Focused crawling : a new approach to topic-specific web resource discovery. In *WWW '99 : Proceedings of the eighth international conference on World Wide Web*, pages 1623–1640, Toronto, Canada. Elsevier North-Holland, Inc.

[Charlet, 2002] Charlet, J. (2002). L'ingénierie des connaissances : développements, résultats et perspectives pour la gestion des connaissances médicales. Habilitation à diriger des recherches, Université Paris 6.

[Chen and Matthews, 2007] Chen, C. and Matthews, M. M. (2007). Extending description logic for reasoning about ontology evolution. In *WI '07 : Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, pages 452–456. IEEE Computer Society.

[Chen and Sycara, 1998] Chen, L. and Sycara, K. (1998). WebMate : A personal agent for browsing and searching. In Sycara, K. P. and Wooldridge, M., editors, *Proceedings of the 2nd International Conference on Autonomous Agents (Agents'98)*, pages 132–139, New-York. ACM Press.

[Chirita et al., 2007] Chirita, P. A., Firan, C. S., and Nejdl, W. (2007). Personalized query expansion for the web. In *SIGIR '07 : Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 7–14, Amsterdam, The Netherlands. ACM.

[Collins-Thompson and Callan, 2005] Collins-Thompson, K. and Callan, J. (2005). Query expansion using random walk models. In *Proceedings of the 2005 ACM CIKM International Conference on Information and Knowledge Management*, pages 704–711.

[Corcho et al., 2002] Corcho, O., Fernandez-Lopez, M., Perez, A. G., and Vicente, O. (2002). WebODE : An Integrated Workbench for Ontology Representation, Reasoning, and Exchange. In *EKAW '02 : Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web*, pages 138–153. Springer-Verlag.

[Corcho and Pérez, 2000] Corcho, O. and Pérez, A. G. (2000). A roadmap to ontology specification languages. In R. Dieng, O. C., editor, *Proceedings of the 12th European Workshop on Knowledge Acquisition, Modeling and Management*, volume 1937 of *Lecture Notes in Computer Science*, pages 80–96, Juan-les-Pins, France. Springer-Verlag GmbH.

[Coyle and Smyth, 2007] Coyle, M. and Smyth, B. (2007). Supporting intelligent web search. *ACM Trans. Internet Technol.*, 7(4) :20.

[Cudré-Mauroux et al., 2006] Cudré-Mauroux, P., Aberer, K., Abdelmoty, A. I., Catarci, T., Damiani, E., Illarramendi, A., Jarrar, M., Meersman, R., Neuhold, E. J., Parent, C., Sattler, K.-U., Scannapieco, M., Spaccapietra, S., Spyns, P., and Tré, G. D. (2006). Viewpoints on emergent semantics. *J. Data Semantics VI*, LNCS 4090 :1–27.

[De Bra, 1999] De Bra, P. (1999). Design issues in adaptive hypermedia application development. In Brusilovsky, P. and Bra, P. D., editors, *Proceedings of the Second Workshop on Adaptive Systems and User Modeling on the World Wide Web*, pages 29–39, Toronto and Banff, Canada.

[De Bra and Calvi, 1998] De Bra, P. and Calvi, L. (1998). AHA ! An open adaptive hypermedia architecture. *The New Review of Hypermedia and Multimedia*, 4 :115–139.

[De Leenheer, 2004] De Leenheer, P. (2004). Revising and managing multiple ontology versions in a possible worlds setting. In *Proc. of On the Move to Meaningful Internet Systems 2004 : OTM 2004 Workshops*, volume 3292 of *LNCS*, pages 798–809. Springer Berlin / Heidelberg.

[Denny, 2004] Denny, M. (2004). Ontology tools survey, revisited. http ://www.xml.com/pub/a/2004/07/14/onto.html.

[Dhyani et al., 2002] Dhyani, D., Ng, W. K., and Bhowmick, S. S. (2002). A survey of web metrics. *ACM Comput. Surv.*, 34(4) :469–503.

[Edmundson, 1967] Edmundson, H. P. (1967). axiomatic characterization of synonymy and antonymy. In *Proceedings of the 1967 conference on Computational linguistics*, pages 1–11, Morristown, NJ, USA. Association for Computational Linguistics.

[Euzenat and Shvaiko, 2007] Euzenat, J. and Shvaiko, P. (2007). *Ontology Matching*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[Farquhar et al., 1997] Farquhar, A., Fikes, R., and Rice, J. (1997). The ontolingua server : A tool for collaborative ontology construction. *Journal of Human-Computer Studies*, 46 :707–728.

[Fattahia et al., 2008] Fattahia, R., Wilson, C. S., and Colea, F. (2008). An alternative approach to natural language query expansion in search engines : Text analysis of non-topical terms in web documents. *Information Processing and Management*, 44(4) :1503–1516.

[Fellbaum, 1998] Fellbaum, C. D. (1998). *WordNet : An Electronic Lexical Database*. MIT Press, Cambridge, MA.

[Fensel, 2001] Fensel, D. (2001). Ontologies : dynamics networks of meaning. In *Proceedings of the 1st Semantic web working symposium*, Stanford, CA, USA.

[Fernandez et al., 1997] Fernandez, M., Gomez-Perez, A., and Juristo, N. (1997). METHONTOLOGY : from Ontological Art towards Ontological Engineering. In *Proceedings of the AAAI97 Spring Symposium Series on Ontological Engineering*, pages 33–40, Stanford, USA.

[Flouris and Plexousakis, 2005] Flouris, G. and Plexousakis, D. (2005). Handling ontology change : Survey and proposal for a future research direction. Technical Report FORTH-ICS/TR-362, FORTH-ICS.

[Flouris et al., 2006] Flouris, G., Plexousakis, D., and Antoniou, G. (2006). Evolving ontology evolution. In *SOFSEM 2006 : Theory and Practice of Computer Science, 32nd Conference on Current Trends in Theory and Practice of Computer Science*, volume 3831 of *Lecture Notes in Computer Science*, pages 14–29, Merín, Czech Republic.

[Frasincar et al., 2002] Frasincar, F., Houben, G., and Vdovjak, R. (2002). Specification framework for engineering adaptive web applications. In *Proc. World Wide Web Conference, Web Engineering track*.

[Freund and Toms, 2005] Freund, L. and Toms, E. G. (2005). Contextual search : from information behaviour to information retrieval. In *Proceedings of CAIS conference*.

[Fu et al., 2005] Fu, G., Jones, C. B., and Abdelmoty, A. I. (2005). Ontology-based spatial query expansion in information retrieval. In *ODBASE : OTM Confederated International Conferences*.

[Gauch et al., 2003] Gauch, S., Chaffee, J., and Pretschner, A. (2003). Ontology-based personalized search and browsing. *Web Intelli. and Agent Sys.*, 1(3-4) :219–234.

[Gendarmi and Lanubile, 2006] Gendarmi, D. and Lanubile, F. (2006). Community-driven ontology evolution based on folksonomies. In Heidelberg, S. B. ., editor, *On the Move to Meaningful Internet Systems 2006 : OTM 2006 Workshops*, volume 4277/2006 of *Lecture Notes in Computer Science*, pages 181–188.

[Gennari et al., 2002] Gennari, J. H., Musen, M. A., Fergerson, R. W., Grosso, W. E., Crubézy, M., Eriksson, H., Noy, N. F., and Tu, S. W. (2002). The Evolution of Protégé : An Environment for Knowledge-Based Systems Development. *International Journal of Human-Computer Studies*, 58 :89–123.

[Glover et al., 1999] Glover, E. J., Lawrence, S., Birmingham, W. P., and Giles, C. L. (1999). Architecture of a metasearch engine that supports user information needs. In *CIKM '99 : Proceedings of the eighth international conference on Information and knowledge management*, pages 210–216, Kansas City, Missouri, United States. ACM.

[Gärdenfors and Rott, 1995] Gärdenfors, P. and Rott, H. (1995). Belief revision. In *Handbook of logic in artificial intelligence and logic programming (Vol. 4) : epistemic and temporal reasoning*, pages 35–132. Oxford University Press.

[Gruber, 1993] Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowl. Acquis.*, 5(2) :199–220.

[Guarino, 1998] Guarino, N. (1998). Formal ontology and information systems. In *Proceedings of the 1st International Conference on Formal Ontologies in Information Systems (FOIS-98)*, pages 3–15. IOS Press.

[Guelfi and Pruski, 2006] Guelfi, N. and Pruski, C. (2006). On the use of ontologies for an optimal representation and exploration of the web. *Journal of Digital Information Management (JDIM)*, 4(3).

[Guelfi et al., 2007a] Guelfi, N., Pruski, C., and Reynaud, C. (2007a). Les ontologies pour la recherche ciblée d'information sur le Web : une utilisation et extension d'OWL pour l'expansion de requêtes. In *Proceedings of the Ingenierie des Connaissances 2007 (IC07) french conference*, Grenoble.

[Guelfi et al., 2007b] Guelfi, N., Pruski, C., and Reynaud, C. (2007b). Understanding and Supporting Ontology Evolution by Observing the WWW Conference. In *Proceedings of the International Workshop on Emergent Semantic and Ontology Evolution (ESOE 07)*, Busan, South-Korea.

[Guelfi et al., 2008] Guelfi, N., Pruski, C., and Reynaud, C. (2008). Towards the adaptive web using metadata evolution. In Calero, C., Ángeles Moraga, M., and Piattini, M., editors, *Handbook of research on Web information systems quality*. Idea Group Publishing.

[Gutierrez et al., 2006] Gutierrez, C., Hurtado, C., and Vaisman, A. (2006). The meaning of erasing in rdf under the katsuno-mendelzon approach. In *Proceedings of the 9th International Workshop on the Web and Databases (WebDB-06)*.

[Haase and Stojanovic, 2005] Haase, P. and Stojanovic, L. (2005). Consistent evolution of OWL ontologies. In Asunción Gómez-Pérez, J. E., editor, *Proceedings of the Second European Semantic Web Conference*, volume 3532 of *Lecture Notes in Computer Science*, pages 182–197, Heraklion, Greece. Springer.

[Haav and Lubi, 2001] Haav, H.-M. and Lubi, T.-L. (2001). A survey of concept-based information retrieval tools on the web. In Caplinkas, A. and Eder, J., editors, *Advances in Databases and Information Systems, Proc. of 5th East-European Conference ADBIS*2001*, volume 2, pages 29–41, Vilnius "Technika".

[Hahn and Marko, 2002] Hahn, U. and Marko, K. G. (2002). Ontology and lexicon evolution by text understanding. In *Proceedings of the ECAI 2002 Workshop on Machine Learning and Natural Language Processing for Ontology Engineering (OLT 2002)*.

[Halasz and Schwartz, 1994] Halasz, F. and Schwartz, M. (1994). The dexter hypertext reference model. *CACM*, 37(2) :30–39.

[Hansson, 1994] Hansson, S. O. (1994). Kernel contraction. *Journal of Symbolic Logic*, 59(3) :845–859.

[Heflin et al., 1999a] Heflin, J., Hendler, J., and Luke, S. (1999a). Coping with changing ontologies in a distributed environment. In *Ontology Management. Papers from the AAAI Workshop. WS-99-13*, pages 74–79. AAAI Press.

[Heflin et al., 1999b] Heflin, J., Hendler, J., and Luke, S. (1999b). SHOE : A Knowledge Representation Language for Internet Applications. Technical Report CS-TR-4078, Dept. of Computer Science, University of Maryland.

[Heflin and Hendler, 2000] Heflin, J. and Hendler, J. A. (2000). Dynamic ontologies on the web. In *AAAI/IAAI*, pages 443–449.

[Huang, 2000] Huang, L. (2000). A survey on web information retrieval technologies. Technical report, ECSL.

[Jansen et al., 2000] Jansen, B. J., Spink, A., and Pfaff, A. (2000). Linguistic aspects of web queries. In *American Society of Information Science*, Chicago.

[Jarrar and Meersman, 2002] Jarrar, M. and Meersman, R. (2002). Formal ontology engineering in the DOGMA approach. In *On the Move to Meaningful Internet Systems, 2002 - DOA/CoopIS/ODBASE 2002 Confederated International Conferences DOA, CoopIS and ODBASE 2002*, pages 1238–1254. Springer-Verlag.

[Jin et al., 2005] Jin, L., Liu, L., and Yang, D. (2005). A Model Transformation Based Conceptual Framework for Ontology Evolution. In *Knowledge-Based Intelligent Information and Engineering Systems, 9th International Conference, KES*, pages 325–331, Melbourne, Australia.

[Joho et al., 2002] Joho, H., Coverson, C., Sanderson, M., and Beaulieu, M. (2002). Hierarchical presentation of expansion terms. In *SAC '02 : Proceedings of the 2002 ACM symposium on Applied computing*, pages 645–649. ACM Press.

[Kalfoglou and Schorlemmer, 2003] Kalfoglou, Y. and Schorlemmer, M. (2003). Ontology mapping : the state of the art. *Knowl. Eng. Rev.*, 18(1) :1–31.

[Kang and Lau, 2004] Kang, S. H. and Lau, S. K. (2004). Ontology revision using the concept of belief revision. In *Proceedings of the 8th International Conference on Knowledge-Based Intelligent Information and Engineering Systems (KES-04), part III*, pages 8–15. Springer-Verlag.

[Keberle et al., 2007] Keberle, N., Litvinenko, Y., Gordeyev, Y., and Ermolayev, V. (2007). Ontology Evolution Analysis with OWL-MeT. In Flouris, G. and d'Aquin, M., editors, *Proc of the Int Workshop on Ontology Dynamics (IWOD'2007) at European Semantic Web Conference (ESWC)*, Innsbruck, Austria.

[Kelly et al., 2005] Kelly, D., Dollu, V. D., and Fu, X. (2005). The loquacious user : a document-independent source of terms for query expansion. In *SIGIR 2005 : Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 457–464.

[Kim and Choi, 1999] Kim, M.-C. and Choi, K.-S. (1999). A comparison of collocation-based similarity measures in query expansion. *Information Processing and Management*, 35(1) :19–30.

[Klein, 2004] Klein, M. (2004). *Change Management for Distributed Ontologies.* PhD thesis, Vrije Universiteit Amsterdam.

[Klein and Fensel, 2001] Klein, M. and Fensel, D. (2001). Ontology versioning for the semantic web. In *Proceedings of the International Semantic Web Working Symposium (SWWS)*, Stanford University, California, USA.

[Klein et al., 2002] Klein, M., Fensel, D., Kiryakov, A., and Ognyanov, D. (2002). Ontology versioning and change detection on the web. In *EKAW '02 : Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web*, pages 197–212. Springer-Verlag.

[Klein and Noy, 2003] Klein, M. and Noy, N. (2003). A component-based framework for ontology evolution. In *Workshop on Ontologies and Distributed Systems at IJCAI-03*, Acapulco, Mexico.

[Kleinberg et al., 1999] Kleinberg, J., Kumar, S., Raghavan, P., Rajagopalan, S., and Tomkins, A. (1999). The web as a graph : Measurements, models and methods. In *International Conference on Combinatorics and Computing.*

[Kleinberg, 1999] Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5) :604–632.

[Kotis and Vouros, 2006] Kotis, K. and Vouros, A. (2006). Human-centered ontology engineering : The HCOME methodology. *Knowl. Inf. Syst.*, 10(1) :109–131.

[Kudelka et al., 2007] Kudelka, M., Lehecka, O., Snasel, V., and El-Qawasmeh, E. (2007). Web pages clustering based on web patterns. In *Proceedings of the International Conference on Digital Information Management*, volume 2, pages 657–664, Lyon, France. IEEE Computer Society.

[Lam et al., 2004] Lam, S. J., Sleeman, D., and Vasconcelos, W. (2004). ReTAX+ : A Cooperative Taxonomy Revision Tool. In *Proceedings of AI-2004 Conference*, pages 64–77, Cambridge, UK. Springer-Verlag.

[Lassila, 1998] Lassila, O. (1998). Web metadata : A matter of semantics. *IEEE Internet Computing*, 2(4) :30–37.

[Lawrence, 2000] Lawrence, S. (2000). Context in web search. *IEEE Data Engineering Bulletin*, 23(3) :25–32.

[Lee et al., 1997] Lee, D. L., Chuang, H., and Seamons, K. (1997). Document ranking and the vector-space model. *IEEE Softw.*, 14(2) :67–75.

[Lee and Meyer, 2004] Lee, K. and Meyer, T. (2004). A classification of ontology modification. In *Proceedings of the 17th Australian Joint Conference on Artificial Intelligence (AI-04)*, LNCS, pages 248–258. Springer-Verlag.

[Leea et al., 2007] Leea, S., Seoa, W., Kanga, D., Kima, K., and Lee, J. Y. (2007). A framework for supporting bottom-up ontology evolution for discovery and description of grid services. *Expert Systems with Applications*, 32(2) :376–385.

[Leenheer and Mens, 2007] Leenheer, P. D. and Mens, T. (2007). Using graph transformation for collaborative ontology evolution. In *Proceedings of the Third International Symposium on Applications of Graph Transformation with Industrial Relevance (AGTIVE 2008)*, LNCS. Springer-Verlag.

[Leroy et al., 2003] Leroy, G., Lally, A. M., and Chen, H. (2003). The use of dynamic contexts to improve casual internet searching. *ACM Trans. Inf. Syst.*, 21(3) :229–253.

[Li et al., 1998] Li, W.-S., Shim, J., Candan, K. S., and Hara, Y. (1998). WebDB : A Web Query System and Its Modeling, Language, and Implementation. In *ADL '98 : Proceedings of the Advances in Digital Libraries Conference*, page 216.

[Liu et al., 2008] Liu, C., Zhang, Z., Xie, X., and Liang, T. (2008). Evaluation of meta-search engine merge algorithms. In *ICICSE '08. International Conference on Internet Computing in Science and Engineering*, pages 9–14.

[Maedche et al., 2003] Maedche, A., Motik, B., and Stojanovic, L. (2003). Managing multiple and distributed ontologies on the web. *VLDB Journal*, 12(4) :286–302.

[Magiridou et al., 2005] Magiridou, M., Sahtouris, S., Christophides, V., and Koubarakis, M. (2005). RUL : A Declarative Update Language for RDF. In *In Procs. 4th Intern. Conf. on the Semantic Web (ISWC-2005)*, pages 506–521.

[Marchiori, 1997] Marchiori, M. (1997). The quest for correct information on the web : hyper search engines. In *Proceedings of the sixth international conference on World Wide Web*, pages 1225–1235. Elsevier Science Publishers Ltd.

[Massa and Hayes, 2005] Massa, P. and Hayes, C. (2005). Page-reRank : Using Trusted Links to Re-Rank Authority. In *WI '05 : Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 614–617. IEEE Computer Society.

[Maynard et al., 2007] Maynard, D., Peters, W., d'Aquin, M., and Sabou, M. (2007). Change management for metadata evolution. In *Proceedings of the International Workshop on Ontology Dynamics (IWOD-07)*, pages 27–40.

[McGuinness and van Harmelen, 2004] McGuinness, D. and van Harmelen, F. (2004). OWL Web Ontology Language Overview. W3C Recommendation.

[Menzies, 1999] Menzies, T. (1999). Knowledge maintenance : the state of the art. *The Knowledge Engineering Review*, 14(1) :1–46.

[Menzies and Debenham, 2000] Menzies, T. and Debenham, J. (2000). Expert systems maintenance. In *Encyclopedia of Computer Science and Technology*, pages 35–54.

[Mezaour, 2003] Mezaour, A.-D. (2003). Focused search on the web using wequel. In *Proceedings of the 10th International Workshop on Knowledge Representation meets Databases (KRDB 2003)*, pages 63–74.

[Mizzaro and Tasso, 2002] Mizzaro, S. and Tasso, C. (2002). Ephemeral and persistent personalization in adaptive information access to scholarly publications on the web. In *AH '02 : Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 306–316. Springer-Verlag.

[Mostowfi and Fotouhi, 2006] Mostowfi, F. and Fotouhi, F. (2006). Improving quality of ontology : An ontology transformation approach. In *ICDEW '06 : Proceedings of the 22nd International Conference on Data Engineering Workshops*. IEEE Computer Society.

[Nauman and Khan, 2007] Nauman, M. and Khan, S. (2007). Using PersonalizedWeb Search for Enhancing Common Sense and Folksonomy Based Intelligent Search Systems. In *WI '07 : Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, pages 423–426. IEEE Computer Society.

[Navigli and Velardi, 2003] Navigli, R. and Velardi, P. (2003). An analysis of ontology-based query expansion strategies. In *Proceeding of the Workshop on Adaptive Text Extraction and Mining*, Cavtat-Dubrovnik (Croatia).

[NISO, 2004] NISO (2004). Understanding metadata. Technical report, NISO, Bethesda, MD.

[Noll and Meinel, 2007] Noll, M. G. and Meinel, C. (2007). Web search personalization via social bookmarking and tagging. In *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007*, LNCS, pages 367–380, Busan, South-Korea. Springer.

[Noy and McGuinness, 2001] Noy, N. and McGuinness, D. (2001). Ontology development 101 : A guide to creating your first ontology. Technical Report SMI-2001-0880, Stanford.

[Noy and Stuckenschmidt, 2005] Noy, N. and Stuckenschmidt, H. (2005). Ontology alignment : An annotated bibliography. In *Proceedings of the Dagstuhl Seminar on Semantic Interoperability and Integration.*

[Noy et al., 2006] Noy, N. F., Chugh, A., Liu, W., and Musen, M. A. (2006). A framework for ontology evolution in collaborative environments. In *Proceedings of the 5th International Semantic Web Conference (ISWC-06)*, LNCS, pages 544–558. Springer-Verlag.

[Noy and Klein, 2004] Noy, N. F. and Klein, M. (2004). Ontology evolution : Not the same as schema evolution. *Knowledge and Information Systems*, 6(4) :428–440.

[Noy and Musen, 2002] Noy, N. F. and Musen, M. A. (2002). Promptdiff : a fixed-point algorithm for comparing ontology versions. In *Eighteenth national conference on Artificial intelligence*, pages 744–750, Edmonton, Alberta, Canada. American Association for Artificial Intelligence.

[Oberle et al., 2004] Oberle, D., Volz, R., Motik, B., and Staab, S. (2004). An extensible ontology software environment. In Staab, S. and Studer, R., editors, *Handbook on Ontologies*, International Handbooks on Information Systems, chapter 3, pages 311–333.

[Oyama et al., 2004] Oyama, S., Kokubo, T., and Ishida, T. (2004). Domain-specific web search with keyword spices. *IEEE Trans. on Knowl. and Data Eng.*, 16 :17–27.

[Page and Brin, 1998] Page, L. and Brin, S. (1998). The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the Seventh International World-Wide Web Conference*, pages 107–117.

[Papamarkos et al., 2004] Papamarkos, G., Poulovassilis, A., and Wood, P. (2004). Rdftl : An event-condition-action language for rdf. In *Proc. 3rd Web Dynamics Workshop, at WWW'2004.*

[Papamarkos et al., 2003] Papamarkos, G., Poulovassilis, A., and Wood, P. T. (2003). Event-condition-action rule languages for the semantic web. In Cruz, I. F., Kashyap, V., Decker, S., and Eckstein, R., editors, *Proceedings of SWDB'03, The first International Workshop on Semantic Web and Databases*, pages 309–327.

[Patrânjan, 2005] Patrânjan, P.-L. (2005). *The Language XChange : A Declarative Approach to Reactivity on the Web.* PhD thesis, Ludwig-Maximilians-Universität München.

[Piaget, 1946] Piaget, J. (1946). *Les notions de mouvement et de vitesse chez l'enfant.* Paris : Presses universitaires de France.

[Piaget, 1974] Piaget, J. (1974). *La prise de conscience.* Paris : Presses Universitaires de France.

[Piaget, 1977] Piaget, J. (1977). *Recherches sur l'abstraction réfléchissante.* Paris : Presses universitaires de France.

[Pitkow et al., 2002] Pitkow, J. E., Schütze, H., Cass, T. A., Cooley, R., Turnbull, D., Edmonds, A., Adar, E., and Breuel, T. M. (2002). Personalized search. *Commun. ACM*, 45(9) :50–55.

[Plessers and De Troyer, 2005] Plessers, P. and De Troyer, O. (2005). Ontology change detection using a version log. In *Proceedings of the 4th International Semantic Web Conference*, LNCS, pages 578–592, Galway, Ireland.

[Plessers et al., 2007] Plessers, P., Troyer, O. D., and Casteleyna, S. (2007). Understanding ontology evolution : A change detection approach. *Web Semantics : Science, Services and Agents on the World Wide Web*, 5(1) :39–49.

[Pons and Keller, 1997] Pons, A. and Keller, R. K. (1997). Schema evolution in object databases by catalogs. In *IDEAS '97 : Proceedings of the 1997 International Symposium on Database Engineering & Applications*. IEEE Computer Society.

[Pruski, 2006] Pruski, C. (2006). Study of Knowledge Representation Languages and Knowledge Evolution. Technical report TR-LASSY-06-07, Laboratory for Advanced Software Systems, University of Luxembourg - Luxembourg Kirchberg.

[Qiu and Cho, 2006] Qiu, F. and Cho, J. (2006). Automatic identification of user interest for personalized search. In *WWW '06 : Proceedings of the 15th international conference on World Wide Web*, pages 727–736, Edinburgh, Scotland. ACM.

[Rahm and Bernstein, 2006] Rahm, E. and Bernstein, P. A. (2006). An online bibliography on schema evolution. *SIGMOD Rec.*, 35(4) :30–31.

[Ribeiro and Wassermann, 2007] Ribeiro, M. M. and Wassermann, R. (2007). Base revision in description logics - preliminary results. In *Proceedings of the International Workshop on Ontology Dynamics (IWOD 2007)*, pages 69–82, Innsbruck, Austria.

[Roddick, 1995] Roddick, J. F. (1995). A survey of schema versioning issues for database systems. *Information and Software Technology*, 37 :383–393.

[Rogozan, 2005] Rogozan, D. (2005). *Gestion de l'évolution d'une ontologie : méthodes et outils pour un référencement sémantique évolutif fondé sur une analyse des changements entre versions de l'ontologie*. PhD thesis, Télé-Université du Quebec.

[Ruthven and Lalmas, 2003] Ruthven, I. and Lalmas, M. (2003). A survey on the use of relevance feedback for information access systems. *Knowledge Engineering Review*, 18(2) :95–145.

[Schaffert, 2004] Schaffert, S. (2004). *Xcerpt : A Rule-Based Query and Transformation Language for the Web*. PhD thesis, Ludwig-Maximilians-Universität München.

[Shankaranarayanan and Ram, 2003] Shankaranarayanan, G. and Ram, S. (2003). Research issues in database schema evolution - the road not take. Technical Report 2003-15, The University of Arizona.

[Sieg et al., 2007] Sieg, A., Mobasher, B., and Burke, R. (2007). Ontological user profiles for personalized web search. In *Proceedings of the 5th Workshop on Intelligent Techniques for Web Personalization*.

[Silverstein et al., 1999] Silverstein, C., Marais, H., Henzinger, M., and Moricz, M. (1999). Analysis of a very large web search engine query log. *SIGIR Forum*, 33(1) :6–12.

[Sindt, 2003] Sindt, T. (2003). Formal operations for ontology evolution. In *Proceedings of the International Conference on Emerging Technologies, ICET'03*.

[Sizikov and Soshnikov, 2002] Sizikov, E. and Soshnikov, D. (2002). Using dynamic ontologies based on production-frame knowledge representation for intelligent web retrieval. In *Proceedings of the 4th International Workshop on Computer Science and Information Technologies*, Patras, Greece.

[Song et al., 2006] Song, M., Song, I.-Y., Hu, X., and Allen, R. B. (2006). Integration of association rules and ontology for semantic-based query expansion. *Data and Knowledge Engineering*.

[Sowa, 1984] Sowa, J. F. (1984). *Conceptual Structures - Information Processing in Mind and Machine*. Addison Wesley.

[Specia and Motta, 2007] Specia, L. and Motta, E. (2007). Integrating folksonomies with the semantic web. In *The Semantic Web : Research and Applications, 4th European Semantic Web Conference, ESWC 2007*, LNCS, pages 624–639, Innsbruck, Austria. Springer.

[Spertus and Stein, 2000] Spertus, E. and Stein, L. A. (2000). Squeal : Structured queries on the web. In *Ninth International World-Wide Web Conference*, Amsterdam.

[Stojanovic, 2004] Stojanovic, L. (2004). *Methods and Tools for Ontology Evolution*. PhD thesis, University of Karlsruhe, Universität Karlsruhe (TH), Institut AIFB, D-76128 Karlsruhe.

[Stojanovic et al., 2002] Stojanovic, L., Maedche, A., Motik, B., and Stojanovic, N. (2002). User-driven ontology evolution management. In *In European Conf. Knowledge Eng. and Management (EKAW 2002)*, pages 285–300. Springer-Verlag.

[Stojanovic et al., 2004] Stojanovic, N., Studer, R., and Stojanovic, L. (2004). An approach for step-by-step query refinement in the ontology-based information retrieval. In *IEEE/WIC/ACM International Conference on Web Intelligence (WI'04)*, pages 36–43.

[Stuckenschmidt and Klein, 2003] Stuckenschmidt, H. and Klein, M. (2003). Integrity and Change in Modular Ontologies. In Gottlob, G. and Walsh, T., editors, *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 900–908. Morgan Kaufmann.

[Tamine et al., 2007] Tamine, L., Zemirli, W. N., and Bahsoun, W. (2007). Approche statistique pour la définition du profil d'un utilisateur de système de recherche d'information. *Information - Interaction - Intelligence*, 7(1).

[Tanudjaja and Mui, 2002] Tanudjaja, F. and Mui, L. (2002). Persona : A contextualized and personalized web search. In *HICSS '02 : Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02)*, volume 3. IEEE Computer Society.

[Teevan et al., 2005] Teevan, J., Dumais, S. T., and Horvitz, E. (2005). Personalizing search via automated analysis of interests and activities. In *SIGIR '05 : Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 449–456, Salvador, Brazil. ACM.

[Tombros and Sanderson, 1998] Tombros, A. and Sanderson, M. (1998). Advantages of query biased summaries in information retrieval. In *SIGIR '98 : Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2–10. ACM.

[Trinkunas and Vasilecas, 2007] Trinkunas, J. and Vasilecas, O. (2007). A graph oriented model for ontology transformation into conceptual data model. *Information Technology And Control*, 36(1) :126–132.

[Troncy and Isaac, 2002] Troncy, R. and Isaac, A. (2002). DOE : une mise en oeuvre d'une méthode de structuration différentielle pour les ontologies. In *Proceedings of the 13ièmes Journées Francophones d'Ingénierie des Connaissances, IC'2002*, Rouen, France.

[Uschold and King, 1995] Uschold, M. and King, M. (1995). Towards a methodology for building ontologies. In *Workshop on Basic Ontological Issues in Knowledge Sharing, held in conduction with IJCAI-95.*

[Vallet et al., 2007] Vallet, D., Castells, P., Fernandez, M., Mylonas, P., and Avrithis, Y. (2007). Personalized content retrieval in context using ontological knowledge. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(3) :336–346.

[Velardi et al., 2006] Velardi, P., Navigli, R., Cucchiarelli, A., and Neri, F. (2006). Evaluation of OntoLearn, a methodology for automatic population of domain ontologies. In Buitelaar, P., Cimiano, P., and Magnini, B., editors, *Ontology learning from text : Methods, Applications and Evaluation*. IOS Press.

[Völkel and Groza, 2006] Völkel, M. and Groza, T. (2006). SemVersion : An RDF-based Ontology Versioning System. In *Proceedings of IADIS International Conference on WWW/Internet*, volume 1, pages 195–202, Murcia, Spain.

[Voorhees, 1994] Voorhees, E. M. (1994). Query expansion using lexical-semantic relations. In *SIGIR '94 : Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 61–69, Dublin, Ireland. Springer-Verlag New York, Inc.

[Weichselbraun et al., 2007] Weichselbraun, A., Scharl, A., Liu, W., and Wohlgenannt, G. (2007). Capturing ontology evolution processes by repeated sampling of large document collections. In Heidelberg, S. B. ., editor, *On the Move to Meaningful Internet Systems 2007 : OTM 2007 Workshops*, volume 4805/2007 of *Lecture Notes in Computer Science*, pages 23–24.

[Wu, 2002] Wu, H. (2002). *A reference architecture for adaptive hypermedia applications.* PhD thesis, Technische Universiteit Eindhoven.

[Xu et al., 2008] Xu, S., Bao, S., Fei, B., Su, Z., and Yu, Y. (2008). Exploring folksonomy for personalized search. In *SIGIR '08 : Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 155–162, Singapore, Singapore. ACM.

[Yong and Guan-yu, 2007] Yong, L. and Guan-yu, L. (2007). Research and realization of personalized search engine based on ontology. In *NPC '07 : Proceedings of the 2007 IFIP International Conference on Network and Parallel Computing Workshops*, pages 1016–1020. IEEE Computer Society.

[Yuwono and Lee, 1996] Yuwono, B. and Lee, D. L. (1996). Search and Ranking Algorithms for Locating Resources on the World Wide Web. In *ICDE '96 : Proceedings of the Twelfth International Conference on Data Engineering*, pages 164–171. IEEE Computer Society.

[Zeginis et al., 2007] Zeginis, D., Tzitzikas, Y., and Christophides, V. (2007). On the Foundations of Computing Deltas Between RDF Models. In *Proceedings of the 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007*, LNCS, pages 637–651, Busan, South-Korea. Springer-Verlag.

[Zhuang and Cucerzan, 2006] Zhuang, Z. and Cucerzan, S. (2006). Re-ranking search results using query logs. In *CIKM '06 : Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 860–861, Arlington, Virginia, USA. ACM.

[Zweigenbaum et al., 1995] Zweigenbaum, P., Bachimont, B., Bouaud, J., Charlet, J., and Boisivieux, J.-F. (1995). Issues in the structuring and acquisition of an ontology for medical language understanding. *Information in Medicine*, 34(1/2).

Annexes

Annexe A : Sémantique de OWL

Elément OWL	**Syntaxe abstraite**	**DL**	**Sémantique**
owl :Class	Class(A partial $C_1 \dots C_n$)	$A \sqsubseteq C_1 \sqcap \dots \sqcap C_n$	$A^{\mathcal{I}} \subseteq C_1^{\mathcal{I}} \cap \dots \cap C_n^{\mathcal{I}}$
	Class(A complete $C_1 \dots C_n$)	$A = C_1 \sqcap \dots \sqcap C_n$	$A^{\mathcal{I}} = C_1^{\mathcal{I}} \cap \dots \cap C_n^{\mathcal{I}}$
rdfs :subClassOf	SubClassOf(C_1 C_2)	$C_1 \sqsubseteq C_2$	$C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$
rdfs :subPropertyOf	SubPropertyOf(D_1 D_2)	$O_1 \sqsubseteq D_2$	$D_1^{\mathcal{I}} \subseteq D_2^{\mathcal{I}}$
owl :equivalentClass	EquivalentClasses($C_1 \dots C_n$)	$C_1 = \dots = C_n$	$C_1^{\mathcal{I}} = \dots = C_n^{\mathcal{I}}$
owl :disjointWith	DisjointClasses($C_1 \dots C_n$)	$C_i \sqcap C_j = \bot, i \neq j$	$C_i^{\mathcal{I}} \cap C_j^{\mathcal{I}} = \emptyset, i \neq j$
owl :equivalentProperty	EquivalentProperties($D_1 \dots D_n$)	$D_1 = \dots = D_n$	$D_1^{\mathcal{I}} = \dots = D_n^{\mathcal{I}}$
owl :sameAs owl :sameIndividual	SameAs($I_1 \dots I_n$)	$I_1 = \dots = I_n$	$I_1^{\mathcal{I}} = \dots = I_n^{\mathcal{I}}$
owl :differentFrom owl :allDifferent owl :differentIndividual owl :distinctMembers	DifferentFrom($I_1 \dots I_n$)	$I_i \neq I_j, i \neq j$	$I_i^{\mathcal{I}} \neq I_j^{\mathcal{I}}, i \neq j$
owl :ObjectProperty	ObjProp(O super(O_1)...super(O_n)	$O \sqsubseteq O_i$	$O^{\mathcal{I}} \subseteq O_i^{\mathcal{I}}$
	domain(C_1)...domain(C_m)	$\geq 1 O \sqsubseteq C_i$	$O^{\mathcal{I}} \subseteq C_i^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
	range(C_1)...range(C_l)	$\top \sqsubseteq \forall O.C_i$	$O^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times C_i^{\mathcal{I}}$
	[InverseOf(O_0)]	$O = (^-O_0)$	$O^{\mathcal{I}} = (O_0^{\mathcal{I}})^-$
	[Symmetric]	$O = (^-O)$	$O^{\mathcal{I}} = (O^{\mathcal{I}})^-$
	[Functional]	$\top \sqsubseteq \leq 1\ O$	$O^{\mathcal{I}}$ is functional
	[InverseFunctional]	$\top \sqsubseteq \leq 1\ O^-$	$(O^{\mathcal{I}})^-$ is functional
	[Transitive]	$Tr(O)$	$O^{\mathcal{I}} = (O^{\mathcal{I}})^+$
owl :DatatypeProperty	DatProp(D super(D_1)...super(D_n)	$D \sqsubseteq D_i$	$D^{\mathcal{I}} \subseteq D_i^{\mathcal{I}}$

	domain(C_1)...domain(C_m) range(C_1)...range(C_l) [Functional]	$\geq 1\ D \sqsubseteq C_i$ $\top \sqsubseteq \forall D.C_i$ $\top \sqsubseteq \leq 1\ D$	$D^{\mathcal{I}} \subseteq D_i^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ $D^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times C_i^{\mathcal{I}}$ $D^{\mathcal{I}}$ is functional
owl :someValuesFrom	$restriction(O\ someValuesFrom(C))$ $restriction(D\ someValuesFrom(R))$	$\exists O.C$ $\exists D.R$	$(\exists O.C)^{\mathcal{I}} = \{x \mid \exists y. < x,y > \in O^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$ $(\exists D.R)^{\mathcal{I}} = \{x \mid \exists y. < x,y > \in D^{\mathcal{I}} \wedge y \in R^R\}$
owl :allValuesFrom	$restriction(O\ allValuesFrom(C))$ $restriction(D\ allValuesFrom(R))$	$\forall O.C$ $\forall D.R$	$(\forall O.C)^{\mathcal{I}} = \{x \mid \forall y. < x,y > \in O^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$ $(\forall D.R)^{\mathcal{I}} = \{x \mid \forall y. < x,y > \in D^{\mathcal{I}} \rightarrow y \in R^R\}$
owl :hasValue	$restriction(O\ hasValue(I))$ $restriction(D\ hasValue(V))$	$O : I$ $D : V$	$(\forall O.I)^{\mathcal{I}} = \{x \mid < x, I^{\mathcal{I}} > \in O^{\mathcal{I}}\}$ $(D : V)^{\mathcal{I}} = \{x \mid < x, V^{\mathcal{I}} > \in D^{\mathcal{I}}\}$
owl :minCardinality	$restriction(O\ minCardinality(n))$ $restriction(D\ minCardinality(n))$	$\geq n\ O$ $\geq n\ D$	$(\geq n\ O)^{\mathcal{I}} = \{x \mid \sharp(\{y. < x,y > \in O^{\mathcal{I}}\}) \geq n\}$ $(\geq n\ D)^{\mathcal{I}} = \{x \mid \sharp(\{y. < x,y > \in D^{\mathcal{I}}\}) \geq n\}$
owl :maxCardinality	$restriction(O\ maxCardinality(n))$ $restriction(D\ maxCardinality(n))$	$\leq n\ O$ $\leq n\ D$	$(\leq n\ O)^{\mathcal{I}} = \{x \mid \sharp(\{y. < x,y > \in O^{\mathcal{I}}\}) \leq n\}$ $(\leq n\ D)^{\mathcal{I}} = \{x \mid \sharp(\{y. < x,y > \in D^{\mathcal{I}}\}) \leq n\}$
owl :intersectionOf	IntersectionOf(C_1, C_2)	$C_1 \sqcap C_2$	$(C_1 \sqcap C_2)^{\mathcal{I}} = C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}$
owl :unionOf	UnionOf(C_1, C_2)	$C_1 \sqcup C_2$	$(C_1 \sqcup C_2)^{\mathcal{I}} = C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}}$
owl :complementOf	ComplementOf(C)	$\neg C$	$\Delta^{\mathcal{I}} \backslash C^{\mathcal{I}}$
owl :oneOf	OneOf($I_1 \ldots$)	$\{I_1, \ldots\}$	$(\{I_1, \ldots\})^{\mathcal{I}} = \{I_1^{\mathcal{I}} \ldots\}$

TABLE 2: Syntaxe et sémantique de OWL

Annexe B : Ontologie adaptative du domaine WWW

Cette partie contient le code OWL de l'ontologie adaptative du domaine WWW.

```
<?xml version="1.0"?>
<rdf:RDF xmlns="http://www.owl-ontologies.com/unnamed.owl#"
     xml:base="http://www.owl-ontologies.com/unnamed.owl"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:owl="http://www.w3.org/2002/07/owl#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
    <owl:Ontology rdf:about=""/>
    <owl:Class rdf:ID="addressing">
        <rdfs:subClassOf rdf:resource="#server_technology"/>
        <Semantic_Distance rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
           8.0
        </Semantic_Distance>
        <Emergence_Date rdf:datatype="http://www.w3.org/2001/XMLSchema#date">
           1997-01-01
        </Emergence_Date>
        <Persistence_Duration
rdf:datatype="http://www.w3.org/2001/XMLSchema#duration">
           1
        </Persistence_Duration>
        <Resistance rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
           1.0
        </Resistance>
        <Semantic_Weight rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
           1.0
        </Semantic_Weight>
    </owl:Class>
    <owl:Class rdf:ID="application">
        <Emergence_Date rdf:datatype="http://www.w3.org/2001/XMLSchema#date">
           1997-01-01
        </Emergence_Date>
        <Persistence_Duration
rdf:datatype="http://www.w3.org/2001/XMLSchema#duration">
           1
        </Persistence_Duration>
        <Semantic_Weight rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
           1.0
        </Semantic_Weight>
    </owl:Class>
    <owl:Class rdf:ID="authoring_environment">
        <Emergence_Date rdf:datatype="http://www.w3.org/2001/XMLSchema#date">
           1997-01-01
        </Emergence_Date>
        <Persistence_Duration
```

```
rdf:datatype="http://www.w3.org/2001/XMLSchema#duration">
           1
        </Persistence_Duration>
        <Semantic_Weight rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
           3.0
        </Semantic_Weight>
    </owl:Class>
    <owl:Class rdf:ID="authorization">
        <rdfs:subClassOf rdf:resource="#security"/>
        <Semantic_Distance rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
           9.0
        </Semantic_Distance>
        <Emergence_Date rdf:datatype="http://www.w3.org/2001/XMLSchema#date">
           1997-01-01
        </Emergence_Date>
        <Persistence_Duration
rdf:datatype="http://www.w3.org/2001/XMLSchema#duration">
           1
        </Persistence_Duration>
        <Resistance rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
           1.0
        </Resistance>
        <Semantic_Weight rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
           3.0
        </Semantic_Weight>
    </owl:Class>
    <owl:Class rdf:ID="browser">
        <rdfs:subClassOf rdf:resource="#application"/>
        <Semantic_Distance rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
           2.0
        </Semantic_Distance>
        <Emergence_Date rdf:datatype="http://www.w3.org/2001/XMLSchema#date">
           1997-01-01
        </Emergence_Date>
        <Persistence_Duration
rdf:datatype="http://www.w3.org/2001/XMLSchema#duration">
           6
        </Persistence_Duration>
        <Resistance rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
           10.0
        </Resistance>
        <Semantic_Weight rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
           3.0
        </Semantic_Weight>
    </owl:Class>
    <owl:Class rdf:ID="caching">
        <rdfs:subClassOf rdf:resource="#server_technology"/>
        <Semantic_Distance rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
```

```
            7.0
        </Semantic_Distance>
        <Emergence_Date rdf:datatype="http://www.w3.org/2001/XMLSchema#date">
            1997-01-01
        </Emergence_Date>
        <Persistence_Duration
rdf:datatype="http://www.w3.org/2001/XMLSchema#duration">
            4
        </Persistence_Duration>
        <Resistance rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
            1.0
        </Resistance>
        <Semantic_Weight rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
            3.0
        </Semantic_Weight>
    </owl:Class>
    <owl:Class rdf:ID="cultural">
        <Emergence_Date rdf:datatype="http://www.w3.org/2001/XMLSchema#date">
            1997-01-01
        </Emergence_Date>
        <Persistence_Duration
rdf:datatype="http://www.w3.org/2001/XMLSchema#duration">
            1
        </Persistence_Duration>
        <Semantic_Weight rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
            1.0
        </Semantic_Weight>
    </owl:Class>
    <owl:Class rdf:ID="educationnal_application">
        <rdfs:subClassOf rdf:resource="#application"/>
        <Semantic_Distance rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
            7.0
        </Semantic_Distance>
        <Emergence_Date rdf:datatype="http://www.w3.org/2001/XMLSchema#date">
            1997-01-01
        </Emergence_Date>
        <Persistence_Duration
rdf:datatype="http://www.w3.org/2001/XMLSchema#duration">
            1
        </Persistence_Duration>
        <Resistance rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
            1.0
        </Resistance>
        <Semantic_Weight rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
            1.0
        </Semantic_Weight>
    </owl:Class>
    <owl:DatatypeProperty rdf:ID="Emergence_Date">
```

```
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#AnnotationProperty"/>
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#date"/>
    </owl:DatatypeProperty>
    <owl:ObjectProperty rdf:ID="function_of">
        <rdfs:domain rdf:resource="#authorization"/>
        <rdfs:range rdf:resource="#authoring_environment"/>
        <Semantic_Distance rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
           9.0
        </Semantic_Distance>
        <Resistance rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
           1.0
        </Resistance>
    </owl:ObjectProperty>
    <owl:Class rdf:ID="hypermedia">
        <rdfs:subClassOf rdf:resource="#multimedia"/>
        <Semantic_Distance rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
           5.0
        </Semantic_Distance>
        <Emergence_Date rdf:datatype="http://www.w3.org/2001/XMLSchema#date">
           1997-01-01
        </Emergence_Date>
        <Persistence_Duration
rdf:datatype="http://www.w3.org/2001/XMLSchema#duration">
           3
        </Persistence_Duration>
        <Resistance rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
           5.0
        </Resistance>
        <Semantic_Weight rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
           3.0
        </Semantic_Weight>
    </owl:Class>
    <owl:Class rdf:ID="hypertext">
        <rdfs:subClassOf rdf:resource="#hypermedia"/>
        <Semantic_Distance rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
           5.0
        </Semantic_Distance>
        <Emergence_Date rdf:datatype="http://www.w3.org/2001/XMLSchema#date">
           1997-01-01
        </Emergence_Date>
        <Persistence_Duration
rdf:datatype="http://www.w3.org/2001/XMLSchema#duration">
           3
        </Persistence_Duration>
        <Resistance rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
           1.0
        </Resistance>
        <Semantic_Weight rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
```

```
            3.0
        </Semantic_Weight>
    </owl:Class>
    <owl:Class rdf:ID="indexing">
        <rdfs:subClassOf rdf:resource="#search_technique"/>
        <Semantic_Distance rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
            6.0
        </Semantic_Distance>
        <Emergence_Date rdf:datatype="http://www.w3.org/2001/XMLSchema#date">
            1997-01-01
        </Emergence_Date>
        <Persistence_Duration
rdf:datatype="http://www.w3.org/2001/XMLSchema#duration">
            3
        </Persistence_Duration>
        <Resistance rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
            2.0
        </Resistance>
        <Semantic_Weight rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
            3.0
        </Semantic_Weight>
    </owl:Class>
    <owl:Class rdf:ID="IR_technique">
        <owl:equivalentClass rdf:resource="#search_technique"/>
        <Semantic_Distance rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
            5.0
        </Semantic_Distance>
        <Emergence_Date rdf:datatype="http://www.w3.org/2001/XMLSchema#date">
            1997-01-01
        </Emergence_Date>
        <Persistence_Duration
rdf:datatype="http://www.w3.org/2001/XMLSchema#duration">
            2
        </Persistence_Duration>
        <Resistance rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
            1.0
        </Resistance>
        <Semantic_Weight rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
            3.0
        </Semantic_Weight>
    </owl:Class>
    <owl:ObjectProperty rdf:ID="is_useful_for">
        <rdfs:domain rdf:resource="#markup_language"/>
        <rdfs:range rdf:resource="#metadata_system"/>
        <Semantic_Distance rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
            8.0
        </Semantic_Distance>
        <Resistance rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
```

```
            1.0
        </Resistance>
    </owl:ObjectProperty>
    <owl:Class rdf:ID="language">
        <Emergence_Date rdf:datatype="http://www.w3.org/2001/XMLSchema#date">
            1997-01-01
        </Emergence_Date>
        <Persistence_Duration
rdf:datatype="http://www.w3.org/2001/XMLSchema#duration">
            1
        </Persistence_Duration>
        <Semantic_Weight rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
            1.0
        </Semantic_Weight>
    </owl:Class>
    <owl:Class rdf:ID="markup_language">
        <rdfs:subClassOf rdf:resource="#language"/>
        <Semantic_Distance rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
            6.0
        </Semantic_Distance>
        <Emergence_Date rdf:datatype="http://www.w3.org/2001/XMLSchema#date">
            1997-01-01
        </Emergence_Date>
        <Persistence_Duration
rdf:datatype="http://www.w3.org/2001/XMLSchema#duration">
            1
        </Persistence_Duration>
        <Resistance rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
            1.0
        </Resistance>
        <Semantic_Weight rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
            3.0
        </Semantic_Weight>
    </owl:Class>
    <owl:Class rdf:ID="metadata_system">
        <Emergence_Date rdf:datatype="http://www.w3.org/2001/XMLSchema#date">
            1997-01-01
        </Emergence_Date>
        <Persistence_Duration
rdf:datatype="http://www.w3.org/2001/XMLSchema#duration">
            4
        </Persistence_Duration>
        <Semantic_Weight rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
            3.0
        </Semantic_Weight>
    </owl:Class>
    <owl:Class rdf:ID="multimedia">
        <Emergence_Date rdf:datatype="http://www.w3.org/2001/XMLSchema#date">
```

```
            1997-01-01
        </Emergence_Date>
        <Persistence_Duration
rdf:datatype="http://www.w3.org/2001/XMLSchema#duration">
            2
        </Persistence_Duration>
        <Semantic_Weight rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
            3.0
        </Semantic_Weight>
    </owl:Class>
    <owl:Class rdf:ID="naming">
        <rdfs:subClassOf rdf:resource="#server_technology"/>
        <Semantic_Distance rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
            9.0
        </Semantic_Distance>
        <Emergence_Date rdf:datatype="http://www.w3.org/2001/XMLSchema#date">
            1997-01-01
        </Emergence_Date>
        <Persistence_Duration
rdf:datatype="http://www.w3.org/2001/XMLSchema#duration">
            1
        </Persistence_Duration>
        <Resistance rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
            1.0
        </Resistance>
        <Semantic_Weight rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
            1.0
        </Semantic_Weight>
    </owl:Class>
    <owl:DatatypeProperty rdf:ID="Persistence_Duration">
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#AnnotationProperty"/>
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#duration"/>
    </owl:DatatypeProperty>
    <owl:Class rdf:ID="programming_language">
        <rdfs:subClassOf rdf:resource="#language"/>
        <Semantic_Distance rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
            4.0
        </Semantic_Distance>
        <Emergence_Date rdf:datatype="http://www.w3.org/2001/XMLSchema#date">
            1997-01-01
        </Emergence_Date>
        <Persistence_Duration
rdf:datatype="http://www.w3.org/2001/XMLSchema#duration">
            3
        </Persistence_Duration>
        <Resistance rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
            1.0
        </Resistance>
```

```
        <Semantic_Weight rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
           3.0
        </Semantic_Weight>
    </owl:Class>
    <owl:Class rdf:ID="push_technology">
        <rdfs:subClassOf rdf:resource="#server_technology"/>
        <Semantic_Distance rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
           7.0
        </Semantic_Distance>
        <Emergence_Date rdf:datatype="http://www.w3.org/2001/XMLSchema#date">
           1997-01-01
        </Emergence_Date>
        <Persistence_Duration
rdf:datatype="http://www.w3.org/2001/XMLSchema#duration">
           1
        </Persistence_Duration>
        <Resistance rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
           1.0
        </Resistance>
        <Semantic_Weight rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
           1.0
        </Semantic_Weight>
    </owl:Class>
    <owl:Class rdf:ID="replication">
        <rdfs:subClassOf rdf:resource="#server_technology"/>
        <Semantic_Distance rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
           9.0
        </Semantic_Distance>
        <Emergence_Date rdf:datatype="http://www.w3.org/2001/XMLSchema#date">
           1997-01-01
        </Emergence_Date>
        <Persistence_Duration
rdf:datatype="http://www.w3.org/2001/XMLSchema#duration">
           4
        </Persistence_Duration>
        <Resistance rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
           1.0
        </Resistance>
        <Semantic_Weight rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
           3.0
        </Semantic_Weight>
    </owl:Class>
    <owl:DatatypeProperty rdf:ID="Resistance">
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#AnnotationProperty"/>
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
    </owl:DatatypeProperty>
    <owl:Class rdf:ID="search_technique">
        <owl:equivalentClass rdf:resource="#IR_technique"/>
```

```
        <Emergence_Date rdf:datatype="http://www.w3.org/2001/XMLSchema#date">
           1997-01-01
        </Emergence_Date>
        <Persistence_Duration
rdf:datatype="http://www.w3.org/2001/XMLSchema#duration">
           6
        </Persistence_Duration>
        <Semantic_Weight rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
           3.0
        </Semantic_Weight>
    </owl:Class>
    <owl:Class rdf:ID="security">
        <Emergence_Date rdf:datatype="http://www.w3.org/2001/XMLSchema#date">
           1997-01-01
        </Emergence_Date>
        <Persistence_Duration
rdf:datatype="http://www.w3.org/2001/XMLSchema#duration">
           6
        </Persistence_Duration>
        <Semantic_Weight rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
           3.0
        </Semantic_Weight>
    </owl:Class>
    <owl:DatatypeProperty rdf:ID="Semantic_Distance">
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#AnnotationProperty"/>
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
    </owl:DatatypeProperty>
    <owl:DatatypeProperty rdf:ID="Semantic_Weight">
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#AnnotationProperty"/>
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
    </owl:DatatypeProperty>
    <owl:Class rdf:ID="server_technology">
        <Emergence_Date rdf:datatype="http://www.w3.org/2001/XMLSchema#date">
           1997-01-01
        </Emergence_Date>
        <Persistence_Duration
rdf:datatype="http://www.w3.org/2001/XMLSchema#duration">
           1
        </Persistence_Duration>
        <Semantic_Weight rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
           3.0
        </Semantic_Weight>
    </owl:Class>
    <owl:Class rdf:ID="social">
        <Emergence_Date rdf:datatype="http://www.w3.org/2001/XMLSchema#date">
           1997-01-01
        </Emergence_Date>
        <Persistence_Duration
```

```
rdf:datatype="http://www.w3.org/2001/XMLSchema#duration">
           1
        </Persistence_Duration>
        <Semantic_Weight rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
           1.0
        </Semantic_Weight>
    </owl:Class>
    <owl:Class rdf:ID="tool">
        <rdfs:subClassOf rdf:resource="#application"/>
        <Semantic_Distance rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
           2.0
        </Semantic_Distance>
        <Emergence_Date rdf:datatype="http://www.w3.org/2001/XMLSchema#date">
           1997-01-01
        </Emergence_Date>
        <Persistence_Duration
rdf:datatype="http://www.w3.org/2001/XMLSchema#duration">
           365
        </Persistence_Duration>
        <Resistance rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
           1.0
        </Resistance>
        <Semantic_Weight rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
           3.0
        </Semantic_Weight>
    </owl:Class>
</rdf:RDF>
```

Annexe C : Ontologie adaptative représentant la catégorie des chercheurs appliqués

Cette partie contient l'ontologie de la catégorie des chercheurs appliqués par opposition aux chercheurs fondamentaux dont l'ontologie est représentée à la figure 4.5.

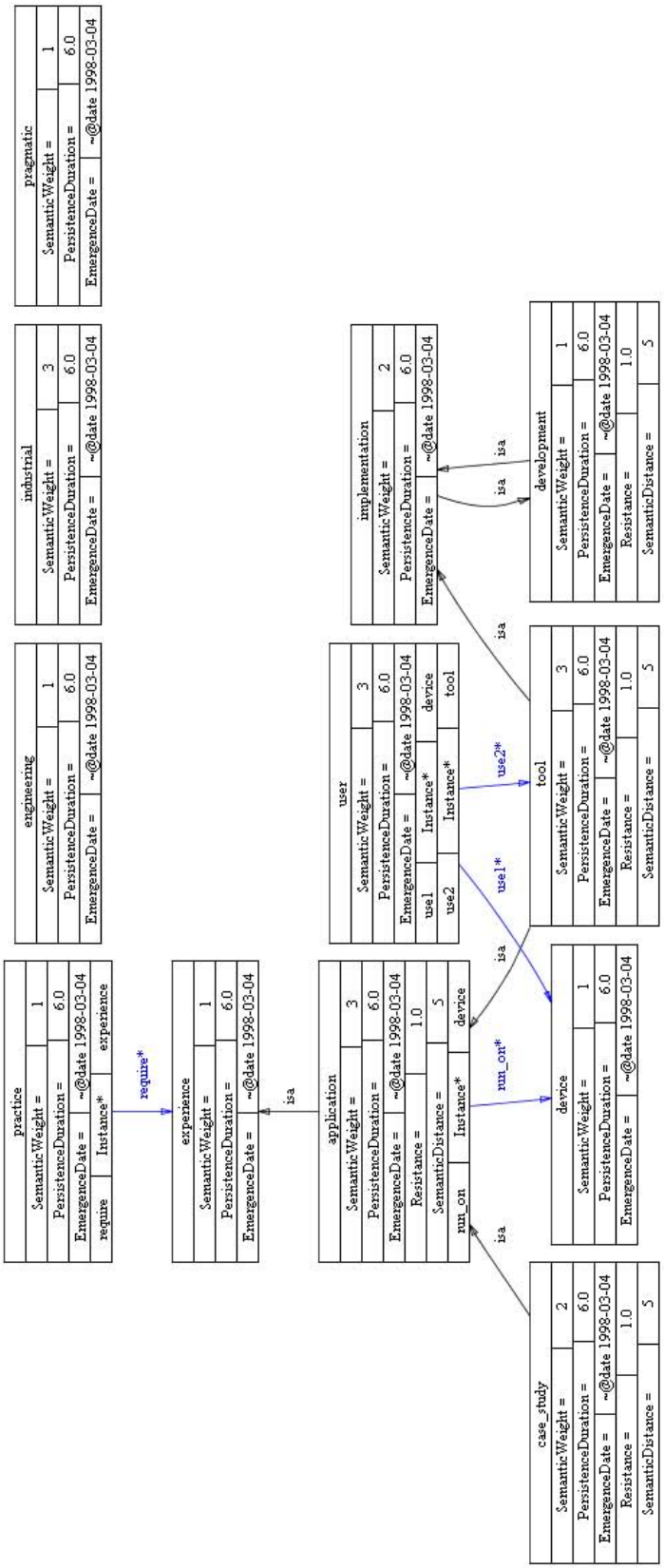


FIGURE 10 – Catégorie des chercheurs appliqués