Analyses formelle et relationnelle de concepts pour la construction d'ontologies de domaines à partir de ressources textuelles hétérogènes

Rokia Bendaoud

Département de formation doctorale en informatique **École doctorale IAEM Lorraine**
UFR STMIA

Analyses formelle et relationnelle de concepts pour la construction d'ontologies de domaines à partir de ressources textuelles hétérogènes

THÈSE

présentée et soutenue publiquement le 2009

pour l'obtention du

Doctorat de l'université Henri Poincaré – Nancy 1

(spécialité informatique)

par

Rokia Bendaoud

Composition du jury

Rapporteurs :	Pierre Zweigenbaum	Directeur de Recherche, CNRS, ORSAY
	François Jacquenet	Professeur, Université de Saint-Etienne
Examinateurs :	Jean-Marie Pierrel	Professeur, l'Université Henri Poincaré
	Karell Bertet	Maitre de Conférences, Université de La Rochelle
	Amedeo Napoli	Directeur de recherche, CNRS, Nancy
	Yannick Toussaint	Chargé de recherche, INRIA, Nancy

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

Mis en page avec la classe thloria.

Remerciements

Je tiens à remercier en tout premier lieu Amedeo Napoli, qui a accepté d'être mon directeur de thèse. Je lui suis particulièrement reconnaissante pour ses conseils méthodologiques et ses qualités scientifiques qui m'ont été très précieux. Je le remercie aussi de m'avoir acceuillis au sein de l'équipe Orpailleur.

Merci à Yannick Toussaint, qui a co-encadré cette thèse pour ses encouragements, son soutien, son aide pour mes problèmes administratifs ainsi que pour toute l'attention qu'il a porté à mon travail.

Merci à Jean-Marie Pierrel de m'avoir fait l'honneur de présider le jury de thèse. A François Jacquenet et à Pierre Zweigenbaum pour avoir accepté de rapporter cette thèse. Un grand merci à Karell Bertet et à Pierre Zweigenbaum pour leurs nombreux commentaires très pertinents qui ont permis d'améliorer ce mémoire.

Je tiens aussi à remercier mes experts : Merci aux astronomes du laboratoire d'astronomie de Strasbourg : Andrea Preite-Martinez, Pascal Dubois et Soizick Lesteven pour m'avoir fourni le corpus pour mes expérimentations et pour avoir analysé les résultats. Merci aussi aux membres de l'INIST : Claire François, Francoise Tisserand-Bedri et Bernard Taliercio pour avoir accepté d'expertiser mon travail en microbiologie et de m'avoir accordé plusieurs entretiens.

Aux membres de l'équipe Orpailleur : Nizar, Fadi, Julien, Zaineb, Adrien, Laszlo et tout particulièrement au meilleur camarade de bureau Jean-François Mari. Merci aussi à Marie-Dominique Devignes, Malika Smail-Tabbone et Vincent Leroux pour avoir répondu à mes questions en microbiologie et en chimie en m'expliquant la différence entre gène, gènome et protéine... A Bertrand Delecroix, pour sa collaboration, son aide précieuse et ses conseils professionnels et personnels :)

A mes amis par ordre alphabétique : à ma filleule de coeur Ekatharina et à sa maman Alex. A Hana, ma petite soeur que j'adore. A ma meilleure amie Hanane. A Hejer pour son soutien et ses papotages pendant des heures. A mon italien préféré Ignazio pour m'avoir écouté, conseillé et soutenu. A mon ami Nadjib pour les sorties à Alger et les fous rires sur le net. A Najouta, une amir qui a toujours était présente dans les moments de doute. A Rmikia, ses histoires farfelues et ses décisions radicales. A ma binome de coeur Samira. A ma chère amie Sandrine qui m'a acceuilli en france, m'a appris la diversité culturelle et l'ouverture d'esprit. A mon voisin, ami, collegue et confident Stéphane. A Sawsan, pour m'avoir ouvert la porte de sa famille avec ses adorables enfants : Nasser, Raheeg et Fatouma, son mari et de m'avoir fait découvrir «El Mansef». Et enfin, à Yves et Hatem pour les repas amusants, les jeux et la mauvaise foie!!!

A ma famille et plus particulièrement à mes chers parents, qui n'ont cessé de m'encourager et de me remonter le moral. A mon frère Abder et à sa petite princesse Inès, à mon petit frère Sofiane et ses coups de fils amusants et encourageants et à ma soeur Zohra et à mes deux anges Chakibo et Momo. A mon cousin Djilali pour ses chansons et ses papotages.

Aux familles Meilender et Ferry pour m'avoir acceuilli aussi gentillement dans leur famille. A mon Tom, merci d'avoir toujours été là.

A mes parents et à Tom.

Table des matières

Table des figures

Chapitre 1

Introduction générale

1.1 Contexte de travail

La construction d'un modèle pour un domaine spécifique qu'on souhaite étudier consiste à regrouper (collecter) toutes les connaissances du domaine extraites à partir de toutes les ressources dont dispose ce domaine. Cette définition du modèle correspond à celle d'une *ontologie* (ou *base de connaissances*) en informatique. Une ontologie organise les connaissances en hiérarchies de concepts et en relations entre ces concepts, afin de les gérer, de les diffuser, de les partager, de les utiliser et de les faire évoluer. Les connaissances sont des unités qui permettent de raisonner. Raisonner veut dire exploiter des mécanismes d'induction ou de déduction, c'est ce type de raisonnement qui est utilisé par les humains. Pour l'implémenter aux machines, tout un domaine a vu le jour en informatique : l'Intelligence Artificielle (IA).

En IA, un système à base de connaissances est un système capable d'utiliser des connaissances pour raisonner et proposer des solutions à des problèmes tels que des problèmes d'aide à la décision d'experts ou des problèmes de traitement automatique de la langue naturelle [Cornuéjols A., 2003; Russell et Norvig, 2003]. Néanmoins, les machines manipulent des symboles et elles ont besoin d'instructions claires sur la façon de les manipuler. C'est pourquoi les modèles de domaine (les ontologies) sont définis de manière formelle, avec des procédures pour vérifier sémantiquement ces symboles et ainsi répondre à des questions complexes d'experts.

Nous considérons qu'une ressource est tout ce qui contient une donnée susceptible d'être transformée en une connaissance. Les ressources d'un domaine peuvent prendre la forme de bases de données, de dictionnaires, de thésaurus ou de corpus de textes. Les experts les rendent accessibles afin de partager et de diffuser les données de leur domaine, notamment grâce au Web.

Avant de présenter le processus d'extraction et de transformation des données en connaissances à partir des différentes ressources, nous allons tout d'abord définir les termes «donnée», «information» et «connaissance».

1.2 Donnée, information et connaissance

«Connaissance» est un terme utilisé de différentes manières dans plusieurs disciplines, comme la philosophie ou l'informatique. Dans cette thèse, nous nous limitons à une définition tirée des travaux en gestion et représentation de connaissances en informatique présentée par Kayser dans [Kayser, 1997].

- les données sont le résultat d'observations,
- les informations sont le résultat de l'interprétation de ces données,

– les connaissances définissent la façon d'utiliser les informations.

De manière plus formelle, d'après [Schreiber *et al.*, 1999; Wille, 2002] :

– Donnée = signes + syntaxe,

– Information = Données + sens (sémantique),

– Connaissance = information (syntaxe et sémantique) + capacité d'utiliser l'information.

Une donnée est un élément décrit par un signe et associé à une syntaxe. Par exemple : un mot d'un texte, une image, un graphe... sont des données. Mais, une donnée ne peut généralement être comprise ni par une machine, ni par un être humain, car certaines données sont très ambiguës et difficilement interprétables. Une information est une donnée à laquelle un sens (sémantique) a été affecté, comme par exemple, une entrée dans une base de données ou même les méta-données associées à cette entrée, car une information se présente généralement sous forme d'objet-attribut. Par exemple, dire que `ABC` est un `Triangle` (`Triangle :ABC`) est une information. Une connaissance est une information (syntaxe et sémantique) ayant la capacité d'être utilisée pour effectuer des raisonnements. Ainsi, les contraintes, les règles, les axiomes sont des connaissances. Par exemple : «Tous les `Triangles` sont des `Polygones`».

En informatique, la différenciation entre les données et les informations n'est que virtuelle, car les programmes informatiques ne manipulent que des données. D'après Schreiber dans son livre sur la méthodologie de gestion des connaissances CommonKADS, que ce soit pour un programme ou un humain, la frontière entre donnée et information n'est pas franche, car elle est fortement dépendante du contexte d'utilisation [Schreiber *et al.*, 1999]. Par exemple, imaginons deux personnes, un français et un espagnol. Si ces deux personnes lisent un livre en espagnol, pour le français qui ne comprend pas la langue, le livre contient des données alors que pour l'espagnol c'est de l'information. La connaissance par contre est nettement distinguée, car les mécanismes de raisonnements ne sont possibles qu'avec des connaissances.

1.3 Des données aux connaissances

Dans cette thèse, nous nous intéressons tout d'abord à la formalisation des connaissances des experts, ainsi qu'à la découverte d'unités de connaissances à partir de ressources textuelles hétérogènes. Ainsi, nous nous intéressons au processus permettant de passer de données brutes à des unités de connaissances. Ce processus est le processus d'«Extraction de Connaissances à partir de Données» (ECD). Les unités de connaissances doivent être «non triviales, potentiellement utiles, compréhensibles et réutilisables» [Fayyad *et al.*, 1996]. La figure 1.1 présente les étapes du processus d'ECD pour passer des données aux connaissances. Ce processus est composé de trois grandes étapes : d'abord le prétraitement des ressources de données, ensuite l'application de la méthode de fouille où des méthodes symboliques et numériques sont appliquées pour faire émerger des patrons intéressants, et enfin l'interprètation et l'évaluation par les experts du domaine de ces patrons afin d'extraire les unités de connaissances.

Le processus ECD est *itératif* et *interactif* : itératif car il peut être exécuté plusieurs fois et interactif car chaque étape est contrôlée par un expert du domaine, appelé *analyste* qui le dirige afin d'obtenir des connaissances répondant à ces objectifs. Cet analyste interprète, évalue et sélectionne des unités de connaissances pour construire le modèle qu'il considérera comme modèle de connaissances du domaine (ontologie).

Les ontologies sont utilisées dans plusieurs domaines d'application, comme la communication entre agent [Finin *et al.*, 1994], la découverte ou la composition de services Web [Paolucci *et al.*, 2002; Sirin *et al.*, 2003], le traitement automatique de la langue [Nirenburg et Raskin, 2004] ou encore le raisonnement à partir de cas [Taboada *et al.*, 2000; Wriggers *et al.*, 2007; d'Aquin *et al.*,

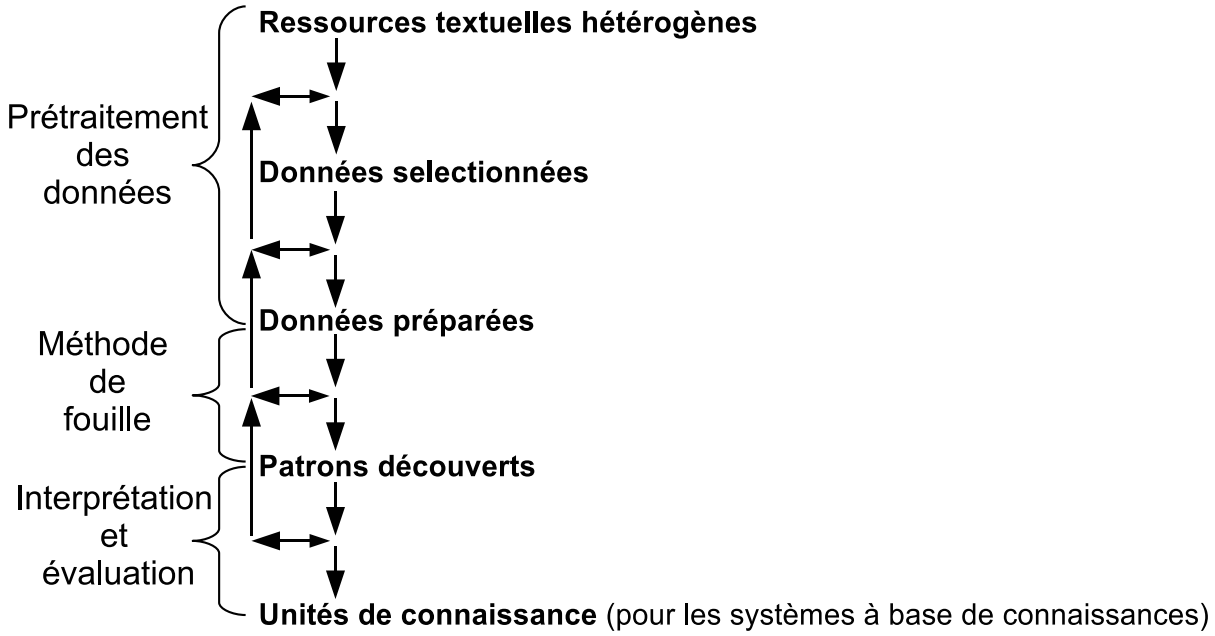


FIG. 1.1 – Le processus d'extraction de connaissances à partir de données : des données aux connaissances

2005]. Mais la construction de telles ontologies est très coûteuse en temps et en main d'œuvre. Elles sont généralement construites manuellement, ce qui provoque un «goulot d'étranglement» dans l'acquisition de connaissances [Cimiano *et al.*, 2005]. Ce problème est attribué au fait qu'une ontologie de domaine doit être la plus exhaustive possible et les experts ne peuvent traiter toutes les données simultanément.

L'objectif de cette thèse est d'aider des experts à construire une ontologie de leur domaine en proposant une approche semi-automatique d'extraction de connaissances à partir de données. Afin de couvrir au maximum le domaine d'intérêt, cette approche doit prendre en entrée différents types de ressources textuelles du domaine : bases de données, thésaurus, corpus de textes ainsi que des ontologies déjà existantes. Chacune de ces ressources est considérée comme un point de vue du domaine, mais aucune n'est dite complète.

1.4 Problématique de la thèse

Pour construire semi-automatiquement une ontologie de domaine à partir de ressources hétérogènes, nous nous sommes intéressés à différents problèmes. Le premier consiste à comprendre la problématique des experts du domaine, c'est-à-dire, leur but et la tâche que doit résoudre l'ontologie qui va être construite. La compréhension du domaine permet de définir les objets du domaine, ceci revient à déterminer quels éléments des différentes ressources textuelles nous serviront à construire l'ontologie la plus précise et la plus exhaustive possible pour le domaine d'intérêt. Le second consiste à regrouper les objets du domaine hiérarchiquement en concepts puis à les relier entre eux afin d'obtenir un schéma de l'ontologie. Dans cette thèse, nous nommons *schéma d'ontologie* (par analogie à un schéma de base de données) l'ensemble des connaissances extraites de toutes les ressources hétérogènes et validées par les experts du domaine avant leur

représentation en un langage formel. Et enfin, le troisième problème traité est la représentation du schéma d'ontologie dans un langage formel pour pouvoir raisonner dessus. Le résultat de cette représentation est appelé *ontologie*.

Le premier problème se pose lorsqu'on veut construire une ontologie à partir des objets du domaine. Il est nécessaire de définir des Descripteurs d'Objets (*DO*) qui nous permettent de regrouper les objets du domaine dans des concepts formels. De plus, les différentes ressources ont chacune une description particulière des objets du domaine. Ainsi, il faut préciser les données à extraire des différentes ressources.

Le deuxième problème se pose lorsqu'on souhaite construire le schéma d'ontologie. Il faut proposer des méthodes de fouille capables de prendre en compte ces différents descripteurs pour construire les hiérarchies de concepts et extraire les relations entre ces concepts.

Le troisième problème se pose pour définir le passage du schéma d'ontologie à l'ontologie, afin de pouvoir raisonner sur les connaissances extraites. Ces connaissances doivent être représentées en un langage formel. Néanmoins, la représentation de ces connaissances n'étant pas triviale, il faut proposer une transformation pour chacune des unités de connaissance extraites dans le formalisme choisi ainsi que les différents types d'interrogation permis par ce formalisme.

1.5 Approches et principales contributions de la thèse

Pour répondre aux trois problèmes présentés dans la section précédente, nous proposons dans cette thèse une méthodologie et un processus nommés PACTOLE : «Property And Class Characterization from Text to OntoLogy Enrichment».

Le premier apport de cette thèse est relatif à la façon de décrire les objets du domaine à partir desquels le schéma d'ontologie est extrait. Avec l'aide des experts du domaine et d'après les différentes ressources disponibles, trois types de descripteurs d'objets sont considérés. Le premier type de descripteur est constitué d'un ensemble prédéfini de classes affectées manuellement aux objets par les experts du domaine. Ce descripteur nous permet de reprendre le travail de classification des experts et de l'enrichir. Le deuxième type de descripteur est constitué d'attributs binaires décrivant des caractéristiques propres aux objets du domaine. Et enfin, le troisième descripteur d'objet regroupe des attributs relationnels, c'est-à-dire les relations entre les objets du domaine. Nous proposons pour chaque type de descripteurs des méthodes d'extraction à partir des différentes ressources.

Le second apport consiste à construire le schéma de l'ontologie en utilisant des méthodes de fouille. Ces méthodes de fouille regroupent et relient les objets du domaine en fonction des différents descripteurs d'objets. La première méthode, l'Analyse Formelle de Concepts (AFC) regroupe un ensemble d'objets partageant un ensemble d'attributs binaires dans un concept formel, et hiérarchise ces concepts formels en un treillis de concepts. La seconde méthode, l'Analyse Relationnelle de Concepts (ARC), une extension de l'AFC, est utilisée pour regrouper un ensemble d'objets partageant un ensemble d'attributs binaires et relationnels dans un concept formel. Ainsi l'ARC permet de prendre en compte les relations entre objets. Les treillis résultant des méthodes de fouille constituent le schéma d'ontologie du domaine. Nous présentons aussi le passage entre le schéma d'ontologie et l'ontologie du domaine en représentant l'ontologie avec le langage des logiques de descriptions (DL) $\mathcal{FLE}$ et en l'implémentant dans le langage *Web Ontology Language* (OWL). Nous présentons également différentes questions auxquelles notre système répond automatiquement en expliquant les mécanismes de raisonnement utilisés.

1.6 Domaines d'application

Pour démontrer que notre méthodologie est indépendante du domaine d'application et qu'elle peut être utilisée sur n'importe quel domaine, nous l'avons expérimentée sur deux domaines très différents.

1.6.1 L'astronomie

Le premier domaine est celui de l'astronomie et plus précisément, la détection et la classification d'objets célestes, *i.e.* l'attribution d'une classe à un objet donné. Cette tâche est très difficile car traditionnellement, la classification des objets est faite manuellement par les astronomes. Elle consiste d'abord à lire les articles scientifiques où apparaissent les objets (première ressource), ces objets pouvant être repérés dans les textes à l'aide d'un dictionnaire de nomenclature[1] (deuxième ressource), puis à trouver la classe de cet objet dans un ensemble de classes prédéfini dans la hiérarchie de la base SimBad [2] (troisième ressource). Cette méthode a permis de classifier plus de quatre millions d'objets célestes dans la base SimBad [3] . Mais SimBad n'est pas une ontologie, car en particulier elle ne contient pas de définition formelle des classes d'objets. Dans ce domaine, notre objectif est de proposer une méthodologie qui aide les experts du domaine à formaliser SimBad, c'est-à-dire classifier les objets célestes d'après les trois ressources présentées précédent. Puis, construire un modèle formel où les objets célestes sont classifiés dans des hiérarchies de concepts. Enfin, nous proposons des questions complexes d'experts auxquelles notre processus répond automatiquement.

1.6.2 La microbiologie

Le second domaine d'application est le domaine de la microbiologie, et plus précisément la classification des bactéries afin d'étudier leur résistance aux antibiotiques par mutations de gènes. La résistance des bactéries est un problème complexe et les microbiologistes cherchent des similarités entre les bactéries qui permettent d'expliquer le phénomène de résistance aux antibiotiques. Cette similarité peut se traduire par un ensemble d'attributs des bactéries ou la présence de gènes mutants dans les bactéries. La figure 1.2 présente à gauche le schéma global de résistance de la bactérie «`Mycobacterium_Tuberculosis`» à l'antibiotique «`Quinolone`». Cette résistance peut être expliquée par la mutation de la région «`Tyr122`» en la région mutante «`Ser-83`», cette région étant une «`partie-du`» gène «`GyrB`» qui lui même était «`partie-de`» l'enzyme «`Gyrase_Topoisomerase`» qui est «`partie-de`» la bactérie «`Mycobacterium_Tuberculosis`». Dans cette thèse, nous simplifions ce schéma pour notre expérimentation et réduisons le schéma à trois types de données (`antibiotiques`, `bactéries` et `gènes`) reliés par deux relations («`isPartOfGenomeOf`» et «`isResisting`»).

Dans ce domaine, la ncbi Taxonomy[4] (première ressource) est une classification manuelle de plus de 13890 bactéries dans 5929 classes prédéfinies. Cependant cette hiérarchie ne tient compte ni des gènes que contiennent les bactéries ni de la capacité des bactéries à résister aux antibiotiques. Il existe aussi des milliers d'articles scientifiques qui décrivent ce phénomène de résistance (deuxième ressource). Les experts du domaine ont aussi mis en place des bases de

[1] Base SimBad : `http://vizier.u-strasbg.fr/cgi-bin/Dic-Simbad?` Date : 14/07/09
[2] Base SimBad : `http://simbad.u-strasbg.fr/simbad/sim-display?data=otypes` Date : 14/07/09
[3] Base SimBad : `http://simbad.u-strasbg.fr/simbad/sim-fid` Date : 14/07/09
[4] Classification ncbi : `http://www.ncbi.nlm.nih.gov/Taxonomy/Browser/wwwtax.cgi?name=Eubacteria`. Date : 14/07/09

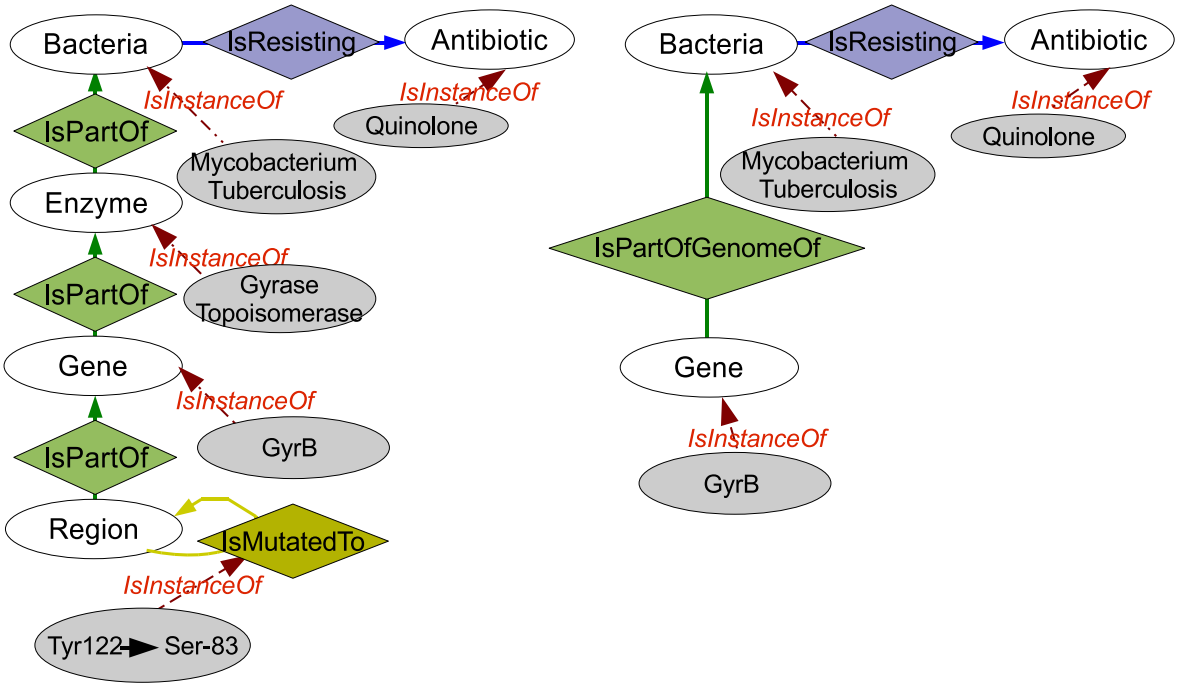


FIG. 1.2 – A gauche le schéma global décrivant le phénomène de résistance des bactéries aux antibiotiques par mutation d'une région dans un gène et à droite le schéma simplifié pour notre expérimentation

données contenant des attributs binaires des bactéries, comme par exemple MESH DATABASE[5] (troisième ressource), des fonctions des gènes dans les bactéries, comme par exemple, la GENE ONTOLOGY[6] (quatrième ressource) et des antibiotiques, comme la classification des ligands (cinquième ressource). Ainsi, dans ce domaine, il faut aussi donner une définition à chaque classe de la base NCBI, puis proposer des méthodes de classification semi-automatiques des bactéries. Des questions complexes d'experts auxquelles notre système répond automatiquement seront proposées.

1.7 Organisation de la thèse

Cette thèse est organisée en six chapitres. Cette introduction générale constitue le premier chapitre. Le deuxième et le troisième fixent le contexte et l'état de l'art relatifs à la problématique de cette thèse. Les trois suivant détaillent les contributions de la thèse. La dernière partie présente la conclusion et les perspectives de ce travail.

Chapitre 2 : Un monde avec des ressources hétérogènes. Dans ce chapitre, les différentes ressources hétérogènes à partir desquelles les connaissances peuvent être extraites sont présentées. Puis, un guide des méthodologies proposées dans la littérature pour construire une ontologie de domaine est détaillé. Enfin, les langages du Web sémantique ainsi que les langages des logiques

[5] MESH DATABASE `http://www.ncbi.nlm.nih.gov/sites/entrez?db=mesh`. Date : 14/07/09
[6] GENE ONTOLOGY : `http://www.geneontology.org/`. Date : 14/07/09

de descriptions (LD) utilisés pour représenter et implémenter une ontologie de domaine sont détaillés.

Chapitre 3 : Extraction de connaissances. Dans ce chapitre, nous détaillons les processus permettant de passer de données brutes à des connaissances sur lesquelles il est possible de raisonner. Ces processus sont les processus d'Extraction de Connaissances à partir de Données ECD. Ensuite, nous présentons la méthode de fouille que nous avons choisie pour extraire des unités de connaissances à partir de données brutes. Cette méthode est l'Analyse Formelle de Concepts (AFC), une approche mathématique, qui grâce à ses propriétés et à ses extensions, permet de classifier des objets d'après les attributs binaires et relationnels qu'ils partagent.

Chapitre 4 : Méthodologie Pactole : Prétraitements des ressources. Dans ce chapitre, nous présentons notre méthodologie et notre processus nommés PACTOLE «Property And Class Characterization from Text to OntoLogy Enrichment». Puis, nous détaillons la première partie du processus PACTOLE qui consiste à extraire les différents descripteurs d'objets présents dans les ressources hétérogènes. Ensuite, nous proposons des méthodes d'extraction de ces descripteurs d'objets.

Chapitre 5 : Méthodologie Pactole : Extraction de connaissances à partir de ressources. Dans ce chapitre, nous appliquons le processus PACTOLE sur les deux domaines spécifiques, l'astronomie et la microbiologie. Pour chaque domaine spécifique, nous utilisons les éléments extraits du chapitre précédent pour construire le schéma de l'ontologie. Ensuite, nous détaillons notre représentation du schéma d'ontologie en logique de descriptions et les différentes questions auxquelles notre système peut répondre.

Chapitre 6 : Expérimentations et évaluation. Dans ce chapitre, nous présentons l'expérimentation de notre méthodologie dans les deux domaines d'application : l'astronomie et la microbiologie, ainsi qu'une évaluation de chaque étape de cette méthodologie. Puis, nous proposons des méthodes d'interaction avec les experts du domaine, pour améliorer les résultats et raffiner la classification des objets du domaine.

Conclusion et perspectives. Cette partie conclut notre travail en proposant quelques perspectives.

Chapitre 2

Un monde aux ressources hétérogènes

Sommaire

Introduction

Dans ce chapitre, nous présentons les ressources textuelles hétérogènes collectées par les experts d'un domaine spécifique afin de modéliser leur domaine. Pour chacune de ces ressources nous détaillons ses caractéristiques ainsi qu'un exemple. Nous donnons ensuite les différentes méthodologies de construction d'ontologies afin de proposer un guide de construction d'après les besoins exprimés par les experts. Enfin, nous présentons les différents langages proposés dans le domaine du Web sémantique pour passer du schéma d'ontologie à une ontologie représentée en un langage formel. Nous puisons nos exemples dans « les formes géométriques ».

2.1 Ressources hétérogènes

Dans cette thèse, nous distinguons trois types de ressources textuelles : celles contenant des données brutes comme les corpus de textes ou les dictionnaires, celles contenant des données structurées comme les bases de données ou les thésaurus et enfin, celles contenant directement

de la connaissance comme les ontologies. Généralement ces ressources ne répondent pas aux mêmes besoins des experts. Dans les sous-sections suivantes, nous présenterons ces différentes ressources textuelles en expliquant à quels besoins elles répondent.

2.1.1 Corpus de textes

Les textes sont les premières et les plus importantes ressources de données dans n'importe quel domaine spécifique. Depuis des millénaires, les hommes utilisent l'écriture pour stocker leurs connaissances. Brewster et al. [2003] définissent les textes comme étant une ressource de connaissances dans n'importe quel domaine spécifique, car cette ressource est maintenue et mise à jour (à chaque découverte et chaque année, des millions d'articles, livres, revues... sont publiés). Un corpus de textes est une collection de *documents textuels*. Sinclair [1996] le définit comme étant « une collection de morceaux de langage qui sont sélectionnés et organisés selon des critères linguistiques explicites pour servir d'échantillon du langage. » Plusieurs autres auteurs reprennent cette définition, tels que Ananiadou et Mc Naught [2005] ou encore Feldman [2007] qui ajoutent que « la sélection d'un corpus de textes peut être faite pour une application ou une tâche particulière ». Nous utiliserons cette définition en précisant que « cette application est définie par les experts du domaine ».

Constitution d'un corpus. Plusieurs cas de figure peuvent mener à l'élaboration d'un corpus [Lame, 2002]. Si le domaine spécifique sur lequel nous souhaitons extraire des connaissances est bien documenté, alors l'enjeu est de collecter ces documents. Deux solutions sont possibles. La première consiste à retrouver ces documents à partir du Web, *i.e.* plusieurs requêtes décrivant le domaine spécifique sont posées à des moteurs de recherche. La deuxième solution est de demander aux experts du domaine de rassembler des textes traitant tous du même problème spécifique. Si le domaine spécifique est mal documenté, alors le corpus doit être créé pour cette tâche. Ce cas de figure se présente quand le domaine spécifique doit capturer, par exemple, la capacité d'un expert à traiter un problème. La connaissance des experts est alors capturée à partir de transcriptions d'interviews. Les corpus de textes constituent la plus grande ressource de données pour extraire de la connaissance. Cependant, la constitution d'un corpus de textes dans un domaine spécifique est une tâche très difficile, car il faut regrouper beaucoup de documents textuels pour couvrir tout le domaine et en même temps faire en sorte que les documents ne soient pas trop dispersés. Cela signifie que le corpus ne couvre pas plus que le domaine spécifique et l'application que les experts veulent étudier.

Exemples de textes. Nous citons quelques exemples dans le domaine de la géométrie extraits du site WikiPedia[7] dans la figure 2.1. Les deux textes présentés dans cette figure, présentent respectivement le terme `triangle` et le terme `carré`. Plusieurs données sont intéressantes dans ces textes ; par exemple nous pouvons définir un `triangle` comme étant une « forme plane », nous pouvons aussi faire une distinction entre un `triangle` qui a « trois angles » et un `carré` qui a « quatre angles ».

Dans cette thèse nous nous intéressons aux corpus de textes décrivant des objets d'un domaine. Nous nommons objets du domaine, tout terme pouvant décrire le domaine d'application. Par exemple, les termes `carré` et `triangle` sont des objets du domaine des formes géométriques.

[7] WikiPedia : `http://fr.wikipedia.org/wiki`. Date : 14/07/09

En géométrie euclidienne, un triangle est une figure plane, formée par trois points et par les trois segments qui les relient. La dénomination de « triangle » est justifiée par la présence de trois angles dans cette figure, ceux formés par les segments entre eux. Les trois points sont les sommets du triangle, les trois segments ses côtés, et les trois angles ses angles.

Un carré est un polygone régulier à quatre côtés, tous de même longueur. C'est un quadrilatère qui est à la fois un rectangle car il a quatre angles droits, et un losange car ses quatre côtés ont la même longueur.

FIG. 2.1 – Des exemples de textes traitant des formes géométriques

Dictionnaires et Encyclopédies. Les dictionnaires et les encyclopédies sont des corpus de textes particuliers. A partir du 12^{ieme} siècle, les scientifiques ont voulu construire une ressource plus structurée et plus facile à utiliser que les textes pour retrouver des données. C'est ainsi, que les dictionnaires ont été mis en place. Un dictionnaire donne pour chaque mot d'une langue sa définition et/ou ses synonymes ou équivalents dans une autre langue (si le dictionnaire est bilingue). Ces données sont stockées par ordre alphabétique, sous forme de lemme.

Exemple d'un dictionnaire. Cet exemple de la définition du terme `triangle` est tiré de Wikitionary[8]

`Triangle` nom masculin

1. (Mathématiques) Figure qui a trois côtés, et donc trois angles. D'après un principe de géométrie, un triangle quelconque est entièrement « connu », quand on connaît un de ses côtés et deux de ses angles, car on peut conclure immédiatement la valeur du troisième angle et la longueur des deux autres côtés.
2. (Par extension) Objet de forme triangulaire.
3. (Musique) Instrument à percussion de l'orchestre, fait d'une baguette métallique courbée de façon triangulaire et percutée avec une tringle métallique. Le triangle émet un son cristallin.

Une encyclopédie est un ouvrage qui expose : (1) un ensemble de textes sur le même domaine spécifique, (2) une structure qui organise les données entre elles de manière à donner aux lecteurs une vue panoramique de l'ensemble et des axes de recherche et ainsi permettre aux lecteurs de retrouver plus facilement les informations recherchées. La plus célèbre des encyclopédie est « l'encyclopédie ou Dictionnaire raisonné des sciences, des arts et des métiers » de Diderot et D'Alembert qui est la première encyclopédie française, éditée de 1751 à 1772.

Exemple d'une encyclopédie. Nous donnons comme exemple d'encyclopédie une partie de l'encyclopédie de Diderot et d'Alembert. Dans cette partie (voir figure 2.2), le domaine des mathématiques est divisé en sous-domaines et les sous domaines sont eux mêmes divisés et ainsi de suite.

Quoique mieux structurés que les corpus de textes, les dictionnaires et les encyclopédies présentent moins de richesse dans les données. Un expert du domaine utilise un dictionnaire ou

[8]Wikitionary : `http://fr.wiktionary.org/wiki/triangle`. Date : 14/07/09

FIG. 2.2 – Une partie de l'encyclopédie de Diderot et d'Alembert

une encyclopédie pour définir un mot, un terme ou rechercher une information. Mais, s'il veut étudier un phénomène, savoir quelles expériences ont déjà été éffectuées sur un domaine, seuls les corpus de textes peuvent répondre à ses besoins.

2.1.2 Thésaurus

Un thésaurus est généralement issu d'un long processus de tri effectué manuellement par des experts pour stocker les informations de leur domaine d'intérêt. Il contient des termes classés en une sorte de dictionnaire hiérarchisé, mais aussi des relations entre les différents termes. Parmi les thésaurus, nous citons par exemple, le thésaurus « Art and Architecture Thesaurus » [9] sur les arts et architectures ou encore le thésaurus « Union List of Artist Names »[10]... Un thésaurus est une structuration hiérarchisée d'un ou plusieurs domaines dans lequel les notions sont décrites par des termes d'une ou plusieurs langues naturelles et les relations entre notions par des signes conventionnels. Il définit plusieurs relations entre termes. Les relations des termes peuvent êtres divisées en trois familles de relations : (1) les relations hiérarchiques (entre descripteurs), (2) les relations d'équivalence (entre descripteurs et non-descripteurs), ce qui signifie que pour chaque deux synonymes, le thésaurus définit le préférentiel qu'il faut utiliser et enfin, (3) les relations d'association (entre descripteurs). Ces différentes relations sont détaillées comme suit :

- NT : terme plus spécifique (*narrower term* en anglais). Cette relation signifie que ce terme est plus spécifique que le premier. Par exemple :
  `Polygone`
  NT `Triangle`,
- BT : terme plus général (*broader term* en anglais). Cette relation signifie que ce terme est plus général que le premier. Par exemple :

[9] Art and Architecture Thesaurus : `http://www.getty.edu/research/conducting_research/vocabularies/aat/`. Date : 14/07/09

[10] Union List of Artist Names : `http://e-culture.multimedian.nl/resources/ulan.rdfs` Date : 14/07/09

`Triangle`
BT `Polygone`,

- RT : terme en relation (*related term* en anglais). Cette relation signifie que les termes sont reliés par une relation, la relation n'est pas définie. Par exemple :
  `Triangle`
  RT `Angle`,
- UF : terme utilisé (*used for* en anglais). Cette relation est utilisée comme lien entre un descripteur et ses synonymes ou non-descripteurs. Par exemple, pour dire que les deux termes `Segment` et `Coté` sont des synonymes, il faut utiliser le terme `Segment` :
  `Coté`
  UF `Segment`,
- U : terme non utilisé. Cette relation est la relation inverse de UF. Elle est utilisée comme lien entre un non-descripteur et son descripteur. Reprenons le même exemple :
  `Segment`
  U `Coté`.

Les thésaurus sont des structures très utilisées dans l'indexation et la recherche d'information. Mais, ces structures ne sont pas formelles et ne définissent pas réellement les liens entre termes, ce qui revient à dire qu'avec la relation RT, nous pouvons savoir que le terme `Triangle` est en relation avec le terme `Angle`, mais la relation n'est jamais précisée.

Hiérarchie de matières. Les hiérarchies de matières (en anglais *topic hierarchies*) sont des thésaurus particuliers. Elle sont de simples ensembles de termes organisés par une hiérarchie mélangeant les relations : is-a, part-of, contained-in ... Par exemple : Open Directory hierarchy[11]

Vocabulaires intégrés. Les vocabulaires intégrés aussi appelés méta-thésaurus, sont des ensembles de plusieurs thésaurus manuellement regroupés en un seul. Par exemple, le vocabulaire intégré médicals en ligne « Unified Medical Language System » UMLS [McCray et Nelson, 1995].

Ressources linguistiques. Les ressources linguistiques sont des types de dictionnaires qui regroupent les termes synonymes en un ensemble appelé *synset*. Ces synsets sont reliés entre eux par différentes relations sémantiques et syntaxiques. La plus connue de ces ressources linguistiques est la ressource WordNet[12]

Nous pouvons extraire des données très intéressantes à partir des ressources linguistiques, des hiérarchies de matières ou des thésaurus : définitions, synonymes et même une hiérarchisation des objets du domaine. Cependant ces ressources ne sont pas formelles et quoi qu'elles contiennent des définitions, elles ne définissent pratiquement pas d'autres relations que les relations de synonymie et d'hyponymie entre les données.

2.1.3 Base de données

Au début de l'expansion de l'informatique, vers les années soixante, il fallait trouver un moyen de gérer de grands volumes de données et de les structurer afin de les retrouver plus facilement. Les bases de données (BD) ont vu le jour lorsque Edgar Frank Codd [1970] proposait de stocker des données hétérogènes dans des tables, permettant d'établir des relations entre elles. Les bases de données permettent de stocker des informations (objet-attribut), de construire des

[11] Open Directory hierarchy : `http://www.dmoz.org/Computers/Usenet/Hierarchies/`. Date : 14/07/09
[12] WordNet : `http://wordnet.princeton.edu/`. Date : 14/07/09

tables reliées entre elles par des relations, hiérarchiques ou n-aires. Elles facilitent la recherche et la classification d'informations. Nous présentons une définition des bases de données extraite de Godin [2000].

Définition 1 *Une base de données est un ensemble de données :*
- *fortement structurées,*
- *persistantes,*
- *définies dans un schéma,*
- *gérées par système de gestion de bases de données.*

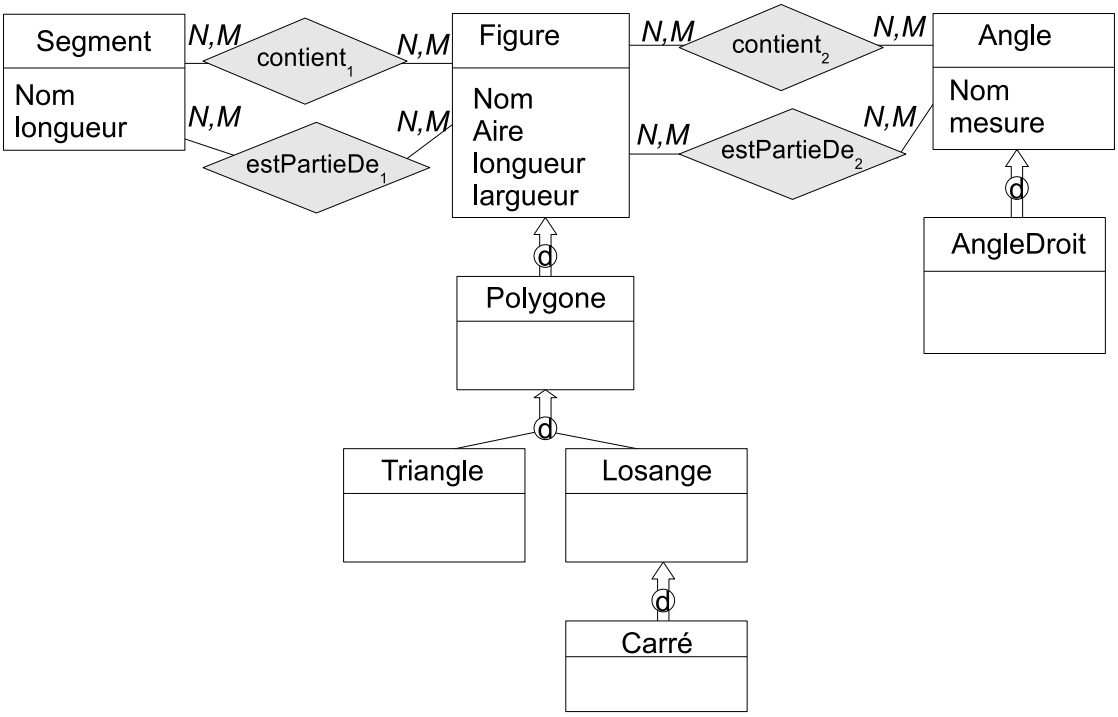


FIG. 2.3 – Un exemple d'un schéma de base de données sur les formes géométriques

Exemple d'un schéma de base de données. La figure 2.3 présente un exemple de schéma de base de données sur les formes géométriques. Ce schéma nous permet de décrire les cinq éléments suivants : (1) des tables, c'est-à-dire un ensemble d'objets partagent les mêmes attributs, (2) des attributs, par exemple, la table `Angle` possède les attributs `nom, mesure`, (3) des relations, par exemple, `contient` entre `Figure` et `Angle`, ou encore `estPartieDe` entre `Angle` et `Figure`, (4) des relations hiérarchiques entre les tables, par exemple la table `Triangle` est une sous-table de la table `Polygone` qui est elle même est une sous table de la table `Figure`, et enfin (5) des cardinalités d'une relation, par exemple, une `Figure` peut contenir `N Segment`.

Système de gestion de bases de données. Un Système de Gestion de Bases de Données (SGBD) est un ensemble d'outils logiciels permettant la création et l'utilisation de bases de données [Boudjlida, 1999]. Il doit offrir à l'utilisateur les moyens de décrire le schéma de la base de données. Ces moyens constituent ce qu'on appelle généralement le Langage de Descriptions de

Données (LDD). Plusieurs langages pour SGBD ont été développés pour interroger les bases de données et retrouver facilement les données recherchées. Le plus utilisé est le langage « Structured query language » (SQL) [ISO, 1989; April 1990] qui peut être vu comme étant composé de trois sous-langages : un langage de manipulation de données, un langage de descriptions de données et un langage de contrôle des données [Boudjlida, 1999].

Une base de données nous permet de décrire un ensemble d'objets partageant les mêmes attributs. Elle nous permet aussi de définir des relations entre les différentes tables qu'elle possède. Le SGBD nous permet d'interroger les bases de données et offre ainsi aux experts d'un domaine des services inexistants dans les autres ressources telles que les corpus de textes ou les thésaurus. Néanmoins, une base de données est une ressource moins riche que les corpus de textes, car elle est limitée à son schéma (schéma de base de données). Mais, les bases de données ne définissent pas formellement les tables des objets et leurs relations. Elles ne disposent pas non plus d'axiomes qui permettent par exemple d'expliquer que « tout ce qui possède une aire peut être mesuré ».

2.1.4 Ontologie

Le mot ontologie vient du grec ontologia qui veut dire « parler » (-logia) de l'« être » (onto-). L'ontologie est une discipline philosophique qui décrit la « science de l'existant » ou la « science de l'être ». Platon (427-347 AV-JC) a été l'un des premiers philosophes à s'intéresser à la représentation du monde et à l'abstraction des entités dont on parle. Cette représentation est l'idée fondamentale de l'ontologie. Aristote (384-322 AV-JC) a introduit la notion de concept et de taxonomie entre ces concepts. Il a aussi introduit la notion de sous-concept/super-concept, car il voulait distinguer les genres afin de les classifier formellement. C'est ce principe qui est utilisé pour définir la notion moderne de concept d'ontologie et l'héritage entre concepts. Aristote a aussi introduit un certain nombre de règles d'inférence appelées « syllogismes » qui ont été utilisées par la logique moderne des systèmes de raisonnement [Sowa, 2000].

En informatique, une ontologie ne décrit plus la « science de l'existant », mais « une ontologie est une spécification explicite et formelle d'une conceptualisation partagée » [Gruber, 1993]. Dans cette définition le terme « explicite » veut dire que les concepts de l'ontologie sont explicitement définis ; « formelle » montre qu'elle est représentée dans un formalisme qui permet aux machines d'effectuer du raisonnement ; « conceptualisation » se réfère à un modèle d'abstraction ; enfin « partagée » signifie qu'une ontologie n'est pas propre à un seul individu mais validée par un groupe [Studer *et al.*, 1998].

Plusieurs ontologies sont accessibles par le Web. Par exemple, plusieurs bibliothèques d'ontologies sont disponibles sur le site Web DAML[13]

Exemple d'ontologie. L'ontologie de la figure 2.4 présente un petit exemple d'ontologie sur les formes géométriques. Cette ontologie contient un ensemble de concepts, comme `Triangle`, un ensemble de relations comme `estPartieDe` entre `Figure` et `Segment`, et d'attributs, comme `aPourLongueur`, des instances, comme `ABC` et enfin des types de données, comme `entier`. Ces notions sont expliquées dans le paragraphe suivant.

Le schéma d'ontologie. Une ontologie est une structure formelle, c'est-à-dire qu'elle est représentée dans un langage formel. Or, lors de l'extraction de connaissances à partir des ressources hétérogènes d'un domaine d'intérêt, l'ensemble des différents éléments extraits ne constitue pas

[13] DAML : `http://www.daml.org/`. Date : 14/07/09

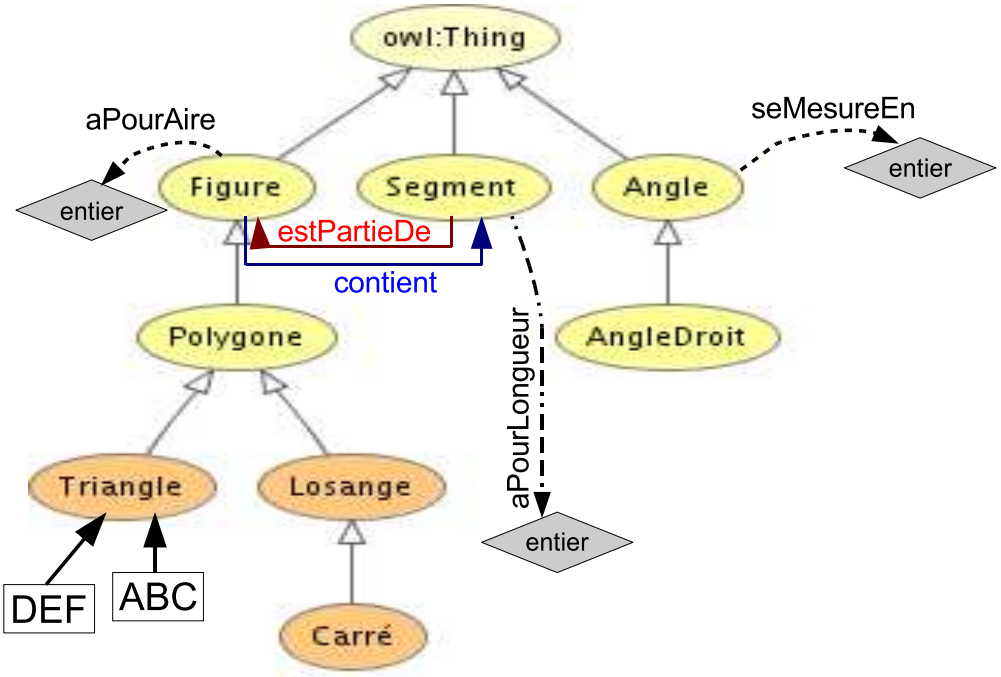


FIG. 2.4 – Une partie d'ontologie des formes géométriques

encore une ontologie car il n'est pas encore formalisé. Néanmoins, il constitue ce que nous appelons *schéma d'ontologie* (par analogie à un schéma de base de données, présenté dans la section précédente). Le terme d'ontologie est utilisé dans cette thèse pour parler de la représentation du schéma de l'ontologie en un langage formel.

Nous présentons tout d'abord la notion de schéma d'ontologie puis celle de lexique d'ontologie. Nous reprenons la définition de Cimiano [2006] pour décrire les éléments d'un schéma d'ontologie.

Définition 2 (schéma d'ontologie) *Un schéma d'ontologie est une structure* $\mathcal{O} := (C, \sqsubseteq_C, R, \sigma_R, \sqsubseteq_R, A, T)$ *constituée de :*

- *trois ensembles disjoints* C, R *et* A *appelés respectivement ensemble des concepts, ensemble des relations et ensemble des attributs,*
- *une relation de subsomption sur l'ensemble des concepts* $\sqsubseteq_C$ *avec un élément* $\top$ *constituant la racine de la hiérarchie des concepts,*
- *une fonction* $\sigma_R \rightarrow C^+$ *appelée signature de la relation,*
- *un ordre partiel* $\sqsubseteq_R$*, appelé relation de subsomption sur l'ensemble* R*, où* $r_1 \sqsubseteq_R r_2$ *implique que* $|\sigma_R(r_1)| = |\sigma_R(r_2)|$ *et* $\pi_i(\sigma_R(r_1)) \sqsubseteq_C \pi_i(\sigma_R(r_2))$ *pour tout* $1 \leq i \leq |\sigma_R(r_1)|$*,*
- *un ensemble* T *de types de données comme par exemple : entier, chaînes de caractères,...*

La fonction σ_R affecte à chaque relation, les concepts qui sont reliés par cette relation. Par exemple, dans la figure 2.4, $\sigma_R(contient)$ relie la relation `contient` aux deux concepts : `Figure` et `Segment`. $\pi_i(t)$ est le i^{eme} composant du tuple t.

L'ensemble R est l'ensemble des n aires entres concepts. Dans cette thèse, nous nous limitons aux « relations transversales ». Une relation transversale est une relation binaire entre deux concepts autre que la relation de subsomption.

Lexique d'un schéma d'ontologie. Le terme lexique est utilisé dans plusieurs domaines. En Traitement Automatique de la Langue Naturelle (TALN), il est défini en tant qu'ensemble des lemmes d'une langue naturelle ou d'un domaine particulier (lexique d'un corpus de textes) [Jurafsky et Martin, 2000]. Dans le domaine de l'extraction de connaissances et de la construction d'ontologie, le lexique d'un schéma d'ontologie est l'ensemble des références pour les concepts, les relations et les attributs [Cimiano, 2006].

Définition 3 (Lexique d'un schéma d'ontologie) *Un lexique pour un schéma d'ontologie* $\mathcal{O} := (C, \sqsubseteq_C, R, \sigma_R, \sqsubseteq_R, A)$ *est une structure :*

$$Lex := (S_C, S_R, S_A, Ref_C, Ref_R, Ref_A)$$

Où :
- *les trois ensembles* S_C, S_R, S_A *sont appelés « instances » pour les concepts, les relations et les attributs respectivement,*
- *la relation* $Ref_C \subseteq S_C \times C$ *est appelée référence lexicale pour les concepts,*
- *la relation* $Ref_R \subseteq S_R \times R$ *est appelée référence lexicale pour les relations,*
- *la relation* $Ref_A \subseteq S_A \times A$ *est appelée référence lexicale pour les attributs.*

et :

$$\forall s \in S_C, Ref_C(s) := \{c \in C | (s,c) \in Ref_C\},$$
$$\forall c \in C, Ref_C^{-1}(c) := \{s \in S_C | (s,c) \in Ref_C\}.$$

Ref_R *et* Ref_A *sont définies de la même façon que* Ref_C.

Dans l'ontologie de la figure 2.4 sur les formes géométriques, `Carré`, `Losange`, `Polygone`, `Triangle`, `Figure`, `Angle`, `AngleDroit` et `Segment` constituent les concepts du schéma de l'ontologie. `contient` et `estPartieDe` sont les relations du schéma de l'ontologie. `aPourLongueur`, `seMesureEn` et `aPourAire` sont des attributs du schéma de l'ontologie et enfin `ABC, DEF` constituent les instances du schéma de cette ontologie. `ABC, DEF` sont deux instances du concept `Triangle`, et donc : Ref_C(`ABC`) := {`Triangle, Polygone, Figure`} et Ref_C^{-1}(`Triangle`) := {`ABC, DEF`}. Il est très important de noter que plusieurs lexiques peuvent être associés au même schéma d'ontologie. Cela dépend de la définition des ensembles S_C, S_R, S_A.

2.1.5 Bilan

Dans cette section nous avons présenté les différentes ressources contenant les différents points de vue d'un domaine. Chacune d'elles présente des données complémentaires qui serviront à construire une ontologie la plus exhaustive possible. Le rôle des experts est primordial, tout d'abord dans le choix du domaine spécifique puis, dans la précision de l'application pour laquelle une ontologie doit être construite. Ce sont les experts qui sélectionnent le ou les corpus de textes et qui indiquent les différentes bases de données, thésaurus ou bases de connaissances déjà existantes en rapport avec l'application choisie.

2.2 Guide des méthodologies de construction d'ontologies

Il existe dans la littérature une dizaine de méthodologies de construction d'ontologie. Pour qu'un expert choisisse la méthodologie qui convient à ses besoins, il doit tout d'abord définir le but de l'ontologie ainsi que les caractéristiques de la méthodologie et de l'ontologie résultante.

2.2.1 Identification du but de d'une ontologie

Une ontologie est construite dans un domaine spécifique et dans un but précis. Il ne faut pas penser à modéliser un domaine, mais toujours choisir la meilleure solution selon l'application. La méthodologie de Uschold et King [1996] fut la première à être proposée et constitue la base des autres méthodologies. De plus, elle établit un guide de construction qui définit clairement les méthodes de construction de l'ontologie. La première étape de cette méthodologie est d'identifier clairement le but de l'ontologie, de définir pourquoi celle-ci a été construite (est-ce qu'elle doit être réutilisée, partagée...). Toutes les autres méthodologies reprennent ce même principe. Dans la méthodologie de Grüninger et Fox [1995], il faut définir des scénarios de motivation. Ces scénarios consistent à poser le problème qui motive la construction. Il s'agit généralement d'un problème de partenaires industriel, qu'on doit résoudre par l'ontologie. Un scénario de motivation est constitué par un ensemble de questions (appelées questions de compétence) écrites en langue naturelle et auxquelles doit répondre l'ontologie. Ces questions doivent traiter de l'activité de l'ontologie et de son organisation (quels agents doivent intervenir ?, pourquoi faire ?, ...).

Noy et McGuinness [2001] reprennent la même idée en définissant les premières questions à se poser et dont les réponses peuvent changer durant la vie de l'ontologie. Ce sont les experts du domaine qui doivent répondre à toutes ces questions avant d'entreprendre la construction d'une ontologie :

- quel domaine couvrira cette ontologie ?
- à quoi servira cette ontologie ?
- à quel type de questions répondra-t-elle ?
- qui utilisera et maintiendra l'ontologie ?

2.2.2 Caractéristiques d'une méthodologie

Après avoir répondu aux questions sur le but de l'ontologie, les experts du domaine doivent caractériser la méthodologie qu'ils veulent utiliser. Nous reprenons des caractéristiques de différentes méthodologies qui serviront à les classifier. Une méthodologie peut :

1. être contrôlée par un expert : une ontologie doit répondre aux attentes des experts du domaine, pour cela chaque étape de sa méthodologie de construction doit être contrôlée par ce dernier,
2. utiliser des ressources du domaine : une ontologie ne doit pas être construite sans tenir compte des ressources existantes,
3. représenter formellement l'ontologie : une ontologie est formelle et elle doit être représentée formellement pour pouvoir raisonner dessus,
4. avoir un processus semi-automatique : la construction manuelle d'une ontologie est une tâche coûteuse en temps et en main d'œuvre,
5. utiliser des outils de fouille de données : puisque le processus de la méthodologie est semi-automatique, ce processus doit donc utiliser des méthodes d'apprentissage,
6. évaluer l'ontologie résultante : une ontologie étant construite pour des besoins précis, elle doit être évaluée pour savoir si elle répond à ces besoins,
7. prendre en compte l'évolution de l'ontologie : les besoins et les ressources de l'ontologie peuvent changer durant son cycle de vie.

Dans les paragraphes suivants, nous détaillerons les différentes caractéristiques d'une méthodologie de construction d'ontologie.

Méthodologies contrôlées par un expert. Comme dit précédemment, une ontologie construite sans l'intervention d'un expert est une ontologie irréaliste [Noy et McGuinness, 2001]. Les concepts doivent être les plus proches des objets du domaine (physiquement ou logiquement). C'est pour cela qu'une méthodologie doit être contrôlée par un expert du domaine et que cet expert doit intervenir pour valider chaque étape.

Méthodologies utilisant des ressources du domaine. Certaines méthodologies comme METHONTOLOGY [Gómez-pérez *et al.*, 2004] ou encore la méthodologie de Noy et Guiness [2001] recommandent d'utiliser des ressources existantes, mais permettent de construire une ontologie dans le cas où le domaine ne possède pas de ressources. Toutefois, dans cette thèse, nous pensons que dans la mesure du possible, une ontologie ne doit pas être construite *ex nihilo*, car il est important de tenir compte des travaux antérieurs et de réutiliser ou d'améliorer les résultats obtenus. La première méthodologie à avoir proposé d'intégrer la connaissance des experts et les ressources du domaine est la méthodologie KADS [Schreiber *et al.*, 1999].

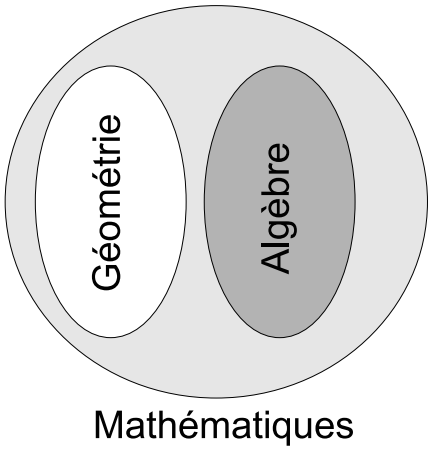


FIG. 2.5 – Application de la méthodologie KACTUS sur l'exemple des formes géométriques

Un domaine d'intérêt peut disposer de différents types de ressources comme présenté dans la section 2.1, tels que des corpus de textes, des dictionnaires, des thésaurus, des bases de données... La prise en compte des ressources du domaine peut être faite de deux manières différentes.

Décomposition en composants. La méthodologie KACTUS a été proposée dans [Bernaras *et al.*, 1996] afin d'étudier la réutilisation de l'ontologie dès sa création. Le principe est de décomposer l'ontologie en composants afin de construire une nouvelle ontologie. Par exemple dans la figure 2.5, une ontologie sur le domaine « mathématiques » peut être divisée en composants : Algèbre, Géométrie, ... La méthodologie CYC [Lenat et Guha, 1990] reprend cette idée de décomposition en permettant à des agents de communiquer en n'utilisant qu'une partie de l'ontologie existante.

Enrichissement d'une ontologie déjà existante. La méthodologie SENSUS [Valente *et al.*, 1999] (décrite dans la figure 2.6) enrichit une ontologie déjà existante à partir de ressources hétérogènes. Elle est composée des cinq étapes suivantes :

– Identifier des termes de l'ontologie existante dans les différentes ressources ;

- Relier les termes (extraits dans l'étape précédente) aux concepts dont ils sont instances ;
- Ajouter les chemins jusqu'à la racine, c'est-à-dire extraire la partie de la hiérarchie qui est reliée aux concepts extraits dans l'étape précédente ;
- Ajouter les nouveaux termes du domaine dans l'ontologie, c'est-à-dire les nouveaux termes extraits ;
- Ajouter les sous-arbres complémentaires, ce qui revient à compléter la hiérarchie de concepts en ajoutant les sous-arbres reliés aux nouveaux concepts.

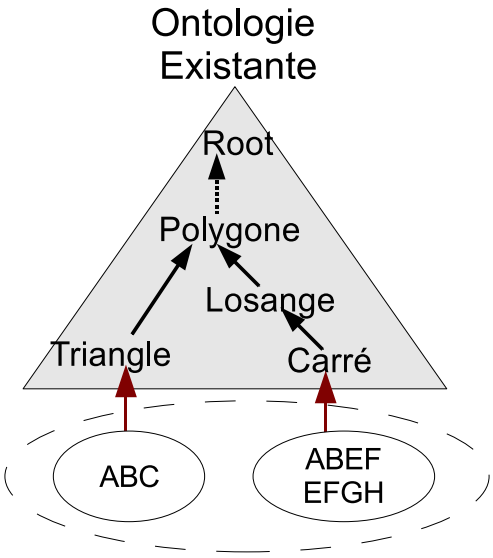


FIG. 2.6 – Méthodologie Sensus

Méthodologies avec un processus semi-automatique. La construction manuelle des ontologies est une tâche très coûteuse en temps et en compétences. A contrario une méthodologie ne peut pas être automatisé à 100 %, car il faut absolument qu'elle soit contrôlée par des experts du domaine. Les deux méthodologies proposant un processus semi-automatique sont METHONTOLOGY [Gómez-pérez *et al.*, 2004], et ON-TO-KNOWLEDGE [Staab *et al.*, 2001]. METHONTOLOGY est une méthodologie détaillée où pour chaque étape, des méthodes sont proposées, tandis que ON-TO-KNOWLEDGE est une méthodologie plus générale qui décrit essentiellement la vie de l'ontologie. ON-TO-KNOWLEDGE présentée dans la figure 2.7 décompose la vie de l'ontologie en cinq étapes :

- Étude de la faisabilité de l'ontologie : identification de la problématique, de la faisabilité et du domaine spécifique de la construction afin d'imaginer la solution adéquate au problème,
- Ontologie : construction de l'ontologie du domaine représentée en logique de descriptions,
- Raffinement : enrichissement de l'ontologie construite à l'étape précédente avec l'aide d'un expert du domaine,
- Évaluation : identification des problèmes de l'ontologie en utilisant des mesures d'évaluation,
- Maintenance : organisation et mise en place d'un processus de maintenance.

Méthodologies utilisant des outils de fouille Les deux méthodologies avec des processus semi-automatiques proposent d'utiliser des outils de fouille de données. METHONTOLOGY ne

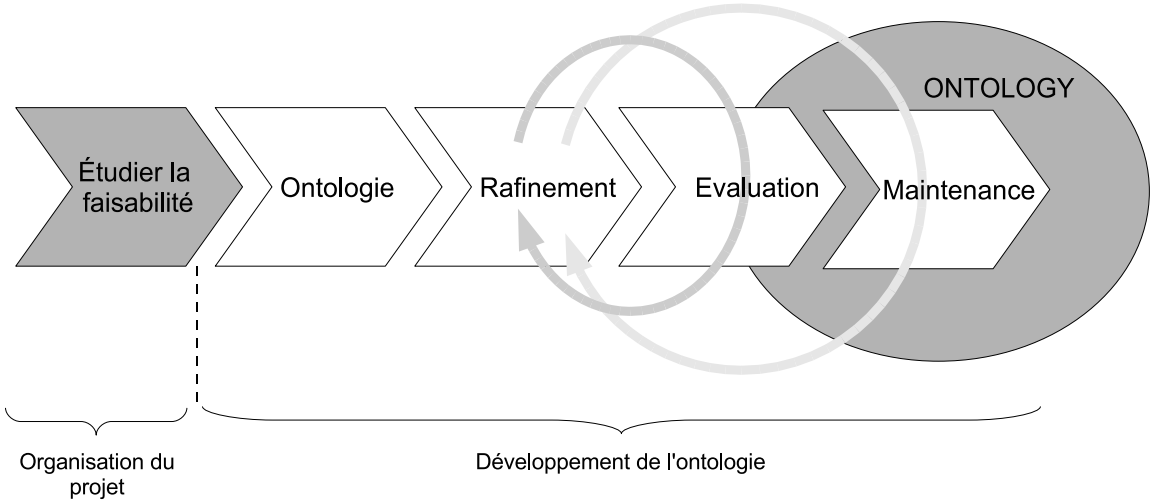


FIG. 2.7 – Méthodologie On-To-Knowledge

précise pas quelle méthode utiliser. En revanche, les auteurs de ON-TO-KNOWLEDGE proposent dans plusieurs publications d'utiliser des méthodes de fouille de textes [Maedche et Staab, 2000; Staab et Maedche, 2001].

Méthodologies évaluant l'ontologie résultante. La méthodologie de Uschold et King [1996] propose d'émettre un jugement technique à propos de l'environnement logiciel intégré dans l'ontologie sans préciser une réelle méthode d'évaluation. A l'opposé, la méthodologie de Grüninger et Fox [1995] définit des théorèmes d'exhaustivité, c'est-à-dire qu'elle signale dans quelles conditions on peut affirmer qu'une réponse à une question de compétence est complète. ON-TO-KNOWLEDGE [Staab *et al.*, 2001] propose de prouver l'utilité de l'ontologie construite en vérifiant que l'ontologie est capable de répondre aux questions de compétence et satisfait le besoin pour lequel elle a été créée. De plus, ils considèrent aussi la possibilité de raffiner l'ontologie pour mieux répondre aux besoins de l'application.

Méthodologies évolutives et incrémentales. Grüninger et Fox [1995] ont été les premiers à proposer une méthodologie évolutive et incrémentale. Cette idée a été reprise par pratiquement toutes les autres méthodologies telles que : KACTUS, METHONTOLOGY ou encore ON-TO-KNOWLEDGE.

Nous présenterons dans la section 3.4 une classification de ces différentes méthodologies afin de donner le choix à l'utilisateur de construire une ontologie d'après les besoins de son application.

2.3 Langages du Web sémantique

Il existe différents langages de représentation des ontologies. La plupart ont été mis en place dans le cadre du *Web sémantique*. Le Web actuel ne traite que le niveau syntaxique (mots clés, cooccurrence de termes ...), *i.e.*, seule la structure du document est définie, sa compréhension n'est destinée qu'aux utilisateurs humains. Seul l'utilisateur peut évaluer si un document correspond aux requêtes qu'il pose. Mais l'augmentation constante des informations sur le Web rend très

difficiles leurs manipulations par les utilisateurs. Pour beaucoup de requêtes, les moteurs de recherche répondent avec des millions de documents. Il est alors difficile pour un être humain de sélectionner les documents pertinents. La nouvelle génération du Web, le Web Sémantique, a pour ambition de lever cette difficulté. Les ressources du Web seront aisément accessibles aussi bien à l'homme qu'à la machine grâce à la représentation sémantique de leurs contenus. Selon Tim Berners-Lee, le Web sémantique désigne : « une extension du Web dans laquelle les informations sont fournies avec une signification bien définie, permettant aux ordinateurs et aux personnes de mieux travailler en coopération » [Berners-Lee, Mai 2001]. Pour présenter la différence entre une page du Web actuel et une page écrite pour le Web sémantique, un exemple d'une page Web sur les formes géométriques est présenté dans la figure 2.8. Dans ce qui suit, cette page sera représentée avec les différentes extensions proposées pour le Web actuel [Fensel *et al.*, 2002] afin d'illustrer les différences entre elles.

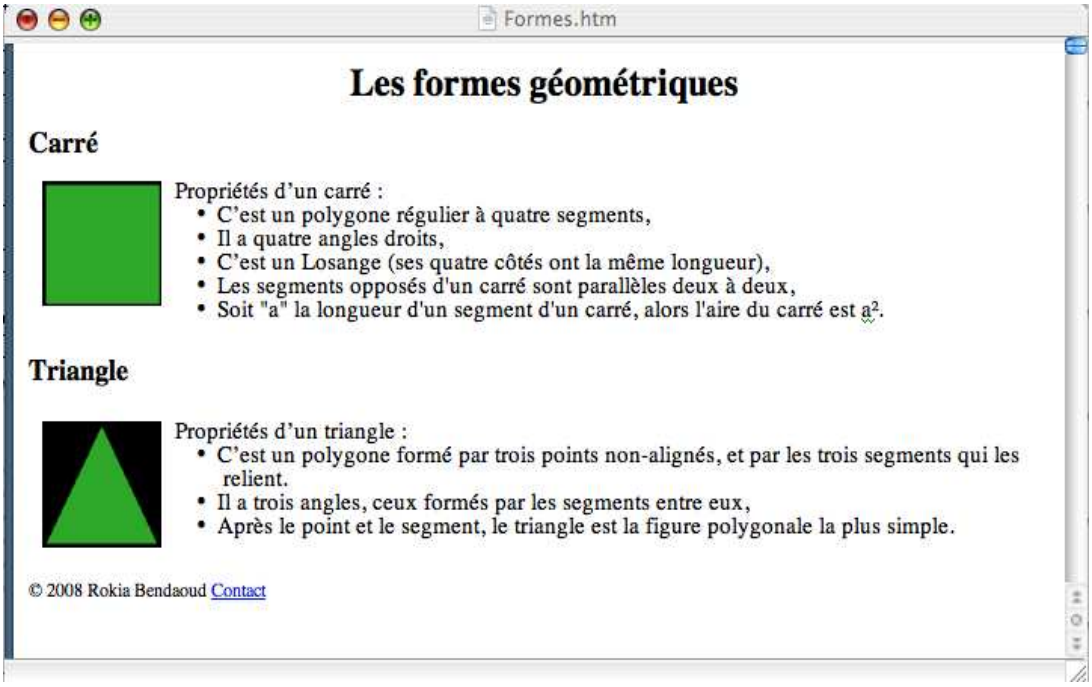


FIG. 2.8 – Un exemple d'un site sur les formes géométriques

Les pages du Web actuelles sont décrites en *Hyper Text Markup Language* (HTML) qui permet la mise en forme graphique des documents sans prendre en compte l'aspect sémantique des mots. Pour rendre ces pages Web compréhensibles par les utilisateurs et les machines, il faut introduire de la sémantique dans ces pages. La figure 2.9 présente les extensions du Web actuel en forme de gâteau. La première extension du Web est constituée par l'Unicode et les URI. Unicode est une norme développée par le Consortium Unicode[14]. Elle vise à donner à tout caractère, de n'importe quel système d'écriture de langue, un nom et un identifiant numérique unique et ce, de manière unifiée quelle que soit la plate-forme informatique ou le logiciel.

Uniform Resource Identifier (URI) est une norme du W3C qui définit chaque ressource du réseau par une chaîne de caractères unique. Les URI ont été proposés par Tim Berners-Lee afin d'unifier les adresses des pages Internet. Par exemple, pour la page sur les formes géométriques,

[14] Unicode : `http://unicode.org/`. Date : 14/07/09

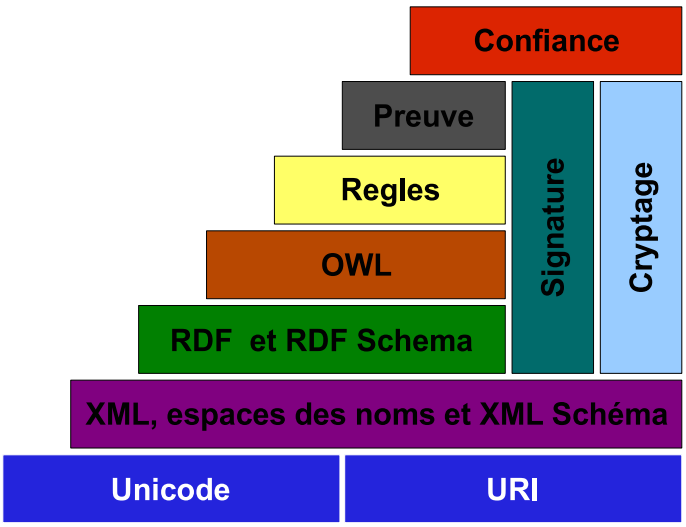


FIG. 2.9 – Le gâteau du Web sémantique

son URI pourrait être : `http://www.mathematique.fr/geometrie$\#$formesgeometriques`

Le langage XML. La première extension du Web est constituée par le langage *eXtensible Markup Language* (XML) qui est une norme W3C[15]. C'est un langage de balises simples avec un format de texte très flexible, mis en place pour l'échange structuré des données sur le Web. XML permet de structurer la page Web en des parties distinctes et pour cela, il inclut des données supplémentaires : des méta-données. La figure 2.10 montre une présentation possible en XML de la forme géométrique `Carré` du site de la figure 2.8. Avec cette structure XML, il sera possible de rechercher parmi l'ensemble des `Formes` celles avec un `nombre de segments` égale à `4`, un `nombre d'angles` égale à `4`, telles que le `type d'angle` est un `angles droits` et le `type de segments` est : `les segments opposés sont parallèles deux à deux`. Nous pouvons ainsi retrouver la forme `Carré`. La recherche d'information en XML est plus efficace qu'une recherche avec des pages en HTML, mais le XML s'intéresse à la structure du document, la recherche reste donc purement syntaxique. C'est l'utilisateur qui définit comment le document est constitué et la machine ne peut pas interpréter la sémantique du domaine. De la sorte, dans notre exemple de la représentation du `Carré`, la machine ne peut pas interpréter le sens de l'héritage. Par exemple, lorsqu'on recherche un `Carré` la machine ne pourra en aucun cas déduire qu'un `Carré` est un `Losange`. Pour répondre à ces questions, la machine doit avoir plus de connaissances. Elle doit définir de façon formelle les concepts mis en œuvre et les relations entre ces concepts, c'est le rôle des ontologies.

2.3.1 Les frameworks RDF et RDFS

La première couche contenant de la sémantique dans le gâteau des langages du Web Sémantique (voir figure 2.9) est composée de *Resource Description Framework* (RDF)[16] et *Resource Description Framework Schema* (RDFS). RDF [Lassila, 1998] est décrit généralement en XML en y ajoutant un formalisme simple (actuellement le plus simple possible [Antoniou et van Harmelen, 2004]) et beaucoup plus expressif que XML. Son formalisme peut être vu comme un formalisme

[15] *eXtensible Markup Language* (XML) : `http://www.w3.org/XML/`. Date : 14/07/09
[16] *Resource Description Framework* (RDF) : `http://www.w3.org/RDF/`. Date : 14/07/09

```
<FigureGeometriques>
      <Forme>
        <Titre> Carré  </Titre>
         <Propriétés>
            <Propriété type= "heritage"> C'est un polygone régulier </Propriété>
            <Propriété type= "heritage"> C'est un losange </Propriété>
            <Propriété type= "nombre d'angles"> 4 </Propriété>
            <Propriété type= "type d'angle"> Angles droits </Propriété>
            <Propriété type= "nombre de segments"> 4 </Propriété>
            <Propriété type= "type de segments"> Les segments opposés sont parallèles
             deux à deux,</Propriété>
            <Propriété type= "Calcul d'aire"> Soit "a" la longueur d'un segment d'un
            carré, alors l'aire du carré est a². </Propriété>
         </Propriétés>
      </Forme>
</FigureGeometriques>
```

FIG. 2.10 – Description d'un Carré dans le langage XML

```
<rdfs:Class rdf:about="http://www.mathematique.fr/geometrie#Carre" />
<rdf:property rdf:about="http://www.schema.org/TR/rdf-schema#Contient">
    <rdfs:domain rdf:resource="http://www.mathematique.fr/geometrie#Carre>
    <rdfs:range rdf:resource="http://www.mathematique.fr/geometrie#Segment">
</rdf:property>
<rdf:property rdf:about="http://www.schema.org/TR/rdf-schema#Contient">
    <rdfs:domain rdf:resource="http://www.mathematique.fr/geometrie#Carre>
    <rdfs:range rdf:resource="http://www.mathematique.fr/geometrie#AngleDroit">
</rdf:property>
<rdfs:subClassOf rdf:resource="#Losange"/>
</rdfs:Class>
```

FIG. 2.11 – Description d'un Carré dans le framework RDFS

de la logique du premier ordre avec uniquement des prédicats d'arité un et deux, le quantificateur existentiel et la conjonction. RDFS [Brickley et Guha, 2000] est une extension de RDF. Il ajoute les définitions de concepts, les hiérarchies d'héritage des concepts et des attributs ainsi que la définition du domaine et du co-domaine d'une relation (restriction de rôle). Nous reprenons les avantages de RDFS présentés dans [Horrocks *et al.*, 2004] sur l'exemple du site des formes géométriques de la figure 2.8. La figure 2.11 présente la représentation en RDFS de la figure géométrique `Carré`. En utilisant RDFS, on peut :

- déclarer des concepts tels que : `Figure`, `Triangle`, `Rectangle`, ...
- déclarer que `Triangle` est un sous-concept de `Figure`,
- déclarer que `ABC` est une instance de `Triangle`,
- déclarer que `estPartieDe` est une relation entre `Coté` (domaine) et `Figure` (co-domaine),
- déclarer que `Aire` est un attribut entre `Figure` (domaine) et `entier` (co-domaine),
- déclarer que `ABC` est une instance de `Triangle` dont l'`Aire` prend comme valeur `4`.

Néanmoins, ce formalisme ne peut pas répondre à certains types de questions du type « Est-ce qu'un objet peut être à la fois `Triangle` et `Carré` ? ».

2.3.2 Logiques de descriptions

Les Logiques de Descriptions (LD) [Baader *et al.*, 2003] sont des familles de formalismes dont les ancêtres sont les réseaux sémantiques, ou plutôt, d'après John Sowa [2000] les réseaux sémantiques définitionnels. Les réseaux sémantiques datent du philosophe Porphire (300 AP-JC), où seule la relation « est-un » était représentée. Les LD sont des langages de représentation des connaissances s'appuyant à la fois sur une représentation structurée, comme les langages à objets ou de frames et, sur une sémantique formellement et précisément définie à la manière des logiques des prédicats. Les entités centrales de ces langages de représentation de connaissances sont les concepts, les instances (individus) et les rôles/attributs (ou propriétés). Un individu est une entité structurée par des attributs. Les attributs sont considérés généralement comme des rôles quand ils représentent une relation entre deux individus, ou comme attributs binaires quand ils indiquent la valeur d'une caractéristique d'un individu. Un concept est constitué d'un ensemble d'individus regroupés par des attributs communs. L'apport des logiques de descriptions peut être résumé en trois points :

- trois éléments fondamentaux : les concepts atomiques, les rôles atomiques et les individus,
- des constructeurs adéquats pour construire des définitions complexes de concepts à partir des concepts atomiques,
- la possibilité de déduire de nouvelles connaissances (i.e. rendre de la connaissance implicite explicite) à partir de définitions de concepts et/ou à partir de la description d'individus (de leurs attributs).

Un système LD traite deux types de connaissances : le niveau terminologique « Terminological Box » (TBOX) et le niveau factuel « Assertional Box » (ABOX). La TBOX contient la terminologie des axiomes tandis que la ABOX ne contient que les assertions. Les axiomes terminologiques peuvent êtres divisés en axiomes d'inclusion et axiomes d'égalité. Un exemple d'axiome d'égalité est la définition d'un carré : `Carré` $\equiv$ `Losange` $\sqcap$ `=4contient.AngleDroit`. Cette formulation signifie qu'un `Carré` est équivalent à un `Losange` à exactement 4 `AnglesDroit`. Un `Carré` est un `Losange` est un exemple d'axiome d'inclusion. Il signifie que toutes les instances de `Carré` sont des instances de `Losange`. Les assertions de l'ABOX peuvent aussi être divisées en deux types : les assertions de concept et les assertions de rôle. Le premier type d'assertion est du type `C(a)` ou simplement `a`. Cela veut dire que l'individu `a` est une instance du concept `C`. Le deuxième type d'assertion est du type `R(a,b)`. Il indique que l'individu `a` est en relation `R` avec l'individu `b`.

Les deux éléments TBOX et ABOX constituent une Base de Connaissances (BC). La sémantique associée à une BC en LD s'exprime généralement sous forme inspirée de la théorie des modèles. La définition suivante est celle de l'interprétation dans les logiques de descriptions [Napoli, 1997].

Définition 4 *Une interprétation $\mathcal{I} = (\Delta^{\mathcal{I}}, .^{I})$ est donnée d'un ensemble $\Delta^{\mathcal{I}}$ appelé domaine de l'interprétation et d'une fonction d'interprétation $.^{I}$ qui fait correspondre à un concept un sous-ensemble de $\Delta^{\mathcal{I}}$ et à un rôle un sous-ensemble de $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ de telle sorte que les équations de la dernière colonne de le tableau 2.1 soient satisfaites.*

Mécanismes de raisonnement

Le principal avantage d'une Base de Connaissances (BC) est de pouvoir être associée à des mécanismes de raisonnement. Ces mécanismes s'appuient sur les deux premières opérations qui servent de briques de base aux suivantes :

- La *Subsomption* (ou test de subsomption) : vérifie qu'un concept `C` subsume un concept `D` noté : $\models$ `D` $\sqsubseteq$ `C`. Ainsi sur l'exemple des figures géométriques la réponse au test de subsomption suivant $\models$ `Triangle` $\sqsubseteq$ `Polygone` est vrai.
- La *Satisfiabilité* détermine s'il n'y a pas de contradiction dans un concept (s'il peut admettre des instances).
- La *classification des concepts* permet de déterminer la position relative de chaque concept dans la hiérarchie de concepts.
- La *classification d'instances* permet de déterminer la liste des concepts auxquels appartient une instance. Soit sur l'exemple des figures géométriques, l'individu `WXYZ` instance de `Losange` tel qu'il existe la relation `contient (WXYZ,=4 AngleDroit)`. Suivant ce mécanisme, il est possible de déterminer que l'individu `WXYZ` est une instance de `Carré`. En effet, l'instance `WXYZ` remplit l'ensemble des conditions nécessaires et suffisantes d'appartenance à ce concept.
- La *recherche d'instances* (en anglais *instance retrieval*) permet de déterminer pour un concept l'ensemble des individus qui en sont instances.

L'efficacité de certains mécanismes de raisonnement, plus complexes, est conditionnée par la DL choisie. Parmi ceux-là nous citerons :

- La *recherche du concept le plus spécifique* (en anglais *most specific concept*) qui consiste à déterminer pour un concept (ou un individu) quel est le concept le plus spécifique qui le subsume (ou quel est le concept le plus spécifique dont il est instance).
- La *recherche du subsumant commun le plus spécifique* (en anglais *least common subsumer*) qui recherche le concept le plus spécifique qui subsume en même temps deux concepts donnés (ou dont deux individus donnés sont instances).

2.3.3 Web Ontology Language

Un langage d'ontologie doit nous permettre d'écrire explicitement et dans une conceptualisation formelle un modèle du domaine [Antoniou et van Harmelen, 2004]. Ce langage doit avoir :

- une syntaxe,
- une sémantique formelle,
- un support de raisonnement efficace,
- une expressivité suffisante.

Une sémantique dite formelle signifie que cette sémantique est interprétée de la même manière quelle que soit la personne ou la machine qui l'interprète. La sémantique est un pré-requis pour pouvoir raisonner. Un support de raisonnement efficace est très important pour pouvoir :

- Vérifier la consistance de l'ontologie et des connaissances ;
- Vérifier les relations incohérentes entre les concepts ;
- Automatiser la classification des instances dans les concepts.

L'automatisation du raisonnement permet de vérifier la consistance de l'ontologie beaucoup plus rapidement qu'une vérification manuelle. Cela permet aussi de vérifier des ontologies écrites par plusieurs auteurs et d'éliminer les répétitions ou les incohérences. Ainsi, l'automatisation du raisonnement nous permet de construire et de partager des ontologies à partir de différentes ressources.

Nom du constructeur	syntaxe en LD	syntaxe OWL	sémantique associée
Concept	C	C (URI)	$\mathtt{C}^I \subseteq \Delta^I$
Concept universel	$\top$	`owl :Thing`	$\top^I = \Delta^I$
Bottom	$\bot$	`owl :Nothing`	$\bot^I = \emptyset$
Intersection	$C \sqcap D$	`intersectionOf C D`	$(\mathtt{C} \sqcap \mathtt{D})^I = \mathtt{C}^I \cap \mathtt{D}^I$
Union	$C \sqcup D$	`unionOf C D`	$(\mathtt{C} \sqcup \mathtt{D})^I = \mathtt{C}^I \cup \mathtt{D}^I$
Négation	$\neg C$	`complementOf C`	$(\neg\ \mathtt{C})^I = \Delta^I \setminus \mathtt{C}^I$
Énumération	$\{a, b \ldots\}$	`oneOf a b ...`	$\{a, b \ldots\}^I = \{a^I, b^I \ldots\}$
Quantificateur existentiel	$\exists R.C$	`restriction(R someValuesFrom(C)`	$(\exists R.C)^I = \{x \mid \exists y.\ (x,y) \in R^I \wedge y \in C^I\}$
Quantificateur universel	$\forall R.C$	`restriction(R allValuesFrom(C)`	$(\forall R.C)^I = \{x \mid \forall y.\ (x,y) \in R^I \rightarrow y \in C^I\}$
Restriction à une valeur	$\ni R.a$ *ou* $R.\{a\}$	`restriction (R hasValues(a)`	$(\ni R.a)^I = \{x \mid (x, a^I) \in R^I\}$
Restrictions non qualifiées de cardinalité	$= nR$	`restriction(R cardinality(C)`	$(= nR)^I = \{x \mid \mathtt{card}\{y \mid (x,y) \in R^I\} = n\}$
	$\geq nR$	`restriction(R minCardinality(C)`	$(\geq nR)^I = \{x \mid \mathtt{card}\{y \mid (x,y) \in R^I\} \geq n\}$
	$\leq nR$	`restriction (R maxCardinality(C)`	$(\leq nR)^I = \{x \mid \mathtt{card}\{y \mid (x,y) \in R^I\} \leq n\}$
Quantificateur existentiel	$\exists S.T$	`restriction (S someValuesFrom(T)`	$(\exists S.T)^I = \{x \mid \exists y.\ (x,y) \in S^I \wedge\ \ y \in T^I\}$
Quantificateur universel	$\forall S.T$	`restriction (S allValuesFrom(T)`	$(\forall S.T)^I = \{x \mid \forall y.\ (x,y) \in S^I \rightarrow y \in T^I\}$
Restriction à une valeur	$\ni S.a$ *ou* $S.\{a\}$	`restriction (S hasValues(a)`	$(\ni S.a)^I = \{x \mid (x, a^{\mathcal{D}}) \in S^I\}$
Restriction non qualifiée de cardinalité	$= nS$	`restriction (S cardinality(T)`	$(= nS)^I = \{x \mid \mathtt{card}\{y \mid (x,y) \in S^I\} = n\}$
	$\geq nS$	`restriction (S minCardinality(T)`	$(\geq nS)^I = \{x \mid \mathtt{card}\{y \mid (x,y) \in S^I\} \geq n\}$
	$\leq nS$	`restriction (S maxCardinality(T)`	$(\leq nS)^I = \{x \mid \mathtt{card}\{y \mid (x,y) \in S^I\} \leq n\}$

TAB. 2.1 – Constructeurs de concepts en Logique de Descriptions LD et leurs correspondances en OWL. `C` et `D` sont des concepts, `T` est un concept particulier qui correspond à un type de données (*Datatype* en OWL), n est un nombre, `a` et `b` sont des individus, `R` un rôle (*ObjectProperty* en OWL) et `S` un attribut dont le co-domaine correspond à un concept de même type que `T` (un attribut de données ou *DatatypeProperty* en OWL).

Le langage de représentation des ontologies dans le Web sémantique est le langage *Web Ontology Language* (OWL) qui est décrit dans plusieurs recommandations W3C (en particulier [WOWG, Février 2004a; Février 2004b]). Le langage OWL repose sur les frameworks RDF et RDFS, mais il est plus expressif. Horrocks et al. [2004] ont présenté les avantages d'OWL par rapport aux frameworks RDF et RDFS. En plus de ce que permet d'exprimer RDFS, OWL permet de :

- déclarer que les concepts `Triangle` et `Carré` sont disjoints,
- déclarer que les instances `ABC` et `ABD` sont distinctes,
- déclarer que `contient` est la relation inverse de la relation `estPartieDe`,
- déclarer que le concept `Triangle` a exactement 3 valeurs de la relation `contient` et du co-domaine `Segment`,
- déclarer que le concept `Triangle` est défini précisément en tant que sous-concept du concept `Figure` avec la relation `contient` égale à 3 et le co-domaine `Angle`,
- déclarer l'attribut `Aire` est un attribut fonctionnel (chaque individu a au plus une valeur de l'attribut `Aire`).

Le tableau 2.1 présente les différents éléments des logiques de descriptions, leur écriture en OWL et leurs interprétations. La figure 2.12 présente la description OWL du notre exemple de la figure 2.4.

```
<owl:Class rdf:about="#Carré">
    <owl:equivalentClass>
      <owl:Restriction>
        <owl:onProperty rdf:resource="#Contient"/>
        <owl11:onClass rdf:resource="#Segment"/>
        <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">4</owl:cardinality>
      </owl:Restriction>
    </owl:equivalentClass>
    <owl:equivalentClass>
      <owl:Restriction>
        <owl:onProperty rdf:resource="#Contient"/>
        <owl11:onClass rdf:resource="#AngleDroit"/>
        <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">4</owl:cardinality>
      </owl:Restriction>
    </owl:equivalentClass>
    <rdfs:subClassOf rdf:resource="#Losange"/>
    <owl:disjointWith rdf:resource="#Triangle"/>
</owl:Class>
```

FIG. 2.12 – Description d'un `Carré` dans le langage OWL

Méthodologies représentant formellement les ontologies. Deux types de langages ont été utilisés pour représenter formellement une ontologie : la logique du premier ordre et les logiques de descriptions.

Logique du premier ordre. La première méthodologie à représenter chaque élément de l'ontologie en un langage formel est la méthodologie de Grüninger et Fox [1995]. Cette représen-

tation est faite en logique de premier ordre. Chaque objet du domaine est représenté en tant que constante ou variable du langage. Les attributs des objets sont représentés par les prédicats unaires et les relations entre concepts par des prédicats d'ordre n. En reprenant l'exemple sur les formes géométriques, `$Carré` est un concept, `contient($Figure,$Segment)` est une relation entre `Figure` et `Segment`. Ainsi les questions auxquelles l'ontologie doit répondre sont réécrites dans la logique du premier ordre.

Logiques de descriptions Les deux méthodologies qui représentent chaque élément du schéma de l'ontologie en logique de descriptions puis l'implémentent en OWL sont METHONTOLOGY [Gómez-pérez *et al.*, 2004] et ON-TO-KNOWLEDGE [Staab *et al.*, 2001]. En LD, les objets sont des instances, les classes sont des concepts et les relations entre concepts sont des rôles. La figure 2.13 présente l'application de la méthodologie METHONTOLOGY sur le domaine des formes géométriques.

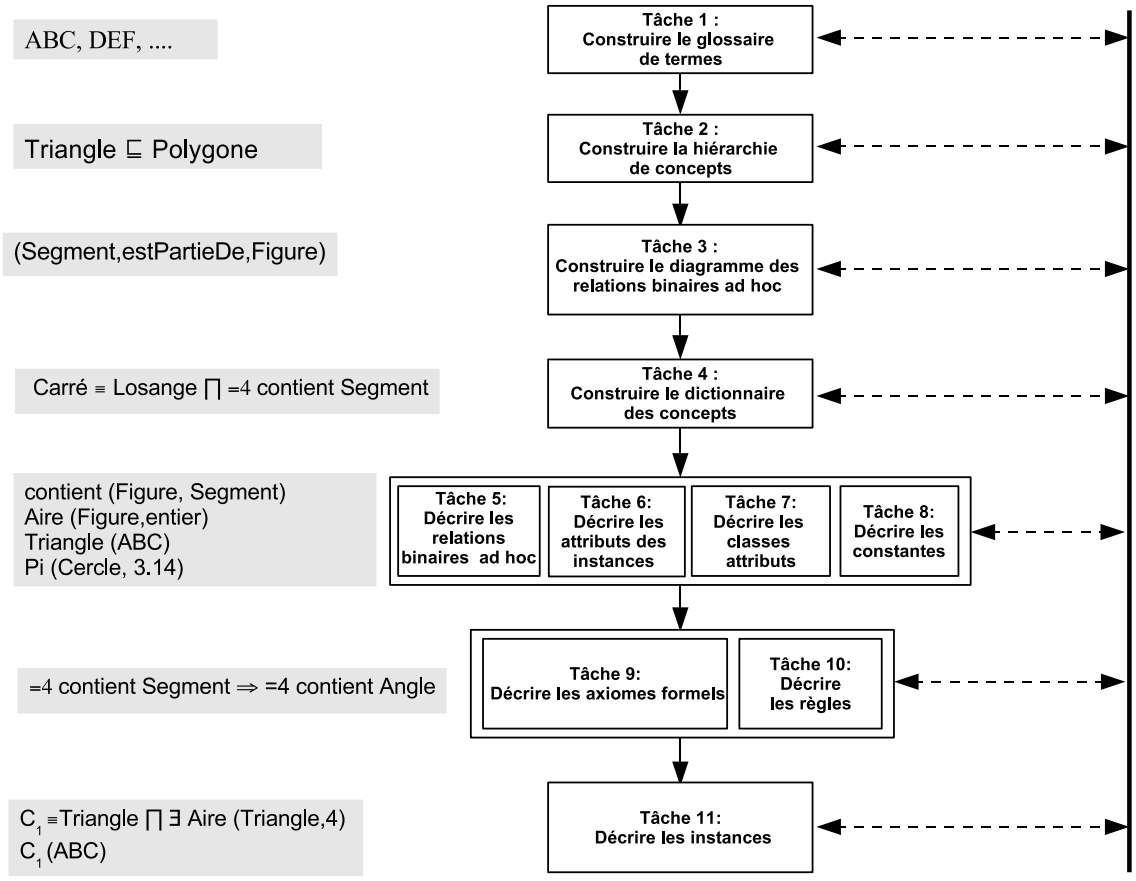


FIG. 2.13 – Application de la méthodologie METHONTOLOGY sur l'exemple des « figures géométriques »

2.3.4 Les environnements de construction d'ontologies et les outils de raisonnement

Afin de faciliter la construction d'ontologies implémentées dans un langage tel que OWL, des environnements d'ontologies ont été proposés contenant un ensemble d'outils autour d'un éditeur. L'éditeur le plus utilisé à l'heure actuelle est l'éditeur PROTÉGÉ [Noy *et al.*, 2001]. Il offre la possibilité d'intégrer des *plugins*, c'est-à-dire des extensions à l'éditeur, par exemple, le plugin « OWLViz » qui permet de visualiser l'ontologie. Ce type de fonctionnalité est aussi présent au sein de l'éditeur de connaissances OILED [Bechhofer *et al.*, 2003]. Il existe aussi des éditeurs qui ont été réalisés pour développer une ontologie d'après une méthodologie déjà définie, par exemple, ONTOEDIT [Sure *et al.*, 2002] qui suit la méthodologie ON-TO-KNOWLEDGE, ou encore, WEBODE [Arpírez *et al.*, 2003] qui suit la méthodologie METHONTOLOGY.

Nous citons aussi « Jena »[17], un toolkit Java pour l'implémentation d'une ontologie en OWL. Jena permet de passer d'un fichier texte où sont définis les concepts d'une ontologie à un document en OWL lisible par un éditeur d'ontologies.

Outils de raisonnement

Pour pouvoir effectuer des raisonnements décrits en OWL et pour les DL en général, des moteurs d'inférences ont été développés. Ces moteurs d'inférences reposent tous sur un test appelé « test de la satisfiabilité d'un concept ». Pour effectuer ce test, ces moteurs utilisent la méthode des tableaux sémantiques [Horrocks et Patel-Schneider, 1998]. Parmi ces moteurs d'inférences, nous citons FACT [Horrocks, 1998] et RACER [Haarslev et Möller, 2001]. Ils permettent tous les deux de raisonner sur OWL et sont développés en JAVA.

Il existe aussi plusieurs langages de requêtes permettant d'interroger une base de connaissances. Parmi ces langages, nous citons le langage « Simple Protocol And Rdf Query Language » (SPARQL)[18]. Ce langage peut être utilisé grâce à des outils comme KOWL, un serveur de connaissance développé dans le cadre du projet Kasimir (`http://labotalc.loria.fr/$~$kasimir/`). Implémentant le protocole et le langage de requête SPARQL, il repose sur le framework Jena et le raisonneur PELLET pour manipuler les ontologies OWL [Badra *et al.*, 2008].

[17] Jena : `http://jena.sourceforge.net/tutorial/RDF$\_$API/index.html`. Date : 14/07/09

[18] Simple Protocol And Rdf Query Language (SPARQL) : `http://www.w3.org/TR/rdf-sparql-query/`. Date : 14/07/09

Chapitre 3

Extraction de connaissances

Sommaire

Introduction

Dans ce chapitre, nous commençons par présenter le processus d'extraction de connaissances à partir de données, puis nous détaillons l'adaptation de ce processus aux bases de données et aux corpus de textes. Ensuite, nous présentons les méthodes de fouille de données que nous avons choisies : l'Analyse Formelle de Concepts (AFC) et l'Analyse Relationnelle de Concepts (ARC). L'AFC construit des treillis de concepts à partir de tableaux binaires objets/attributs binaires.

Ces treillis de concepts regroupent des objets en fonction des attributs binaires qu'ils partagent. Son extension, l'Analyse Relationnelle de Concepts (ARC), regroupe un ensemble d'objets non seulement à partir d'attributs binaires mais aussi à partir des liens inter-objets (instances de relations).

3.1 Processus d'extraction de connaissances

Le processus permettant de passer de données brutes à des connaissances est le processus d'Extraction de Connaissances à partir de Données (ECD). Ce processus est itératif et interactif et consiste à rechercher des unités de connaissances à partir de données brutes. Ces unités de connaissances doivent être « non triviales, potentiellement utiles, compréhensibles et réutilisables » [Fayyad *et al.*, 1996].

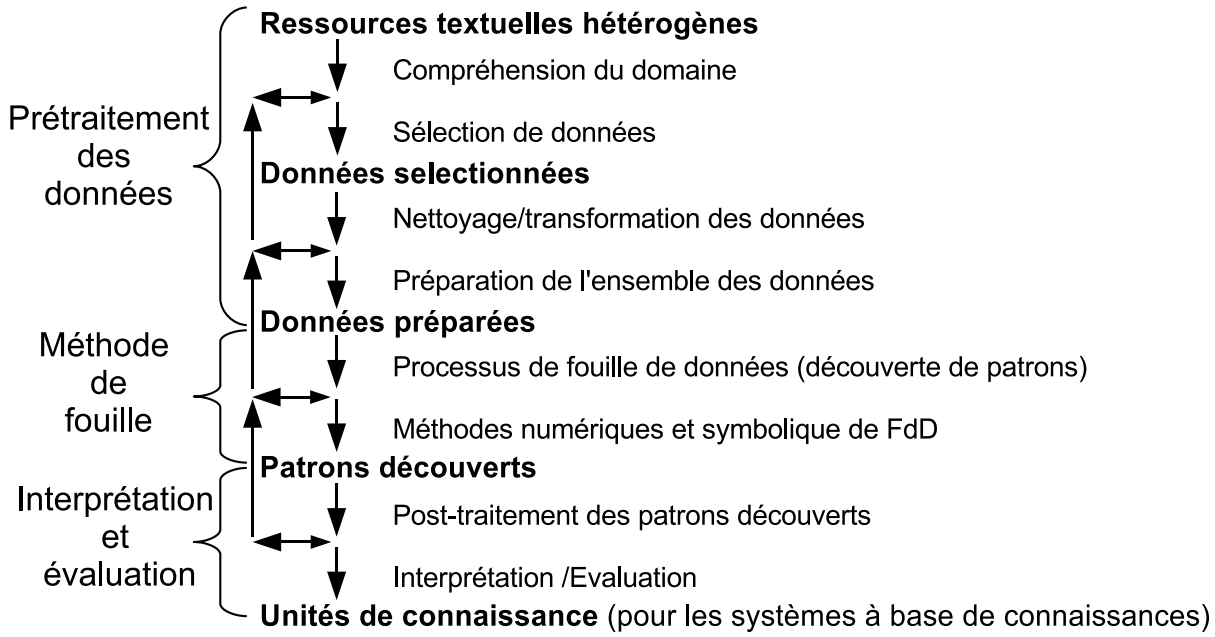


FIG. 3.1 – Le processus d'extraction de connaissances à partir de données : des données aux connaissances

La figure 3.1 présente en détail les étapes du processus d'ECD pour passer des données aux connaissances. Ce processus est composé de trois grandes étapes : (1) le prétraitement des ressources de données, (2) l'application de la méthode de fouille où des méthodes symboliques et numériques sont appliquées pour extraire des patrons. Et enfin (3) l'interprétation et l'évaluation des résultats par les experts du domaine afin d'extraire les unités de connaissances.

Pour mieux expliquer ces étapes, Fayyad [1996] les découpe en neuf sous-étapes en proposant pour chaque sous-étape, les types de questions auxquels elle doit répondre.

(1) développer une compréhension du domaine d'application ainsi que de la connaissance préalablement connue et considérée comme pertinente par les experts. Elle doit répondre aux questions suivantes : Quel est le goulot d'étranglement de ce domaine spécifique ? Quelles étapes sont automatiques et lesquelles doivent absolument être manuelles ? Quels sont les objectifs du

processus ? Et avec quels critères va-t-on déterminer s'ils sont atteints ? A quoi va servir le produit final de ce processus ? classification, visualisation, exploration, ou autres ?

(2) sélection de l'ensemble des données. Dans cette sous-étape, la nature des données doit être décidée. Les données sont-elles homogènes ou hétérogènes ? L'ensemble des données est-il statique ou dynamique ?...

(3) nettoyage des données et prétraitement. Dans cette sous-étape il faut enlever le bruit, compléter le vide des données manquantes et normaliser les données.

(4) réduction des données et transformation. Cela consiste à retrouver les caractéristiques utiles des données selon le but du processus. Cette sous-étape doit répondre aux questions suivantes : Quel est le volume de données traitées ? Quelles variables décrivent ces données ?

(5) choisir le but de la méthode de fouille de données. Quel est le but du processus d'ECD ? Classification, catégorisation, résumé des données, création de modèles des données...

(6) choix de l'algorithme de fouille. Cette sous-étape est très liée au but de la fouille de données ainsi qu'à la compréhension de l'utilisateur final. Les patrons de l'algorithme de fouille doivent être compréhensibles pour pouvoir être interprétés.

(7) l'étape de la fouille de données. Elle consiste à rechercher des patrons intéressants ainsi que leur structuration.

(8) évaluation du résultat de la sous-étape 7. Dans cette sous-étape, les experts doivent définir ce qui est considéré comme de la connaissance. Cette tâche est très difficile et atteindre des résultats acceptables implique l'utilisation de filtres, tels que des filtres statistiques, des filtres de simplicité...

(9) la consolidation des connaissances découvertes. Cette sous-étape implique l'intégration de la nouvelle connaissance avec les connaissances déjà connues, ainsi que la gestion des conflits au cas où une nouvelle connaissance est en contradiction avec une connaissance déjà connue.

Le processus d'ECD a été adapté aux BD puis aux corpus de textes. Dans les sous-sections suivantes, nous détaillons chaque étape de l'application de ce processus à ces deux différentes ressources.

3.1.1 Processus d'extraction de connaissances à partir de bases de données

Le processus d'Extraction de Connaissances à partir de Bases de Données (ECBD) a été mis en place par Fayyad et al. [1996] et est détaillé dans plusieurs ouvrages tels que [Brachman et Anand, 1996; Dunham, 2002]. Ce processus, décrit dans la figure 3.2, s'articule autour de quatre composantes [Simon, 2000] :

1. une ou plusieurs bases de données et leurs systèmes de gestion. Un système d'ECBD doit être capable de traiter des masses de données volumineuses. Le passage à l'échelle d'une petite à une grande application doit se faire de façon transparente pour l'analyste ;
2. un Système à Base de Connaissances (SBC) qui permet à la fois la gestion des connaissances et la résolution de problèmes liés au domaine des données. Le SBC utilise une base de connaissances (une ontologie du domaine) qui est enrichie grâce aux nouvelles connaissances inférées par le SBC ;
3. un système de fouille de données (FDD) pouvant s'appuyer sur des techniques symboliques comme l'extraction de règles d'association [Agrawal et Srikant, 1995], la classification par treillis de concepts [Ganter et Wille, 1999] ou l'induction par des techniques d'arbres de décision [Breiman *et al.*, 1984]. Le système FDD peut également s'appuyer sur des techniques numériques telles que l'analyse des données ou des techniques statistiques [Curran et Moens, 2002] ;

4. une interface se chargeant des interactions avec l'analyste et de la visualisation des résultats. L'analyste est chargé de guider les recherches et de valider les connaissances extraites. Il est donc au centre de ces quatre composantes.

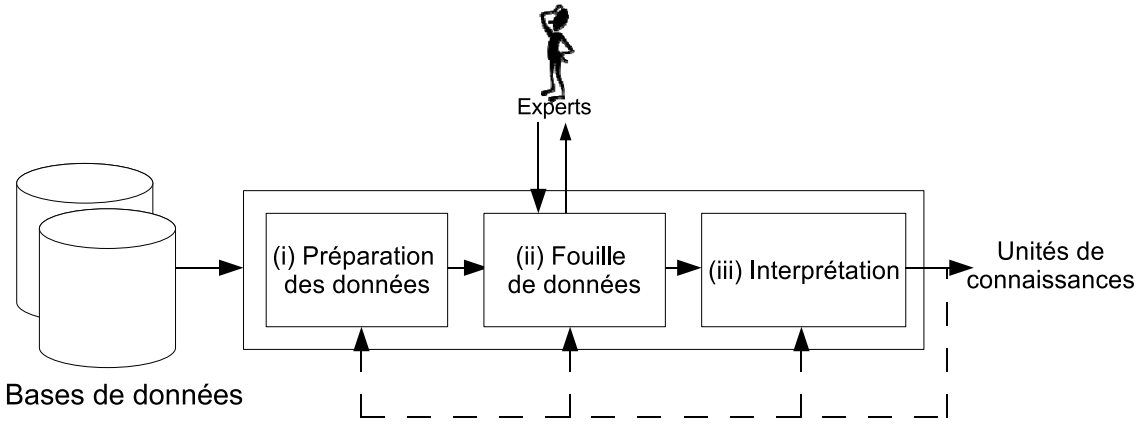


FIG. 3.2 – Le processus d'Extraction de connaissances à partir de bases de données (ECBD)

Préparation des données. Les données dans les bases de données sont généralement brutes et difficilement exploitables par les méthodes de fouille. La première étape du processus d'ECBD consiste en plusieurs opérations qui permettent d'améliorer la qualité des données (tout au moins en vue de la méthode de fouille choisie) et ainsi d'améliorer l'efficacité du processus d'ECBD. Ces opérations sont de différents types où la première étape consiste à intégrer les données, c'est-à-dire prendre en compte plusieurs bases de données. Ensuite l'ordre des autres opérations varie selon la méthode de fouille choisie. Ces opérations sont : le nettoyage des données, la réduction des données et la transformation des données. Ces opérations sont aussi inter-dépendantes, comme par exemple l'opération d'intégration de données qui peut nécessiter le nettoyage des données ou leur transformation. Le nettoyage peut lui-même conduire à une réduction des données. Nous présentons dans les paragraphes suivants ces différentes étapes de prétraitement.

Système d'intégration de données. Un système d'intégration de données est un système qui permet à un utilisateur ou à une machine d'accéder à des données hétérogènes de manière uniforme et transparente, ce qui signifie que l'utilisateur ne verra aucune différence entre l'accès à une ou plusieurs bases de données. Ce système doit alors uniformiser les données ainsi que leur localisation.

Nettoyage des données. Comme présenté précédemment, les données sont souvent brutes, incomplètes, bruitées, voir incohérentes. Ainsi, la manipulation de ces données est très difficile et pour y remédier, une étape de nettoyage est indispensable. Cette étape consiste à combler et à corriger les données manquantes ou incohérentes, car l'efficacité de certains algorithmes de fouille est très sensible aux valeurs manquantes. Différentes approches peuvent être adoptées [Han et Kamber, 2001] :

– ignorer les tuples dans lesquels des valeurs manquent (un tuple dans une base de données est décrit par une ligne de la base : une entrée). Cela peut s'avérer problématique dans les bases de données de petite taille ;

- remplacer les valeurs manquantes par une valeur particulière, par exemple « Unknown » ou « ? ». Néanmoins, ce type de méthodes peut biaiser les algorithmes de fouille qui pourraient considérer ces données comme faisant partie de concepts intéressants ;
- remplacer les valeurs manquantes par une valeur arbitraire, le but ici est d'influencer le moins possible les algorithmes de fouille. Ces valeurs pourraient être décrites par la moyenne des autres valeurs ou encore une valeur probable prédite par des méthodes d'inférence, de régression, d'induction sur la base d'autres données.

Après avoir comblé le manque dans les valeurs des bases de données, reste à éliminer le bruit dans les données. Pour cela, des méthodes de filtrage et de lissage peuvent être appliquées pour diminuer les effets de ce bruit.

Sélection des données. Lorsque la taille des données est très importante et afin de faciliter le travail, des méthodes de sélection peuvent être appliquées sur des sous-ensembles de données sans en altérer la déscription originale. Il existe deux familles de méthodes pour la sélection de données [Guyon et Elisseeff, 2003] :

- les méthodes de filtrage qui estiment l'intérêt de chaque attribut dans le jeu de données puis qui les classifient afin d'en supprimer les moins intéressants. Plusieurs méthodes peuvent être utilisées pour attribuer un score d'intérêt à un attribut comme par exemple le gain d'information [Kohavi et John, 1997] ou le calcul des dépendances entre les attributs [Yu et Liu, 2004]. Le principal désavantage de ces méthodes est qu'elles sont indépendantes de la méthode de fouille utilisée et ainsi elles conduisent à estimer l'intérêt des attributs selon des critères différents de ceux utilisés par la méthode de fouille.
- les méthodes enveloppantes et intégrées sont quant à elles dépendantes de la méthode de fouille considérée. Leur principe est de tester la méthode de fouille sur chaque sous-groupe de données, puis de les comparer et de ne garder que ceux qui offrent les meilleurs résultats. Ces méthodes pourraient presque être considérées elles-mêmes comme des méthodes de fouille. Le principal désavantage de ces méthodes est qu'elles sont coûteuses en temps de calcul. Les algorithmes génétiques sont par exemple utilisés pour ce type de méthode de sélection de données [Saeys *et al.*, 2007].

Transformation des données. La transformation des données consiste à mettre en forme les données pour la méthode de fouille choisie. Il existe plusieurs types de transformations, nous citons par exemple l'échelonnage (en anglais *scaling*) détaillé dans la sous-section 3.3.5 qui consiste à passer de données non binaires à des données binaires. Par exemple, créer des intervalles qui permettent de passer de données multivaluées telles que la mesure de l'`Aire` d'une `Figure` à des données monovaluées : la mesure de l'`Aire` peut être découpée en intervalles : [0,2], [2,4],... La transformation peut aussi être une généralisation ou une spécialisation qui, en s'appuyant sur une hiérarchie de termes ou de concepts, permet de remplacer des termes par leurs parents dans la hiérarchie afin de restreindre le nombre d'attributs. Par exemple, les termes `Triangle isocèle` et `Triangle équilatéral` peuvent être remplacés par leur ancêtre `Triangle` [Carpineto et Romano, 2004]. L'agrégation est une transformation également intéressante lorsque les données peuvent être résumées ou agrégées pour être étudiées dans une dimension différente. Par exemple, seul le nombre de `Segment` est utilisé pour définir une `Figure` et déduire le nombre `Angle` qui est toujours le même que le nombre de `Segment`. Le lissage qui revient à appliquer aux données une fonction d'approximation dans l'objectif d'éliminer les phénomènes locaux et de mettre en évidence les caractéristiques générales est aussi un autre exemple de transformation de données [Han et Kamber, 2001].

3.1.2 Processus d'extraction de connaissances à partir de textes

Les corpus de textes constituent une ressource importante de connaissances. L'extraction de connaissances qu'ils contiennent est directement reliée au Web Sémantique. Avec l'apparition de millions de pages Web, la nécessité d'extraire de la connaissance à partir de ces pages a paru évidente. Maedche et Staab [2000] définissent la construction d'une ontologie de domaine à partir de corpus de textes comme étant une modélisation du domaine à partir des données textuelles. La construction d'ontologie à partir de textes peut être vue comme un processus inversé (voir figure 3.3). En écrivant les textes, les experts du domaine ont une certaine vision de leur domaine. Lors de l'extraction de connaissances du corpus de textes, il faudrait essayer d'extraire ce même modèle imaginé par les experts. Cette tâche de reproduction du modèle initial des auteurs est un challenge très complexe, car seule une partie de ce modèle est exprimée dans les textes.

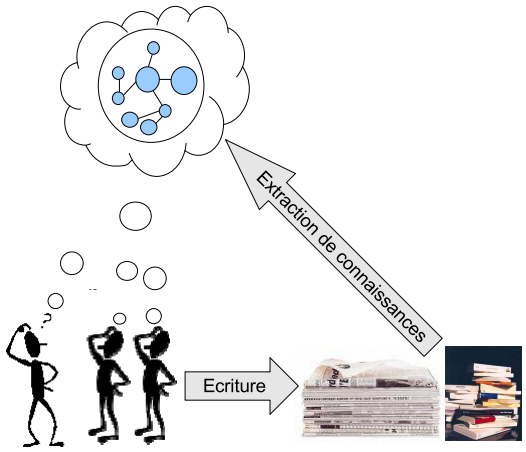


FIG. 3.3 – Construction d'ontologie à partir de textes : livres, journaux ...

Pour illustrer la différence entre ce que pense un auteur et ce qui est décrit dans les textes, Sowa [2000] a proposé un triangle des sens qui illustre les différentes relations entre les symboles, les objets et les sens. La figure 3.4 en présente un exemple. En bas à gauche, il y a l'image d'un chien appelé « Bob ». En bas à droite, il y a le symbole représentant le nom de ce chien. Et en haut, il y a le concept de « Bob », *i.e.* sa représentation réelle. Sowa explique que lorsque l'auteur veut parler du chien « Bob », il n'utilisera que le symbole « Bob » en entraînant une perte d'information. Ainsi, le processus d'extraction de connaissances à partir de textes a besoin en plus, d'une étape supplémentaire qui est l'interprétation des symboles extraits du texte. Les textes sont composés par des ensembles de termes et les termes ne sont que des chaînes de caractères qu'il faut interpréter.

L'extraction de connaissances à partir de textes (ECT) est issue de l'extraction de connaissances à partir de bases de données mais présente un certain nombre de spécificités par rapport à ce processus [Toussaint, 2004]. Nous donnons une définition proposée par [Toussaint, 2004] qui adapte la définition d'ECBD de Fayyad [1996].

Définition 5 (L'extraction de connaissances à partir de textes) *L'extraction de connaissances à partir de textes est un processus non trivial qui construit un modèle de connaissances valide, nouveau, potentiellement utile et au final compréhensible, à partir de textes bruts.*

Le processus d'ECT consiste à extraire des éléments dans une collection de textes qui peuvent

être interprétés comme des éléments de connaissances par un expert du domaine [Feldman et Sanger, 2007]. Ces connaissances nous permettent de construire une ontologie du domaine. Le processus d'ECT se compose de trois grandes parties : la première est la modélisation qui consiste à préparer et à isoler des expressions ou des termes reflétant le contenu des textes à l'aide d'outils de Traitement Automatique de la Langue Naturelle (TALN). La deuxième partie est la fouille de données qui est constituée d'outils de classification d'objets. La troisième partie est l'interprétation et la validation de l'analyste, car dans le cadre de la veille technologique, nous supposons que nos outils sont dédiés à un analyste, expert du domaine en charge de cette activité de veille.

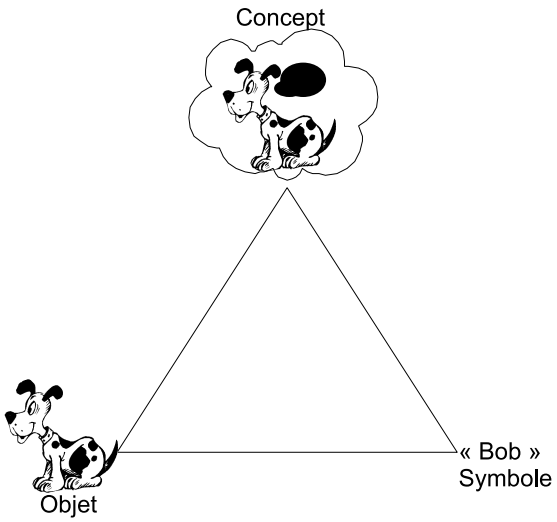


FIG. 3.4 – Le triangle de sens

Le processus d'Extraction de Connaissances à partir de Textes (ECT) suit les mêmes étapes que celui de l'ECBD, avec un prétraitement supplémentaire. La figure 3.5 adaptée de [Fayyad *et al.*, 1996] présente ce processus. Le processus d'ECT comme le processus d'ECBD est itératif et incrémental. Il se décompose en six étapes :

La sélection. Il est nécessaire, avant de construire une ontologie, de constituer l'ensemble des documents sur lequel repose cette élaboration [Condamines, 2005] (voir la sous-section 2.1.1). Un corpus de textes constitue simplement n'importe quel regroupement de documents textuels sur lequel on souhaite extraire des connaissances par des méthodes de fouille. Le nombre de documents varie de quelques milliers de textes à plusieurs millions. Le corpus de textes peut être statique, *i.e.*, le nombre de documents ne varie pas ou dynamique. Un corpus dynamique permet de tester les performances des systèmes de fouille de textes par le passage à l'échelle [Feldman et Sanger, 2007]. Nous donnons l'exemple de PubMed[19] avec 18 millions de résumés d'articles scientifiques qui constitue l'un des plus important corpus en ligne en langue anglaise.

Le prétraitement. Le but de cette étape est d'éliminer le bruit et de combler les manques dans les données. L'élimination de bruit est nécessaire car les corpus de textes incluent plusieurs phrases de styles qui ne contiennent aucune donnée intéressante pour le domaine. Par exemple,

[19]PubMed : `http://www.ncbi.nlm.nih.gov/pubmed/`. Date : 14/07/09

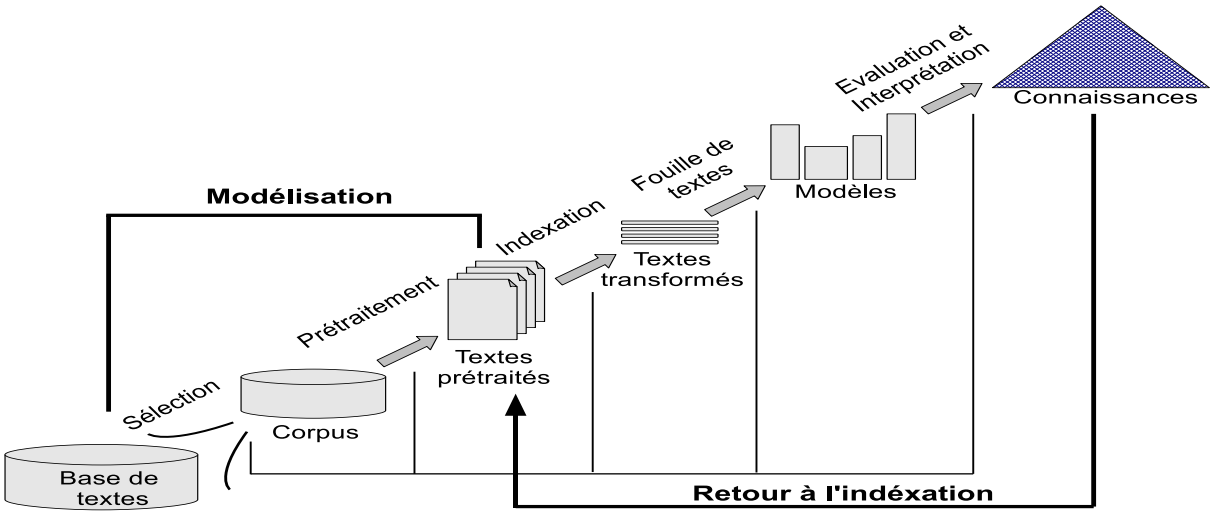


FIG. 3.5 – Schéma global de l'extraction de connaissances à partir de textes (ECT)

toutes les phrases introductives comme « *Dans cet article, nous présentons les notions nécessaires à la compréhension de notre travail* ». Comme nous l'avons expliqué sur le triangle des sens (voir figure 3.4), une partie de l'information est perdue entre ce que l'expert pense et ce qu'il écrit dans les textes. Pour remédier à cette perte d'information, une étape de comblement des manques dans les données est obligatoire. Les prétraitements appliqués dans cette thèse sont présentés dans le chapitre 4.

La transformation. Cette étape consiste à rechercher les meilleures structures pour décrire les données en fonction des outils de fouille de données. Les techniques utilisées sont les mêmes que celles appliquées aux bases de données et présentées dans la sous-section 3.1.

La fouille de données. Cette étape consiste à définir des tâches de classification et de recherche de modèles ainsi que leurs paramètres appropriés. Cette partie est développée dans la sous-section 3.2.5.

L'interprétation et l'évaluation. Pendant cette étape, les unités extraites sont analysées par un expert du domaine. Cette étape est totalement manuelle car elle ne peut être automatisée, seuls les experts du domaine peuvent identifier une unité comme étant une unité de connaissance. Après validation, les unités de connaissance sont stockées dans une base de connaissances, l'ontologie du domaine. La chaîne de traitement du processus de fouille rend le travail de l'analyste efficace en lui donnant accès prioritairement à des connaissances *rares* et/ou potentiellement *nouvelles*. L'analyste doit avoir le rôle de prise de décision finale pour valider ou pas les connaissances extraites, filtrées et jugées pertinentes par rapport au domaine des textes fouillés.

3.2 Méthodes d'extraction de connaissances à partir de ressources textuelles

Dans les paragraphes suivants, nous présentons un panorama des méthodes de transformation des données textuelles puis des méthodes de fouille proposées. Les méthodes de transformation consistent à extraire les données qui seront présentées aux méthodes de fouille. Nous distinguons ces différentes données en les divisant en trois types : (1) l'identification des termes du domaine, (2) la détection des attributs binaires de terme extraits sous forme de paires `(terme,attributs_binaire)` et enfin, (3) la détection des liens entre termes (les relations) extraits sous forme de triplets $(\texttt{terme}_1,\texttt{relation},\texttt{terme}_2)$. Nous présentons dans les sous-sections suivantes un panorama des méthodes proposées dans la littérature pour extraire de la connaissance à partir de différentes ressources.

3.2.1 Méthodes d'extraction de connaissances à partir de thésaurus, de bases de données ou d'ontologies déjà existantes

[Hahn et Schulz, 2004] proposent une méthode de construction d'ontologies s'appuyant sur la logique de descriptions ALC à partir du vocabulaire intégré « Unified Medical Language System » (UMLS). La méthode construit une hiérarchie de concepts à partir des termes de la ressource et extrait des relations entre ces concepts. Chaque concept est décrit formellement en ALC afin de pouvoir appliquer des raisonnements sur la structure résultante. Enfin, l'expert est mis dans la boucle de construction pour vérifier et valider l'ontologie résultante. [Stumme et Maedche, 2001] fusionnent deux ontologies déjà existantes en s'appuyant sur un corpus de textes. Ils utilisent des techniques de TALN pour extraire des termes d'un corpus de textes, puis l'Analyse Formelle de Concepts (AFC), une méthode de fouille de données symbolique, pour relier chacune des deux ontologies existantes aux termes extraits. Les instances, concepts et relations des deux ontologies qui ont pu être reliés aux mêmes termes des textes constituent l'ontologie de fusion. Ces deux méthodes utilisent différentes ressources pour construire une ontologie, mais n'extraient pas de nouvelles connaissances pour enrichir l'ontologie résultante.

[Faatz et Steinmetz, 2004] enrichissent une ontologie déjà existante, construite manuellement par les experts du domaine. Leur méthode consiste à extraire des propositions d'un corpus de textes (une proposition peut être un mot ou une phrase), puis à demander aux experts de placer ces propositions en tant que nouvelles instances ou que nouveaux concepts dans l'ontologie.

3.2.2 Détection des termes du domaine à partir de corpus de textes

La première étape de la transformation des données est l'identification des termes ou unités terminologiques à partir du corpus de textes. Ces termes sont les unités significatives constituées d'un mot (terme simple) ou plusieurs mots (terme complexe) qui désignent une notion univoque à l'intérieur d'un domaine [Dubois *et al.*, 1994]. Dans les dix dernières années, plusieurs logiciels de détection de termes à partir de corpus de textes ont vu le jour. Nous citons, par exemple, les logiciels LEXTER [Bourigault, 1994] et FASTR [Jacquemin, 1997], qui reposent sur des critères morpho-syntaxiques pour extraire des groupes nominaux à partir de corpus de textes en langues française et anglaise. Les deux logiciels permettent l'utilisation de patrons syntaxiques et également de thésaurus déjà existants pour extraire ces termes.

3.2.3 Identification de descripteurs binaires de termes à partir de corpus de textes

L'objectif de l'extraction des descripteurs binaires de termes est le regroupement de ces termes en une hiérarchie de concepts. Un premier type de travaux développé dans ce but repose sur l'identification de patrons syntaxiques. [Hearst, 1992] propose de construire une hiérarchie de termes en détectant la relation d'hyponymie entre les concepts à partir des textes. Les patrons suivants sont extraits : « `X est un Z` » ou « `X, Y, et autres Z` » (par exemple, « `les chiens, les chats et autres animaux`... »). Puis, elle définit « `Z` » comme étant un hyperonyme de « `X et Y` ». Cette méthode est efficace pour des corpus de textes comme les dictionnaires ou les supports pédagogiques, mais pas pour des corpus de textes scientifiques comme les textes d'astronomie ou de microbiologie, car il est difficile de trouver une définition pour chaque classe d'objets.

Un second type de travaux repose sur l'hypothèse de distribution de Harris : « *des termes sont similaires s'ils partagent une similarité linguistique* » [Harris, 1968]. Pour la langue anglaise, le travail de [Grefenstette, 1994] regroupe dans un même concept à l'aide du système SEXTANT (qui effectue l'analyse syntaxique de chaque phrase d'un corpus) tous les noms apparaissant derrière un même ensemble d'expressions (verbe + préposition). [Faure et Nedellec, 1999] généralisent cette idée dans le système d'apprentissage appelé « Acquisition of Semantic knowledge Using Machine learning methods » (ASIUM). ASIUM regroupe dans un même concept tous les termes apparaissant comme arguments (sujets, objets, compléments, préposition, ...) du même ensemble de verbes. [Cimiano *et al.*, 2005] utilise le même principe pour extraire des paires `(terme,attributs_binaire)`. Pour le français, les travaux de [Habert et Nazarenko, 1996] regroupent, en utilisant le système ZELLIG, des termes apparaissant dans des patrons du type « `N prep N` » (nom apparaissant après ou avant un nom et une préposition) ou « `N Adj` » (nom ayant les mêmes adjectifs) dans un même concept. Le travail de [Bourigault, 1994] avec le système LEXICLASS décompose les termes complexes T_1 T_2 (par exemple, « `Star formation` ») en T_1 : terme de tête et T_2 : terme d'expansion (par exemple, `Star` terme de tête - `formation` terme d'expansion) puis regroupe les termes T_1 ayant les mêmes termes T_2 dans un même concept et les termes T_2 ayant les mêmes termes T_1 dans un autre concept. Les méthodes présentées ici ne prennent pas en compte les relations qu'entretiennent les termes entre eux. Dans ce qui suit, nous présentons des méthodes qui extraient de telles relations.

3.2.4 Identification de relations transversales entre termes à partir de corpus de textes

L'identification de relations transversales permet d'avoir une définition plus complète et plus fine des concepts, car les concepts ne sont plus définis seulement par des attributs binaires mais aussi par des attributs relationnels (relations qui les relient à d'autres concepts).

Certains travaux s'intéressent à l'identification de relations (des patrons) dans de très grands corpus de textes hétérogènes sur des thèmes généraux (par exemple des journaux). Les linguistes doivent alors définir des formules, puis des schémas à l'aide d'observations du langage. Parmi ces travaux, le « Système Expert d'Exploration Contextuelle » (SEEK) [Jouis, 1993] vise à extraire des relations de causalité entre des termes dans un corpus de textes en langue française. SEEK utilise des listes de schémas et des règles morphologiques du type : Si <Conditions> Alors <Actions> OU <Conclusions>. La méthode COATIS de [Garcia, 1998] reprend l'idée de SEEK en ajoutant une liste de verbes exprimant la relation de causalité comme « créer, faciliter ou encore pousser à ». Ces travaux ont été adaptés à la langue anglaise par [Goujon, 1999]. Ces

travaux sont très intéressants, mais n'ont été testés que sur la relation de causalité.

D'autres travaux se sont intéressés à l'identification « semi-automatique » de schéma de relation à partir de corpus de textes. Le logiciel STARTEX développé par [Rousselot *et al.*, 1996] extrait à partir d'un verbe v_i (exprimant une relation donnée et choisie par des linguistes) une liste de tous les termes T_1 T_2 liés par le verbe v_i, *i.e.* T_1 v_i T_2 (par exemple, pour le verbe `to contain`, il extrait la liste de termes « `(Galaxy,Stars)` » ou « `(Associations_of_Stars,Stars)` »..., puis, il regroupe dans un concept l'ensemble des termes T_1 d'une même liste et dans un autre concept l'ensemble des termes T_2. Ensuite, il regroupe dans une même relation, les verbes ayant les mêmes listes de termes (T_1,T_2). Le but de cette méthode est d'extraire à la fois une hiérarchie de termes et une hiérarchie de verbes (hiérarchie de relations). Le travail de [Barriere, 2002] utilise le même principe pour proposer une étude des différents patrons syntaxiques, en langue anglaise, exprimant une causalité entre deux termes dans un corpus de textes. Par exemple, les patrons du type « T_1 `resulting in` T_2 » ou « T_1 `can result in` T_2 », ... expriment une relation de causalité entre les deux termes T_1 et T_2. Ces méthodes reposent beaucoup sur la personne qui fixe et valide la liste des verbes.

Le système PROMETHEE développé par [Morin et Martienne, 2000] procède à une première identification de contextes de cooccurrences de termes (supervisée) qu'il analyse afin d'y retrouver des patrons lexico-syntaxiques similaires. Les expressions lexico-syntaxiques dites similaires sont regroupées, ce qui permet de proposer des candidats patrons qui, une fois validés, permettent d'extraire de nouveaux termes. Le travail d'[Aussenac-Gilles *et al.*, 2000] reprend l'idée de PROMETHEE et propose d'extraire des instances de relations manuellement, puis de les généraliser en regroupant les termes qui partagent les mêmes relations avec les autres termes sans prendre en compte les attributs binaires des termes.

Le travail de [Maedche et Staab, 2000] extrait les relations entre termes en utilisant les règles d'associations [Agrawal et Srikant, 1995] pour passer de relations entre termes à des relations entre concepts. Cette méthode nécessite une hiérarchie de concepts préalablement construite et n'étiquette pas les relations entre les concepts. Le nommage des relations est fait manuellement par les experts du domaine.

La méthodologie proposée dans cette thèse permet de regrouper simultanément des termes selon des attributs binaires mais aussi d'après des attributs relationnels (des relations entre termes) et ainsi d'obtenir des hiérarchies de concepts reliés par des relations transversales.

3.2.5 Les méthodes de fouille

La fouille de données est l'étape de l'ECD qui vise à extraire des régularités (ou des irrégularités) de l'ensemble des données préparées. Il existe de nombreuses méthodes de fouille différentes. Le choix de la méthode est déterminant et se fait essentiellement en fonction de l'objectif visé par l'analyste. Les différents objectifs de la fouille (en anglais *mining tasks*) sont [Han et Kamber, 2001] :

- la recherche d'associations entre des attributs qui prennent des valeurs particulières de façon concomitante,
- la classification et prédiction s'appuyant sur la définition d'un modèle à partir d'un jeu de données d'apprentissage,
- la construction de clusters qui regroupent les données selon des mesures de similarité,
- la détection de cas extrêmes révélant une forme d'irrégularité.

Nous distinguons deux types de méthodes de fouille : symboliques et numériques. Ces méthodes peuvent être utilisées sur pratiquement toutes les ressources d'un domaine. Les méthodes

numériques utilisent des mesures statistiques pour regrouper des objets en concepts et construire ensuite la hiérarchie des concepts. Nous citons la méthode de Crouch [Crouch, 1988] qui est l'une des premières méthodes proposées pour construire des hiérarchies de classes à partir de termes. Elle s'appuie sur des distances entre termes comme le « TF/IDF » [Salton et Buckley, 1988] ou encore le « Cosinus ». Curran [Curran et Moens, 2002] enrichit le travail de Crouch en testant plusieurs autres mesures statistiques, telles que : SETCOSINE, SETDICE, DICE, DICE†, JACCARD, JACCARD†,. . ., et il a montré que JACCARD et DICE† donnaient de meilleures résultats en étant testés sur 70 corpus. Les méthodes numériques présentent quelques désavantages tels que le fait que les hiérarchies résultantes ne possèdent pas d'héritage multiple ou encore le fait que ces méthodes ne sont pas déterministes. Tous les travaux présentés dans le paragraphe précédant regroupent successivement les concepts de termes en une hiérarchie par une approche d'agglomération statistique sauf [Cimiano *et al.*, 2005]. [Cimiano *et al.*, 2005] utilisent la méthode symbolique qui est l'Analyse Formelle de Concepts (AFC) [Ganter et Wille, 1999] (détaillée dans la section 5.4) afin d'obtenir une hiérarchie de concepts avec une méthode déterministe. Si l'AFC est appliquée plusieurs fois sur les mêmes données, la hiérarchie de concepts obtenue est la même. [Cimiano *et al.*, 2005] présentent également des tests qui montrent que l'AFC donne de meilleurs résultats que plusieurs mesures statistiques.

Il existe d'autres méthodes symboliques de fouille telles que la classification conceptuelle qui construit une hiérarchie de concepts de façon incrémentale en regroupant un ensemble d'instances d'après leurs descriptions (attributs) et en donnant une définition formelle à chaque concept [Fisher, 1987; Feigenbaum, 1961; Gennari *et al.*, 1989]. La classification par arbres de décision prédit l'appartenance d'un objet à un concept en fonction de ses caractéristiques (attributs). L'avantage de cette classification est que les résultats sont lisibles, mais elle ne teste qu'un seul attribut à la fois. Elle ne prend pas en compte l'héritage multiple et elle ne est pas incrémentale [Breiman *et al.*, 1984; Quinlan, 1986; 1988; Langley, 1996].

Choix de la méthode de fouille de données Dans cette thèse, notre premier objectif est de regrouper un ensemble d'objets selon les attributs qu'ils partagent. Ces attributs nous permettent de définir chaque classe d'objets. Notre deuxième objectif est de prendre en compte les relations transversales entre les concepts d'objets. Ces relation transversales enrichissent les définitions des classes d'objets. Ainsi, pour atteindre ces objectifs, nous choisissons comme méthode de fouille l'Analyse Formelle de Concepts (AFC) et son extension l'Analyse Relationnelle de Concepts (AFC).

3.3 Analyse formelle de concepts et analyse relationnelle de concept

L'Analyse Formelle de Concepts (AFC) [Ganter et Wille, 1999] est un domaine des mathématiques appliquées qui consiste à restructurer la théorie des treillis [Birkhoff, 1967] afin de faciliter son utilisation dans des applications du monde réel et de permettre l'interprétation de ses notions en dehors du cadre théorique aussi bien par des mathématiciens que par des non-mathématiciens. L'AFC permet d'obtenir, à partir d'un contexte formel, la classification d'un ensemble d'objets d'après les attributs binaires qu'ils partagent. Grâce à son extension, l'Analyse Relationnelle de concepts (voir sous-section 3.3.5) permet aussi de regrouper un ensemble d'objets, non seulement par l'ensemble des attributs binaires qu'ils partagent mais aussi par les relations qu'ils entretiennent avec les autres objets.

Cette section se décompose comme suit : tout d'abord les notions de base de la théorie des treillis sont présentées dans les sous-sections 3.3.1 et 3.3.2. Puis, les définitions relatives à l'AFC sont données dans la sous-section 3.3.3. Ensuite, une des propriétés de l'AFC permettant

de fusionner des contextes (« l'apposition de contexte ») est définie dans la sous-section 3.3.4. Puis, l'extension de l'AFC au relationnel avec l'Analyse Relationnelle de Concepts (ARC) ainsi que d'autres extensions sont présentées dans les sous-sections 3.3.5 à 3.3.6. Enfin, la dernière sous-section détaille des méthodes d'extraction de connaissances et de construction d'ontologies s'appuyant sur l'AFC.

3.3.1 Ensemble ordonné

Définition 6 (Relation binaire) *Une **relation binaire** R entre deux ensembles M et N est un ensemble de couples d'éléments (m, n) tels que $m \in M$ et $n \in N$, i.e. un sous ensemble de $M \times N$. $(m, n) \in R$ (aussi noté par $R(m) = n$) signifie que l'élément m est en relation R avec l'élément n. Si $M = N$, on parle de relation binaire sur M. R^{-1} est la relation inverse de R, i.e. la relation entre N et M telle que $nR^{-1}m \Leftrightarrow mRn$.*

Définition 7 (Relation d'ordre partiel) *Une relation ordre partiel R sur un ensemble E est une relation :*

- *réflexive : elle met chaque élément en relation avec lui même : $\forall x \in E, xRx$,*
- *antisymétrique : elle ne met pas deux éléments distincts en relation mutuelle : $\forall (x, y) \in E^2, xRy \wedge yRx \rightarrow x = y$,*
- *transitive : elle permet de transiter par un troisième éléments pour mettre en relation deux éléments différents : $\forall (x, y, z) \in E^3 xRy \wedge yRz \rightarrow xRz$.*

Une relation d'ordre R est souvent notée par $\leq$ (R^{-1} est notée par « $\geq$ ») et on dit que « x *est plus petit que* y » lorsque $x \leq y$.

Définition 8 (Ensemble ordonné) *Un **ensemble partiellement ordonné** (ou simplement ensemble ordonné) est un couple $(E, \leq)$ où E est un ensemble et « $\leq$ » est une relation d'ordre sur E.*

Tout ensemble ordonné, $(E, \leq)$, peut être représenté graphiquement par un diagramme appelé « diagramme de Hasse » (ou diagramme de couverture) et obtenu comme suit :

1. Tout élément de E est représenté par un petit cercle dans le plan
2. Si $x, y \in E$ et $x < y$ alors le cercle correspondant à y doit être au-dessus de celui correspondant à x et les deux cercles sont reliés par un segment.

A partir d'un tel diagramme on peut lire la relation d'ordre comme suit : $x < y$ si et seulement s'il existe un chemin ascendant qui relie le cercle correspondant à x à celui correspondant à y.

3.3.2 Treillis

Définition 9 (Treillis) *Un ensemble ordonné $(E, \leq)$ est un* treillis, *si et seulement si tout couple d'éléments (x, y) de E possède une borne supérieure unique notée : $x \vee y$, et une unique borne inférieure notée : $x \wedge y$. $x \leq x \vee y$ et $y \leq x \vee y$ et il n'existe aucun élément z tel que : $x \leq z \leq x \vee y$ et $y \leq z \leq x \vee y$.*

Un treillis $(E,\leq)$ possède un élément qui subsume tous les autres éléments noté $\top$ et un élément qui est subsumé par tous les autres éléments noté $\bot$.

Fermeture. Une fermeture dans un ensemble ordonné $(E, \leq)$ est une application, $H : E \rightarrow E$, telle que pour tout $(x, y) \in E$, les propriétés suivantes sont vérifiées :

- H est extensive : $x \leq H(x)$.
- H est monotone croissante : $x \leq y \Rightarrow H(x) \leq H(y)$
- H est idempotente : $H(x) = H(H(x))$

Définition 10 (Fermé) *Un sous ensemble X de E est dit fermé pour l'opérateur de fermeture H si et seulement si : $X = H(X)$.*

Connexion de Galois. Soit G et M deux ensembles et $\mathcal{P}(G)$ (respectivement $\mathcal{P}(M)$) l'ensemble des parties de G (respectivement M). Les concepts dans un treillis de concepts sont calculés selon la *connexion de Galois* définie par les deux opérations de dérivation suivantes :

- $\varphi : \mathcal{P}(G) \rightarrow \mathcal{P}(M)$ tel que φ(X)= $\{m \in M | \forall g \in X, gIm\}$,
  φ associe à chaque ensemble d'objets de G, l'ensemble des attributs qu'ils ont en commun dans M.
- $\psi : \mathcal{P}(M) \rightarrow \mathcal{P}(G)$ tel que ψ(Y)= $\{g \in G | \forall m \in Y, gIm\}$.
  ψ associe à chaque ensemble des attributs de M, l'ensemble des objets qui les possèdent dans G.

Les applications φ et ψ sont décroissantes et elles possèdent les propriétés suivantes :

- $\forall$ (G_1,G_2) $\in \mathcal{P}(G)$, $G_1 \subseteq G_2 \Rightarrow \varphi\ (G_2) \subseteq \varphi\ (G_1)$;
- $\forall$ (M_1,M_2) $\in \mathcal{P}(M)$, $M_1 \subseteq M_2 \Rightarrow \psi\ (M_2) \subseteq \psi\ (M_1)$;
- $\forall\ M_1 \in \mathcal{P}(M)$, $M_1 \subseteq \varphi\ (\psi\ (M_1))$ et $\forall\ G_1 \in \mathcal{P}(G)$, $G_1 \subseteq \psi\ (\varphi\ (G_1))$.

Les applications $l = \varphi \circ \psi$ et $l' = \psi \circ \varphi$ sont respectivement des fermetures sur ($\mathcal{P}(G)$,$\subseteq$) et ($\mathcal{P}(M)$,$\subseteq$).

Soient F_G et F_M les ensembles des fermés respectifs pour l et l'. (F_G,$\subseteq$) est le treillis des fermés pour l et (F_M,$\subseteq$) est le treillis des fermés pour l'. (F_G,$\subseteq$) et (F_M,$\subseteq$) sont deux treillis isomorphes.

Définition 11 (La connexion de Galois) *Le couple (φ, ψ) forme une connexion de Galois entre ($\mathcal{P}(G)$,$\subseteq$) et ($\mathcal{P}(M)$,$\subseteq$).*

3.3.3 Analyse formelle de concepts

L'Analyse Formelle de Concepts (AFC) est une méthode mathématique qui permet d'obtenir des concepts structurés hiérarchiquement regroupant des objets partageant les mêmes attributs. La hiérarchie résultant de l'AFC est appelée treillis de Galois [Barbut et Monjardet, 1970] ou treillis de concepts [Ganter et Wille, 1999]. Dans ce qui suit, nous utilisons l'appellation de treillis de concepts. Nous présentons les notions fondamentales de l'AFC extraites de [Ganter et Wille, 1999].

Définition 12 (Contexte formel)

Un contexte formel est un triplet $\mathbb{K} = (G, M, I)$ où G est un ensemble d'objets, M est un ensemble d'attributs et I une relation binaire entre G et M vérifiant :

- *$I \subseteq G \times P$.*
- *$(g, m) \in I$ avec $g \in G$ et $m \in M$ signifie que l'objet g possède l'attribut m ou que l'attribut m est possédée par l'objet g.*

Définition 13 *Connexion de Galois dans un contexte formel Soit $\mathbb{K} = (G, M, I)$ un contexte formel. Pour tout $A \subseteq G$ et $B \subseteq M$ on définit :*

$A' := \{m \in M \mid \forall g \in A \mid gIm \}$,
$B' := \{g \in G \mid \forall m \in B \mid gIm\}$,

TAB. 3.1 – Contexte des animaux $\mathbb{K}_1 = (G, M_1, I_1)$

	aDespoils	aDesplumes	estOvipare	pratiqueAllaitement	peutVoler	estAquatique	aDesDents	aDesNageoires	aUneQueue
Antilope	×			×			×		×
Koala	×			×			×		
Sanglier	×			×			×		×
Carpe			×			×	×	×	×
Buffle	×			×			×		×
Poulet		×	×		×				×
Poisson-chat			×			×	×	×	×
Guépard	×			×			×		×

Définition 14 (Concept formel) *Un concept formel d'un contexte $\mathbb{K} = (G, M, I)$ est une paire (A, B) avec : $A \subseteq G$, $B \subseteq M$, $A' = B$ et $B' = A$, où A' est l'ensemble de tous les attributs de B possédés par les objets de A et de façon duale B' est l'ensemble de tous les objets possédant les attributs de B. Plus précisément, pour un concept (A, B), A et B sont des ensembles de fermés pour les opérateurs de dérivation. Les ensembles A et B sont appelés respectivement* ***extension*** *et* ***intension*** *du concept formel C. $\mathfrak{B}(G, M, I)$ dénote l'ensemble de tous les concepts du contexte $\mathbb{K} = (G, M, I)$.*

Le tableau 3.1 présente un exemple d'un contexte formel $\mathbb{K} = (G, M, I)$ où G est l'ensemble d'objets constitué par des animaux, M est l'ensemble des attributs et I est une relation $G \times M$ telle que : $(g, m) \in I$ si et seulement si l'animal g possède l'attribut m.

Définition 15 (Relation de subsomption) *Soient $(A1, B1)$ et $(A2, B2)$ deux concepts formels de $\mathfrak{B}(G, M, I)$. $(A1, B1) \sqsubseteq (A2, B2)$ si et seulement si $A1 \subseteq A2$ (ou de façon duale $B2 \subseteq B1$). $(A2, B2)$ est dit* ***super-concept*** *de $(A1, B1)$ et $(A1, B1)$ est dit* ***sous-concept*** *de $(A2, B2)$. La relation « $\sqsubseteq$ » est dite relation de subsomption.*

Soient $C_1 = (G_1, M_1) \sqsubseteq C_2 = (G_2, M_2)$, et $\sqsubseteq$ la relation de subsomption entre concepts formels : $C_1 \sqsubseteq C_2 \Leftrightarrow G_1 \subseteq G_2$ ou $M_2 \subseteq M_1$. On dit que C_1 est subsumé par C_2 et que C_2 est le subsumant de C_1.

Hiérarchie. Le terme « hiérarchie » en analyse des données est défini par [Diday *et al.*, 1982] comme suit : « Soit Ω un ensemble d'individus, construire une hiérarchie sur Ω consiste à construire un arbre (pas d'héritage multiple), où chaque nœud porte le nom de palier ; chaque

palier d'une hiérarchie correspond à un ensemble d'individus de Ω ; ces individus sont d'autant plus proches entre eux (au sens de la mesure de ressemblance choisie) que le niveau du palier correspondant est bas. » Dans le domaine de la représentation des connaissances et dans l'AFC, une hiérarchie n'est plus considérée comme un arbre, mais comme un ordre partiel donné par la relation de subsomption [Ganter et Wille, 1999]. C'est cette deuxième définition que nous utilisons dans cette thèse.

Définition 16 (Treillis de concepts) *Soit $(F,\sqsubseteq)$ le produit des deux treillis $(F_G,\sqsubseteq)$ et $(F_M,\sqsubseteq)$ appelés respectivement treillis des extensions et treillis des intensions. $(F,\sqsubseteq)$ est le* ***treillis de concepts*** *associé à la relation binaire I sur $G \times M$. L'ensemble de tous les concepts formels du contexte $\mathbb{K} = (G, M, I)$ muni de l'ordre partiel $\sqsubseteq$ est un treillis appelé* ***treillis des concepts*** *de $\mathbb{K}$ et noté $\underline{\mathfrak{B}}(G, M, I)$.*

La figure 3.6 présente le treillis de concepts correspondant au contexte $\mathbb{K} = (G, M, I)$.

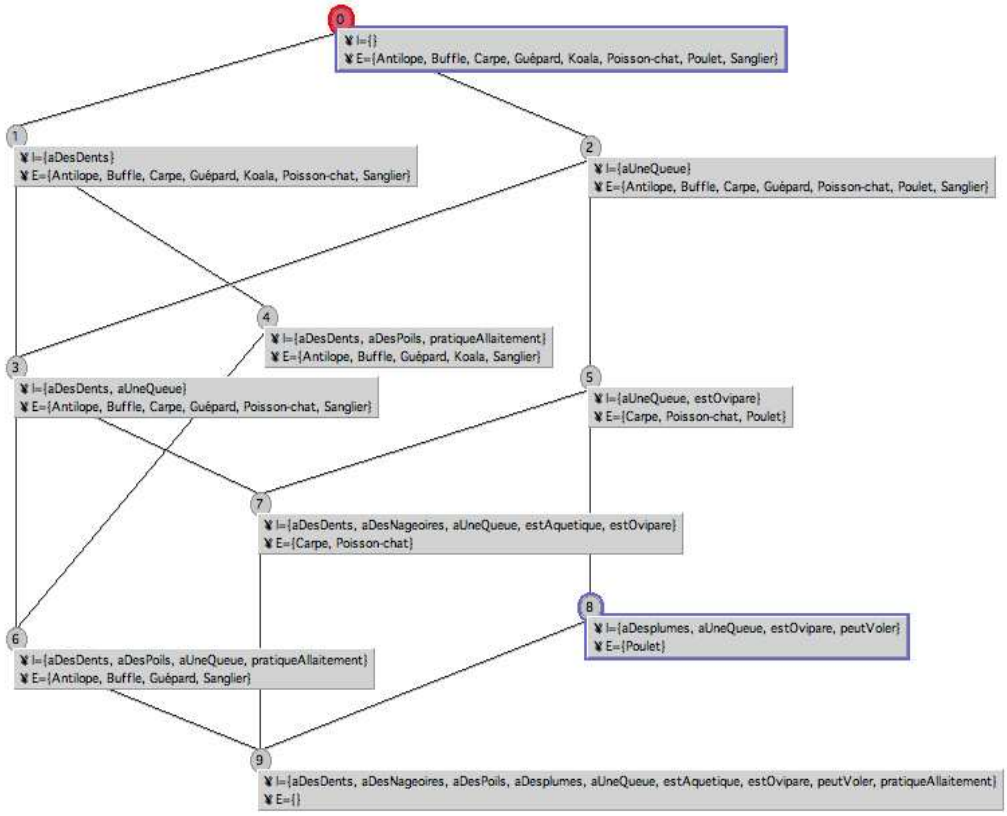


FIG. 3.6 – Treillis des concepts complet du contexte $\mathbb{K} = (G, M, I)$

Méthodes de construction des treillis de concepts. Il existe deux familles différentes d'algorithmes de construction de treillis de concepts : les algorithmes non incrémentaux et les algorithmes incrémentaux.

1. Algorithmes non incrémentaux : ces algorithmes obligent à reconstruire le treillis à chaque fois que l'ensemble des objets ou que l'ensemble des attributs sont modifiés. Les principaux algorithmes de cette famille sont :

- Norris [1978] : recherche des rectangles maximaux en examinant le tableau ligne par ligne ;
- Bordat [1986] : construction simultanée des fermés et du diagramme de Hasse par recherche des fermés couverts par un élément du treillis ;
- Chein [1969] : recherche des rectangles maximaux par opération d'union sur les extensions et d'intersection sur les intensions ;
- Ganter [1984] : construction des fermés pour une relation de fermeture avec utilisation de l'ordre lexicographique.

Des comparaisons de ces algorithmes ont été données dans [Guénoche, 1990] et plus récemment dans [Kuznetsov et Obiedkov, 2002]. L'algorithme de Ganter est le plus efficace pour des données de plus de 15 objets. La méthode de Bordat est la seule à construire directement le diagramme de Hasse du treillis. La méthode de Norris est la plus rapide et celle de Chein est la plus facile à simuler manuellement.

2. Algorithmes incrementaux : Si l'ensemble des attributs ou des objets augmente, les algorithmes incrémentaux permettent de modifier localement les nœuds qui doivent être modifiés et d'insérer éventuellement de nouveaux nœuds [Godin *et al.*, 1991].

Dans tous les treillis de concepts que nous présentons dans cette thèse, nous utilisons la notation des concepts dite réduite : ce qui signifie qu'elle s'appuie sur l'héritage à la fois des attributs et des objets entre les concepts du treillis. Les attributs sont placés au plus haut dans le treillis, ce qui veut dire qu'à chaque fois qu'un concept `C` est étiqueté par un attribut m, alors m est appelé « attribut propre » du concept `C` et tous les descendants de `C` dans le treillis héritent l'attribut m. De façon duale, les objets sont placés au plus bas dans le treillis, ce qui veut dire qu'à chaque fois qu'un concept `C` est étiqueté par un objet g, g est appelé « objet propre » du concept `C` et g est hérité « vers le haut » et tous les ancêtres de `C` le partagent. Ainsi l'extension A d'un concept (A, B) est obtenue en considérant toutes les extensions des descendants du concept `C` dans le treillis et son intension B est obtenue en considérant toutes les intensions des ascendants du concept `C` dans le treillis [Ganter et Wille, 1999]. La figure 3.7 présente le treillis de concept utilisant la notion réduite des concepts du contexte $\mathbb{K} = (G, M, I)$.

La construction de treillis peut présenter différents avantages dans un processus d'ECD [Stumme *et al.*, 1998; Wille, 2002] :

- Le regroupement logique des concepts dans un treillis reflète la façon avec laquelle les humains conceptualisent un domaine, ce qui facilite la lecture des concepts par un analyste.
- Le treillis de concepts n'est construit qu'à partir des données et sans a priori, ce qui peut aider les experts du domaine à extraire des connaissances de ces données.
- Les treillis sont également utilisés en recherche d'information (RI) [Carpineto et Romano, 2004]. L'utilisation de l'AFC en RI est, entre autre, motivée par l'analogie entre objet/attribut et document/terme. Cette utilisation consiste à définir un concept formel comme étant une classe de documents qui correspondent à une requête de l'utilisateur. L'AFC donne la possibilité de généraliser ou de spécifier la recherche en naviguant dans le treillis de concepts. Plus on remonte dans le treillis, plus on généralise la requête. *A contrario* plus on descend dans le treillis, plus on spécialise la requête.

3.3.4 Apposition de contextes

L'apposition de contextes est une propriété de l'AFC qui permet de gérer plusieurs contextes avec le même ensemble d'objets et des ensembles d'attributs disjoints. Pour illustrer cette propriété nous reprenons le contexte formel présenté dans le tableau 3.1 et nous présentons un autre

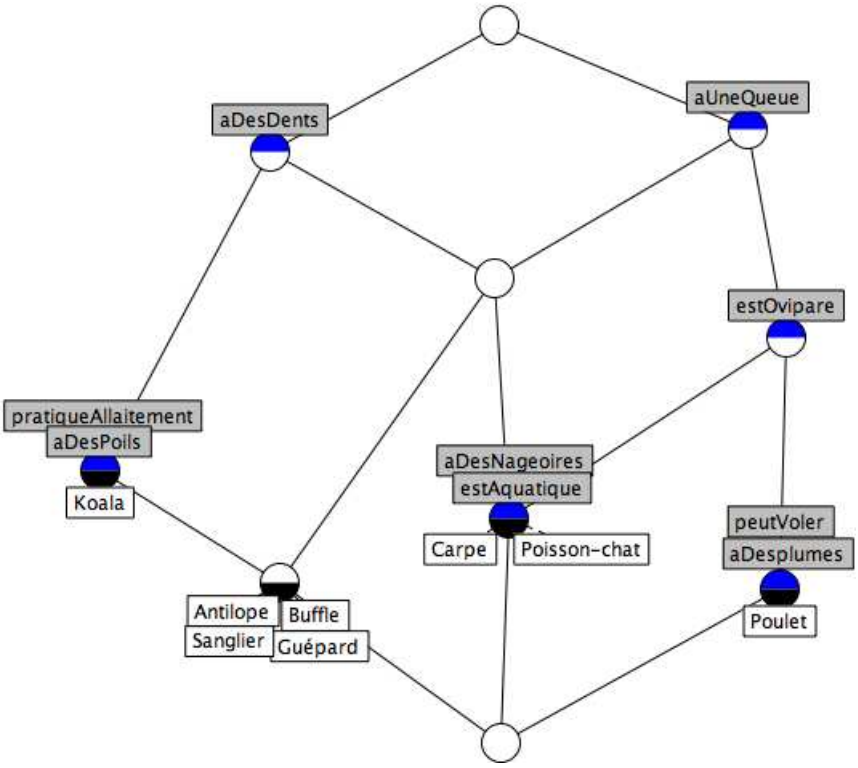


FIG. 3.7 – Treillis de concepts utilisant la notation réduite des concepts

contexte $\mathbb{K}_2 = (G, M_2, I_2)$ donné dans le tableau 3.2 avec son treillis de concepts correspondant donné dans la figure 3.8. Dans ce contexte, le même ensemble d'objets G que dans le contexte $\mathbb{K}_1 = (G, M_1, I_1)$ est associé à un autre ensemble d'attributs M_2 qui constitue les classes des animaux (par exemple l'objet `Antilope` est de la famille des `Bovidés`). Les deux ensembles M_1 et M_2 sont complètement disjoints. Il existe une opération de l'AFC qui nous permet de fusionner ces deux contextes en un seul. Cette opération s'appelle l'apposition de contextes.

TAB. 3.2 – Contexte des animaux $\mathbb{K} = (G, M, I)$

	Bovidés	Mammifères	Marsupiaux	Suidés	Cyprinidae	Gallinacés	Poissons	Ictaluridae	Félins
Antilope	×	×							
Koala		×	×						
Sanglier		×		×					
Carpe					×		×		
Buffle	×	×							
Poulet						×			
Poisson-chat							×	×	
Guépard		×							×

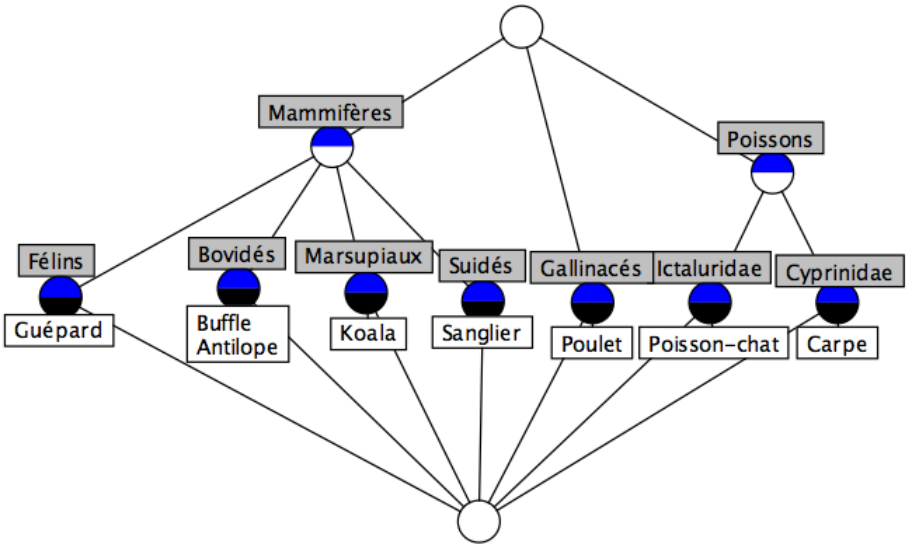


FIG. 3.8 – Le treillis du contexte $\mathbb{K}_2$=(G,M_2,I_2)

La définition de l'apposition est donnée par [Ganter et Wille, 1999] comme suit :

Définition 17 (Apposition de contextes) *Soient* $\mathbb{K}_1 = (G_1, M_1, I_1)$ *et* $\mathbb{K}_2 = (G_2, M_2, I_2)$ *deux contextes formels. Si* $G = G_1 = G_2$ *et* $M_1 \cap M_2 = \emptyset$ *alors* $\mathbb{K} := \mathbb{K}_1 | \mathbb{K}_2 := (G, M_1 \cup M_2, I_1 \cup I_2)$. $\mathbb{K}$ *est* ***l'apposition des deux contextes*** $\mathbb{K}_1$ *et* $\mathbb{K}_2$.

TAB. 3.3 – Contexte d'apposition des animaux $\mathbb{K} = (G, M, I)$

	aDespoils	aDesplumes	estOvipare	pratiqueAllaitement	peutVoler	estAquatique	aDesDents	aDesNageoires	aUneQueue	Bovidés	Mammifères	Marsupiaux	Suidés	Cyprinidae	Gallinacés	Poissons	Ictaluridae	Félins
Antilope	×			×			×		×	×	×							
Koala	×			×			×				×	×						
Sanglier	×			×			×		×		×		×					
Carpe			×			×	×	×	×					×		×		
Buffle	×			×			×		×	×	×							
Poulet		×	×		×				×						×			
Poisson-chat			×			×	×	×	×							×	×	
Guépard	×			×			×		×		×							×

Le contexte d'apposition $\mathbb{K} = (G, M, I)$ est présenté dans le tableau 3.3 avec son treillis de concepts correspondants donné dans la figure 3.9. Dans ce treillis, des objets (ici les animaux) sont

regroupés par le même ensemble d'attributs binaires mais aussi par leur appartenance aux mêmes classes. Par exemple, les objets {`Antilope, Buffle`} sont regroupés dans le même concept, car ils partagent les attributs {`aDesplumes, pratiqueAllaitement, aUneQueue, aDesDents`} mais aussi parce qu'ils sont dans les classes {`Mammifères`, `Bovidés`}.

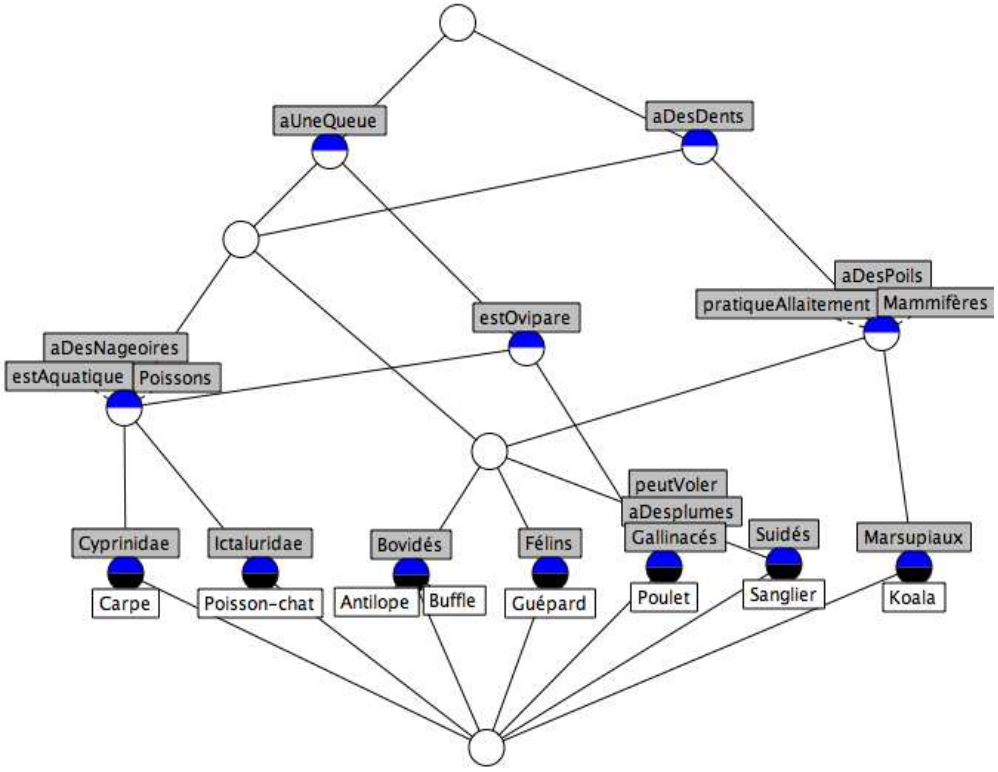


FIG. 3.9 – Le treillis résultant de l'apposition des contextes $\mathbb{K}_1$=(G,M_1,I_1) et $\mathbb{K}_2$=(G,M_2,I_2)

3.3.5 Analyse Relationnelle de Concepts

L'Analyse Relationnelle de Concepts (ARC) est une extension de l'AFC permettant l'extraction de concepts formels à partir d'une description hétérogène des objets formels combinant des attributs binaires et des relations inter-objets [Dao *et al.*, 2004; Rouane-Hacene *et al.*, 2007]. Les concepts formés sont dits « concepts relationnels », car les intensions qu'ils referment font référence à d'autres concepts [Rouane, Septembre 2006].

L'AFC ne permet de traiter que des attributs binaires (Vrai/Faux) et ne permet pas de prendre en compte les relations entre objets. L'ARC propose de prendre en compte ces descriptions relationnelles dans la construction des concepts.

Famille de contextes relationnels. Les données en ARC sont organisées dans une Famille de Contextes Relationnels (FCR) composée d'un ensemble de contextes $\mathbf{K} = \{\mathbb{K}_i\}$ et d'un ensemble de relations $\mathbf{R} = \{r_k\}$. Ainsi, une relation $r_k \subseteq G_i \times G_j$ donne lieu à une relation binaire dont les lignes et les colonnes correspondent respectivement aux objets de G_i et G_j. Formellement, une famille de contextes relationnels est définie comme suit :

Définition 18 (Famille de contextes relationnels) *Une famille de contextes relationnels (*FCR*) est un couple* $(\mathbf{K}, \mathbf{R})$ *où :*

- $\mathbf{K}$ *est un ensemble de contextes* $\mathbb{K}_i = (G_i, M_i, I_i)$,
- $\mathbf{R}$ *est un ensemble de relations* $r_k \subseteq G_i \times G_j$ *où* G_i *et* G_j *sont des ensembles d'objets de contextes dans* $\mathbf{K}$.

Les relations $\mathbf{R}$ sont orientées et décrivent des fonctions ensemblistes, c'est-à-dire, $r_k : G_i \rightarrow 2^{G_j}$. Ainsi, pour toute relation $r_k \subseteq G_i \times G_j$, nous considérons les fonctions suivantes :

- L'ensemble G_i est l'ensemble d'objets du contexte $\mathbb{K}_i$ appelé *domaine* de la relation r_k et noté : dom : $\mathbf{R} \rightarrow \mathbf{G}$, $\mathrm{dom}(r_k) = \mathrm{G}_i$,
- L'ensemble G_j est l'ensemble d'objets du contexte $\mathbb{K}_j$ appelé *co-domaine* de la relation r_k et noté : cod : $\mathbf{R} \rightarrow \mathbf{G}$, $\mathrm{cod}(r_k) = \mathrm{G}_j$,

Traitement des relations en ARC**.** La prise en compte des relations dans les travaux présentés dans la sous-section 3.2.4 est faite à posteriori par rapport à la classification des objets par leurs attributs communs (attributs binaires). L'ARC propose une approche qui consiste à injecter les liens inter-objets dès le départ.

Ainsi, la classification des objets se fait simultanément d'après les attributs binaires qu'ils partagent mais aussi d'après les liens inter-objets (attributs relationnels). Autrement dit, la description des concepts formels possède une partie relationnelle inférée à partir du partage des liens entre les objets de son extension [Rouane, Septembre 2006]. Ainsi, il faut déterminer pour un concept c d'un contexte donné et une relation r dont le domaine est ce même contexte, l'ensemble des concepts cibles par la relation r appliquée à l'extension de $c(r(Ext(c)))$.

Intuitivement, une relation peut être considérée comme un attribut multi-valué et par conséquent prise en charge par un processus d'échelonnage relationnel. Nous présentons dans la sous-section suivante la notion d'échelonnage et détaillons son utilisation pour construire des échelles de relations.

Notion d'échelonnage. L'échelonnage est un mécanisme permettant de convertir un contexte multi-valué en un contexte binaire. Ce mécanisme consiste à remplacer les valeurs concrètes des attributs multi-valués par des abstractions. Ainsi, des échelles sont construites à partir des attributs multi-valués pour construire un contexte binaire. Ces abstractions sont généralement définies par un expert du domaine et constituent des groupes de valeurs concrètes.

Exemple d'échelonnage conceptuel

	Taille (cm)
Jean	186
Paul	172
Pierre	185
Mathieu	198

	Taille <180	180≤ Taille <190	Taille ≥190
Jean		×	
Paul	×		
Pierre		×	
Mathieu			×

FIG. 3.10 – Exemple d'échelonnage d'un contexte multi-valué

Nous prenons un exemple présenté dans le tableau 3.10 décrivant la taille d'individus. Les attributs dans cet exemple ne sont pas des valeurs binaires. Pour les rendre binaires, nous utilisons le mécanisme d'échelonnage et nous définissons des intervalles pour la taille {`Taille < 180, 180` $\leq$ `Taille < 190, Taille` $\geq$ `190`}

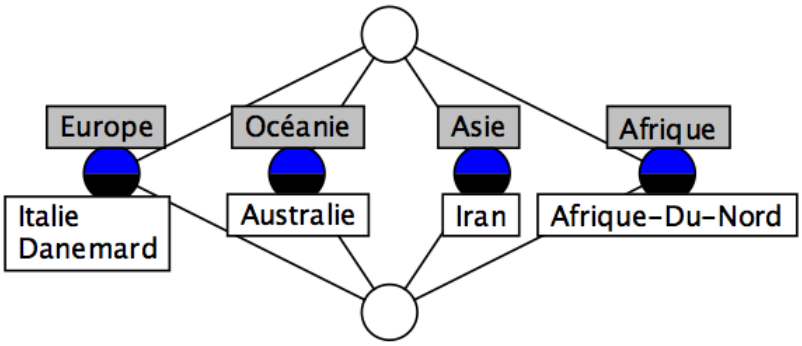


FIG. 3.11 – Le treillis des régions correspondant au contexte $\mathbb{K}_2 = (G_2, M_2, I_2)$

3.3.6 Échelonnage relationnel

L'ARC utilise le mécanisme « d'échelonnage relationnel » (en anglais *relational scaling*) pour définir les attributs relationnels. Pour une relation $r : G_i \longrightarrow G_j$, qui lie les objets de G_i aux objets de G_j, un attribut relationnel est créé et noté $r : c$, où c est un concept du contexte $\mathbb{K}_j$. Ainsi, pour un objet $g \in G_i$, l'attribut relationnel $r : c$ caractérise la « corrélation » entre g et $r(g) = h$ qui est instance du concept $C = (X, Y)$ dans $\mathbb{K}_j$.

Il existe deux niveaux de corrélations : une corrélation « existentielle » ou un « échelonnage existentiel » où $r(g) \cap X \neq \emptyset$ et une corrélation « universelle » ou un « échelonnage universel » où $r(g) \subseteq X$.

Ainsi, à partir de la famille de contextes relationnels (FCR), l'ARC dérive une famille relationnelle de treillis (FRT) ; un treillis pour chaque contexte. Un attribut relationnel est interprété comme étant une relation entre deux concepts. Le concept où apparaît l'attribut relationnel est le domaine de la relation et le concept vers lequel pointe l'attribut relationnel est le co-domaine de la relation.

La famille relationnelle de treillis est extraite par un processus itératif, car l'*échelonnage relationnel* modifie les contextes et par conséquence leurs treillis correspondants, ce qui entraîne un nouvel échelonnage pour tous les contextes contenant une relation qui possède comme co-domaine les treillis qui ont été modifiés. Ce processus itératif s'arrête quand un point fixe est atteint, c'est-à-dire qu'un nouvel échelonnage supplémentaire n'impliquera plus d'extension de contexte [Rouane, Septembre 2006].

Exemple d'échelonnage relationnel

	Afrique	Asie	Europe	Océanie
Afrique-Du-Nord	×			
Iran		×		
Danemark			×	
Italie			×	
Australie				×

TAB. 3.4 – Contexte des régions $\mathbb{K}_2 = (G_2, M_2, I_2)$

	Afrique-Du-Nord	Iran	Danemark	Italie	Australie
Carpe				×	
Koala					×
Sanglier	×	×	×	×	
Guépard	×	×			

TAB. 3.5 – La relation aPourHabitat

	aDespoils	aDesplumes	estOvipare	pratiqueAllaitement	peutVoler	estAquatique	aDesDents	aDesNageoires	aUneQueue	aPourHabitat:C0	aPourHabitat:C2	aPourHabitat:C3	aPourHabitat:C1	aPourHabitat:C4
Carpe			×			×	×	×	×	×		×		
Koala	×			×			×			×	×			
Sanglier	×			×			×		×	×		×	×	×
Guépard	×			×			×		×	×			×	×

TAB. 3.6 – Contexte des animaux $\mathbb{K}_1 = (G, M_1, I_1)$

Nous prenons l'exemple de la famille de contextes relationnels avec $\mathbf{K} = \{\mathbb{K}_1, \mathbb{K}_2\}$, où $\mathbb{K}_1$ est le contexte des animaux présenté dans le tableau 3.3 et $\mathbb{K}_2$ est un contexte avec des régions du monde présenté dans le tableau 3.4, et $\mathbf{R} = \{$aPourHabitat$\}$ qui a comme domaine « Animaux » et comme co-domaine « Régions ». L'application du processus d'ARC produit les deux treillis finaux de la figure 3.12. Le tableau 3.6 présente le contexte des animaux et explique le phénomène d'échelonnage utilisé pour construire le treillis de concepts à gauche de la figure 3.12. Par exemple, l'objet Sanglier est relié aux régions Afrique-du-nord, Iran, Danemark et Italie par la relation aPourHabitat, cela signifie que l'objet Sanglier possédera un attribut relationnel avec tous les concepts où les régions Afrique-du-nord, Iran, Danemark et Italie apparaissent en tant qu'instance. Ainsi, Sanglier possède les attributs relationnels aPourHabitat:C0, aPourHabitat:C1, aPourHabitat:C3 et aPourHabitat:C4.

L'outil GALICIA**.** GALICIA [Valtchev et Missaoui, 2001; Valtchev *et al.*, 2003] est un logiciel libre qui permet de créer, visualiser et sauvegarder des treillis de concepts. C'est une plateforme qui contient une multitude d'algorithmes pour construire des treillis et aussi pour extraire des règles d'association d'un contexte donné. GALICIA est aussi le seul logiciel qui intègre un algorithme pour construire des treillis relationnels à partir d'une famille de contextes relationnels.

3.3.7 Autres extensions de l'analyse formelle de concepts

Dans cette sous-section, nous présentons d'autres approches proposées pour étendre l'AFC et prendre en compte plusieurs types de données et pas seulement des données mono-valuées (objets/attributs binaire). En effet l'utilisation de l'échelonnage conceptuel permet d'élargir les

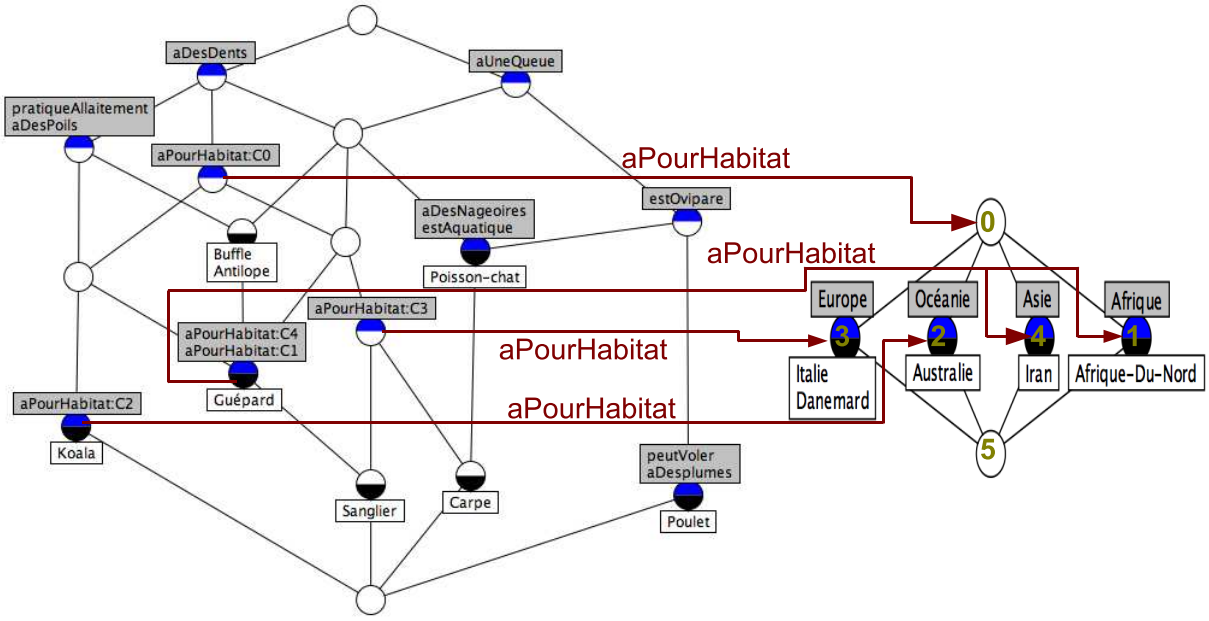


FIG. 3.12 – Les deux treillis résultant de l'application de l'ARC sur la famille de contextes $(\mathbf{K}, \mathbf{R})$

domaines d'application de l'AFC à des données qui peuvent se présenter sous la forme de contextes multivalués. Mais le passage d'un contexte multivalué à un contexte mono-valué ne suit aucune règle et il dépend entièrement des experts du domaine étudié. Pour remédier à ces désavantages, plusieurs travaux de recherche ont été proposés dans le but d'étendre les définitions de l'AFC pour couvrir les données complexes. Nous citons quelques uns de ces travaux. L'Analyse Formelle de Concepts Floue (AFCF) consiste à étendre les résultats de l'AFC aux contextes flous. Un contexte flou est un contexte multivalué où les valeurs de la relation binaire objets/attributs décrivent un degré de vérité pour lequel l'objet est en relation avec l'attribut [Belohlávek et Sklenar, 2005]. L'Analyse de Concepts Logiques (ACL) consiste à étendre les résultats de l'AFC aux contextes logiques. Un contexte logique est un contexte multivalué dans lequel les attributs sont des descriptions qui prennent comme valeurs des formules logiques décrivant les objets du contexte [Ferré, 2002]. Il existe d'autres extensions qui ont été proposées. Ganter et Kuznetsov ont proposé une extension de l'AFC pour l'identification de patrons dans des graphes [Ganter et Kuznetsov, 2001]. Brito et al. [2005] ont défini une approche pour l'analyse des données probabilistes similaire à l'approche AFCF. Elloumi et Jaoua ont défini une approche pour la prise en compte des données imprécises dans des contextes multivalués [Jaoua et Elloumi, 2002]. Enfin, l'extension de l'AFC aux objets symboliques a été proposée par G. Polaillon dans le cadre de ses travaux de thèse [Polaillon, 1998].

3.4 Classification des méthodologies de construction d'ontologie avec l'AFC

Dans cette section, nous utilisons l'AFC pour classifier les méthodologies présentées dans la section 2.2. Il n'existe pas de méthodologie idéale pour toutes les applications, mais chacune d'entre-elles répond à un certain nombre de caractéristiques que l'utilisateur doit définir avant

de choisir la méthodologie à utiliser. L'AFC nous permet de construire une hiérarchie de ces méthodologies d'après leurs caractéristiques et ainsi : un utilisateur doit tout d'abord définir les caractéristiques que doit avoir la méthodologie qu'il veut utiliser, puis naviguer dans le treillis de concepts pour trouver la bonne méthodologie pour son application.

	Ressources obligatoires	Décomposition	Expert	Fouille de textes	Description Formelle	Ressources recommandées	Evolution	Evaluation	Fouille de données	Enrichissement	Semi-automatique
On-to-Knowledge			×	×	×	×	×	×	×		×
Methontology			×	×	×	×			×		×
Sensus	×		×							×	
Uschold et King			×								
Grünger et Fox			×		×			×			
Noy et McGuinness			×			×					
Cyc	×	×	×								
Kactus	×	×	×								

Tab. 3.7 – Le contexte $\mathbb{K}_M = (G_M, M_M, I_M)$ des méthodologies de construction d'ontologies et leurs caractéristiques

Le treillis de concepts des méthodologies est construit à partir du contexte formel $\mathbb{K}_M = (G_M, M_M, I_M)$, tel que G_M est l'ensemble des méthodologies, M_M est l'ensemble des attributs (caractéristiques) des méthodologies (présentées dans la sous-section 2.2.2) et I_M est une relation binaire tel que $I_M(g, m)$ veut dire que la méthodologie g possède la caractéristique m. La caractéristique « méthodologie utilisant des ressources du domaine » est divisée en deux caractéristiques distincts : « méthodologie recommandant l'utilisation de ressources » et « méthodologie obligeant l'utilisation des ressources ». Cette division est importante dans le choix de la méthodologie, car les méthodologies ayant la caractéristique « méthodologie obligeant l'utilisation des ressources » décrivent les méthodes de réutilisation de ces ressources. Le tableau 3.7 présente le contexte binaire $\mathbb{K}_M = (G_M, M_M, I_M)$ et le treillis résultant de l'application de l'AFC sur ce contexte est présenté dans la figure 3.13.

Nous pouvons remarquer que toutes les méthodologies sont contrôlées par les experts du domaine. Néanmoins, nous pouvons distinguer des différences entre les méthodologies. Nous proposons une classification des méthodologies (voir figure 3.13) en trois grandes familles :

- Acquisition de connaissances : regroupe des méthodologies qui extraient manuellement de la connaissance à l'aide des experts du domaine. Nous citons dans cette famille de méthodologies, la méthodologie de Uschold et King, la méthodologie de Noy et McGuinness et celle de Grünger et Fox. Ces méthodologies constituent la base des autres méthodologies.
- Réutilisation et enrichissement de ressources : regroupe les méthodologies qui extraient des connaissances à partir de ressources du domaine déjà existantes. Nous citons dans cette famille de méthodologies, les méthodologies Kactus, Cyc et Sensus.
- Apprentissage à partir des données : regroupe les méthodologies ayant un processus semi-automatique et qui utilisent des méthodes de fouille pour extraire des connaissances à

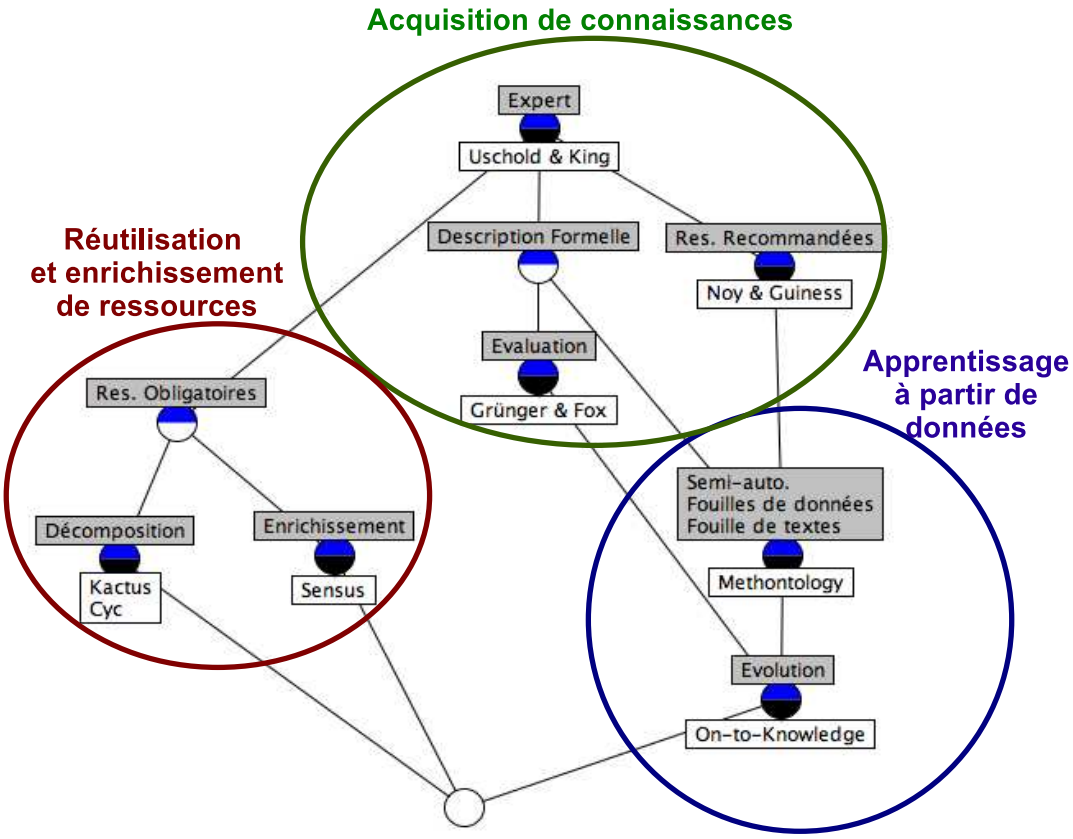


FIG. 3.13 – Treillis résultant du contexte des méthodologie $\mathbb{K}_M = (G_M, M_M, I_M)$

partir de données brutes. Cette famille regroupe les méthodologies METHONTOLOGY et ON-TO-KNOWLEDGE.

Ainsi, si un utilisateur veut enrichir une ontologie ou formaliser puis enrichir des bases de données, alors il doit utiliser les méthodologies de « réutilisation et enrichissement de ressources ». Si par contre, il veut utiliser des méthodes d'apprentissage, il doit utiliser les méthodologies d'« Apprentissage à partir des données ». Par contre aucune méthodologie ne permet de réutiliser et d'enrichir des ressources déjà existantes et en même temps d'utiliser des méthodes d'apprentissage à partir de données afin que le processus de construction de l'ontologie soit semi-automatique.

3.5 Conclusion

Dans ce chapitre, nous avons présenté le processus d'extraction de connaissances à partir de données (ECT) et ses applications sur les bases de données et les corpus de textes. Nous avons également détaillé les méthodes d'extraction de connaissances proposées dans la littérature. Nous avons expliqué pourquoi nous avons choisi la méthode de fouille l'Analyse Formelle de Concepts AFC et son extension l'Analyse Relationnelles de Concepts (ARC). Ensuite, nous avons présenté les notions de base de ces deux méthodes de fouille. Enfin, nous avons proposé une classification des différentes méthodologies proposées dans la section 2.2.

Les notions présentées dans ce chapitre nous permettront de présenter notre méthodologie et notre processus de construction d'ontologies à partir de ressources textuelles hétérogènes

nommées PACTOLE « Property And Class Characterization from Text to OntoLogy Enrichment » dans les deux chapitres suivants.

Chapitre 4

Méthodologie Pactole : Prétraitements des ressources

Sommaire

Introduction

Ce chapitre est divisé en deux parties. La première partie présente notre méthodologie Pactole « Property And Class Characterization from Text to OntoLogy Enrichment ». Il s'agit d'une méthodologie de construction d'ontologies à partir de ressources textuelles hétérogènes.

Nous définissons d'abord ses caractéristiques, puis nous la positionnons par rapport à l'état de l'art. Ensuite, le processus PACTOLE issu de cette méthodologie est détaillé. Ce processus est divisée en trois grandes étapes (voir figure 3.1) : (1) prétraitement des ressources, (2) application de méthodes de fouille de données et enfin (3) représentation et implémentation de l'ontologie.

Dans la deuxième partie de ce chapitre, nous détaillons la première grande étape du processus PACTOLE : le prétraitement des ressources pour l'identification des éléments assertionnels. Les deux autres étapes seront détaillées dans le chapitre 5. L'identification des éléments assertionnels se fait en deux étapes : d'abord définir puis extraire l'ensemble des références lexicales pour les concepts, ce qui signifie l'ensemble des instances que nous utilisons pour construire l'ontologie de domaine. Ensuite, définir puis extraire les différents descripteurs d'objets extraits des différentes ressources afin de construire les références lexicales pour les attributs et les relations. Les différents descripteurs d'objets (*DO*) sont des façons de définir un objet. Par exemple, une classe affectée par les experts du domaine à un objet est un descripteur d'objet, ou un attribut binaire, une relation entre objets sont aussi des descripteurs d'objets. Généralement les descripteurs d'objets sont choisis avec l'aide des experts du domaine d'après leurs connaissances, puis extraits des différentes ressources. Chaque ressource peut fournir un ou plusieurs descripteurs d'objets. Par exemple, un thésaurus peut fournir un ensemble de classes organisées hiérarchiquement. Une base de données ou un corpus de textes peuvent fournir des paires du type `(objet,attribut_binaire)` ou encore un ensemble de triplets du type `(objet`$_i$`,attribut_relationnel,objet`$_j$`)`.

4.1 La méthodologie PACTOLE

Dans cette thèse, nous avons mis en place une méthodologie et un processus nommés PACTOLE « Property And Class Characterization from Text to OntoLogy Enrichment » (présentés dans [Bendaoud *et al.*, 2008b]). Les objectifs de la méthodologie PACTOLE sont tout d'abord de représenter formellement les connaissances déjà existantes dans le domaine, c'est-à-dire proposer aux experts des définitions formelles des classes prédéfinies de leur domaine. Cette représentation permet de raisonner et de poser des questions complexes au système, comme l'instanciation des objets (« Quel est le concept de l'objet `X` sachant ses caractéristiques ? »), la comparaison de concepts (« Existe-il un concept contenant les deux objets `X` et `Y` sachant leurs caractéristiques ? ») ou encore la détection du domaine d'une relation (« Quelle est le concept de l'objet `X` et avec quels objets est-il en relation ? »). Le deuxième objectif est « la découverte de connaissances », c'est-à-dire, proposer des objets, des classes, des attributs, des relations... aux experts du domaine qui, peut-être, les considéreront comme de nouvelles « unités de connaissance ». Afin de montrer l'adaptibilité de cette méthode à différents domaines, nous l'avons appliquée sur deux domaines spécifiques très différents, l'astronomie et la microbiologie.

PACTOLE construit semi-automatiquement une ontologie à partir de différentes ressources du domaine spécifique (thésaurus, bases de données, dictionnaires, corpus de textes,...). Dans cette section, nous présentons tout d'abord les caractéristiques de notre méthodologie. Puis, nous positionnons notre méthodologie par rapport à la classification des méthodologies présentées dans la section 3.4. Enfin, nous présentons un schéma global du processus PACTOLE issu de cette méthodologie.

4.1.1 Caractéristiques de la méthodologie PACTOLE

Dans cette partie, nous présentons les différentes caractéristiques que vérifie notre méthodologie PACTOLE. Son but est de construire une ontologie qui définit formellement des classes d'objets

avec des descripteurs d'objets identifiés dans différentes ressources textuelles hétérogènes. Pour cela, notre méthodologie doit satisfaire les neuf conditions (introduites dans la section 2.2) :

1. être contrôlée par un expert : l'expert doit intervenir à chaque étape de notre méthodologie pour des validations,
2. prendre en compte les ressources déjà existantes dans le domaine (bases de données, thésaurus, ...) afin de ne pas repartir à zéro et de réutiliser ce qui existe,
3. utiliser des méthodes de fouille de données et de fouille de textes,
4. fusionner les différents types de ressources,
5. décrire formellement les concepts du domaine et les relations entre ces concepts,
6. avoir un processus semi-automatique : pour pouvoir traiter des bases de données et des corpus de textes volumineux,
7. être évaluée pour savoir si elle répond aux besoins des experts,
8. enrichir une ontologie déjà existante,
9. être capable d'évolution.

4.1.2 Positionnement de la méthodologie PACTOLE

TAB. 4.1 – Le contexte $\mathbb{K}_M = (G_M, M_M, I_M)$ représentant les méthodologies et leurs caractéristiques

	Ressources obligatoires	Décomposition	Expert	Fouille de textes	Description Formelle	Ressources recommandées	Evolution	Evaluation	Fouilles de données	Enrichissement	Semi-automatique
ON-TO-KNOWLEDGE			×	×	×	×	×	×	×		×
METHONTOLOGY			×	×	×	×			×		×
SENSUS	×		×							×	
Uschold et King			×								
Grünger et Fox			×		×			×			
Noy et Guiness			×			×					
CYC	×	×	×								
KACTUS	×	×	×								
PACTOLE	×		×	×	×		×	×	×	×	×

Nous ajoutons la méthodologie PACTOLE au contexte $\mathbb{K}_M = (G_M, M_M, I_M)$ afin de la positionner par rapport aux autres méthodologies. Le nouveau contexte est présenté dans la table 4.1 et le treillis correspondant à ce contexte est présenté dans la figure 4.1. La méthodologie PACTOLE est singulière, car à notre connaissance c'est la seule méthodologie qui fait partie des deux

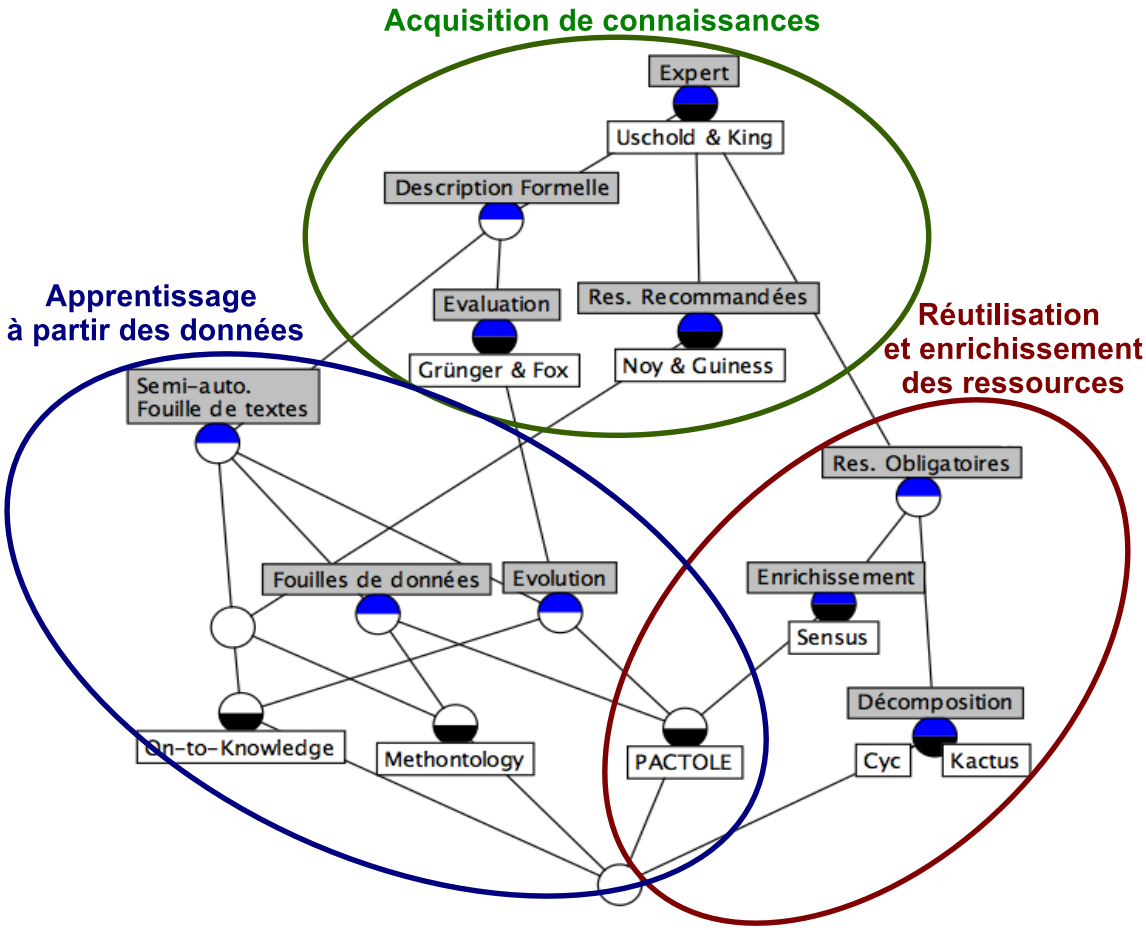


FIG. 4.1 – Treillis de concepts utilisé pour positionner la méthodologie PACTOLE

familles de méthodologies que sont : l'apprentissage à partir des données et la réutilisation et l'enrichissement de ressources. Elle est inspirée des méthodologies METHONTOLOGY [Gómez-pérez *et al.*, 2004], ON-TO-KNOWLEDGE de [Maedche et Staab, 2004] et SENSUS [Valente *et al.*, 1999]. Des méthodologies METHONTOLOGY et ON-TO-KNOWLEDGE, PACTOLE reprend plusieurs idées telles que : l'introduction de l'expert du domaine dans la boucle de construction de l'ontologie afin de valider chaque étape, la détection d'un ensemble de termes à partir d'un corpus de textes et la définition des concepts résultants en une logique de descriptions. A partir de la méthodologie SENSUS, PACTOLE reprend l'idée de s'appuyer sur des ressources du domaine déjà existantes pour construire l'ontologie au lieu de la reconstruire à partir de zéro et également l'idée d'enrichir l'ontologie par d'autres ressources que les ressources utilisées pour la construire.

4.1.3 Le processus PACTOLE

Le processus PACTOLE construit une ontologie de domaine à partir de ressources textuelles hétérogènes en s'appuyant sur les processus d'extraction de connaissances présentés dans le chapitre 3.1 et sur les méthodes de fouille de données présentées dans la section 3.3. Ce processus se décompose en trois étapes. La première étape consiste à identifier les éléments assertionnels (en anglais *assertion component* (ABOX)) d'une ontologie à partir des différentes ressources textuelles

disponibles. Les éléments assertionnels composent le lexique du schéma d'ontologie qui comprend les ensembles des références lexicales pour les concepts, les attributs et les relations, nommés respectivement Ref_C, Ref_A et Ref_R (ces ensembles sont définis dans la sous-section 2.1.4). La seconde étape de notre processus consiste à construire des hiérarchies de concepts et des relations entre ces concepts, ce qui signifie extraire les éléments terminologiques de l'ontologie (en anglais *terminological component* (TBOX)) à partir des éléments assertionnels de l'ontologie. Enfin, dans la troisième étape, le processus PACTOLE représente le schéma d'ontologie en une ontologie en utilisant un formalisme de la représentation des connaissances, la Logique de Descriptions $\mathcal{FLE}$. Cette ontologie est implémentée en un langage du Web sémantique, le langage Web Ontology Language (OWL).

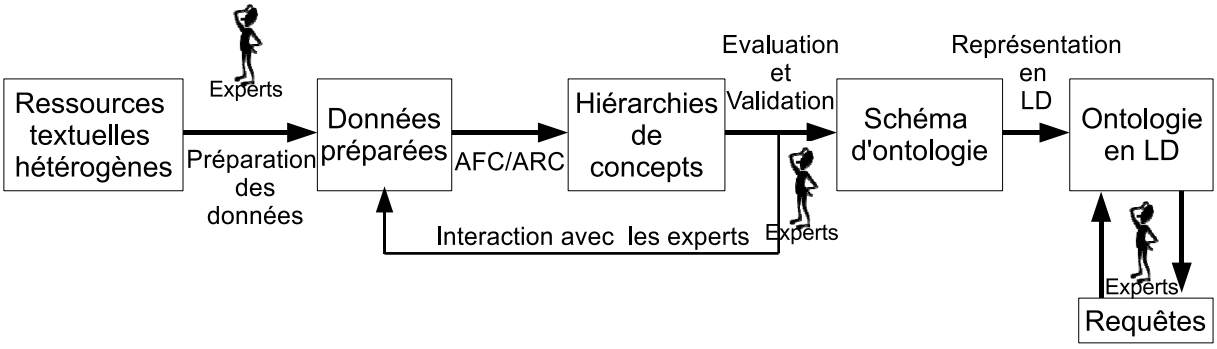


FIG. 4.2 – Schéma global du processus PACTOLE

Le processus PACTOLE (voir figure 4.2) se décompose en six étapes et chaque étape nécessite la validation de l'expert :

1. Préparation des données : cette étape utilise différents types d'outils tels que des outils de Traitement Automatique de la Langue Naturelle (TALN), des outils d'Extraction d'Information (EI) et des outils de transformation pour extraire les objets du domaine (voir la sous-section 4.4) ainsi que les différents types de Descripteurs d'Objets (DO) (voir la section 4.2) à partir des ressources textuelles hétérogènes du domaine,
2. AFC/ARC : cette étape utilise les deux méthodes de fouille que nous avons choisies pour construire des hiérarchies de concepts et des relations entre ces concepts à partir des éléments extraits de l'étape précédente,
3. Evaluation et interprétation des experts : dans cette étape, les experts évaluent les différentes hiérarchies de concepts ainsi que les relations entre ces concepts. Ils ont la possibilité d'interagir avec le processus et de revenir en arrière pour modifier les contextes des méthodes de fouille. Le processus étant itératif, les experts peuvent l'appliquer autant de fois qu'ils le souhaitent, jusqu'à ce que les hiérarchies de concepts résultantes répondent à leurs attentes,
4. Représentation en LD : cette étape permet de passer du schéma d'ontologie à une ontologie représentée avec le formalisme de LD $\mathcal{FLE}$ et implémentée en OWL afin de pouvoir effectuer des raisonnements,
5. Requêtes : cette étape propose les types de requêtes auxquelles notre système peut répondre en utilisant les techniques de raisonnement offertes par les langages de LD.

4.2 Descripteurs d'objets

Le choix des ressources d'un domaine spécifique ne se fait pas par rapport au type de la ressource, mais par rapport aux éléments qu'elle contient. Ces éléments sont nommés Descripteurs d'Objets (*DO*). Pour choisir ces descripteurs d'objets, nous revenons à la définition d'une ontologie donnée dans la sous-section 2.1.4 : une ontologie est composée de hiérarchies de concepts reliés entre eux par des relations transversales. Ainsi, les descripteurs d'objets doivent permettre de construire des hiérarchies de concepts (intension/extension) et des relations entre ces concepts. Ces descripteurs d'objets sont choisis en collaboration avec des experts du domaine et d'après la méthode de fouille que nous avons choisie.

Dans nos deux domaines d'application, les experts ont défini les hiérarchies de classes comme étant le premier descripteur d'objets. Le deuxième descripteur d'objets est constitué d'attributs binaires. Ces attributs sont les caractéristiques propres des objets. Le troisième descripteur d'objets est composé des liens inter-objets, c'est-à-dire des instances de relations qu'entretiennent les objets entre eux. Avant de présenter plus en détails ces différents descripteurs et leurs méthodes d'identification, nous définissons l'ensemble des objets relatifs aux domaines considérés ; ces ensembles sont supposés finis. Pour les deux domaines d'application, les ensembles d'objets que nous considérons sont :

- dans le domaine de l'astronomie, nous définissons l'ensemble des objets célestes comme étant l'ensemble des objets noté G,
- dans le domaine de la microbiologie, nous définissons trois ensembles disjoints d'objets du domaine : les antibiotiques notés G_A, les bactéries notées G_B et les gènes notés G_G.

4.2.1 Descripteur d'objets 1 : Les classes d'objets

Le premier descripteur d'objets (*DO1*) représente les attributs qui à un objet assignent sa classe dans un ensemble prédéfini de classes M_1. Une classe C est définie comme un ensemble d'objets de G ou encore un élément de 2^G (l'ensemble des parties de G). L'ensemble des objets d'une classe C_1 est appelé « extension de la classe » et noté $ext(C_1)$.

Soit M_1 un ensemble de classes d'objets $g_i \in G$ et $\sqsubseteq$ un ordre partiel défini sur M_1 de la façon suivante : $\forall\ C_1, C_2 \in M_1$, $C_1 \sqsubseteq C_2$ si et seulement si $ext(C_1) \subseteq ext(C_2)$ (relation d'inclusion). L'ensemble ordonné (M_1,$\sqsubseteq$) est appelé « hiérarchie source ». Les objets représentent les « feuilles » de cette hiérarchie (nœuds terminaux). Tous les objets d'une classe sont aussi dans les super-classes de celle-ci par la transitivité de la relation d'inclusion.

Hiérarchie source dans le domaine de l'astronomie. Dans le domaine de l'astronomie, c'est la hiérarchie de la base SimBad qui joue le rôle de hiérarchie source du domaine (cette hiérarchie a été choisie par les astronomes). Elle regroupe toutes les classes prédéfinies dans le domaine de l'astronomie. Les objets d'une classe appartiennent à toutes ses super-classes. Par exemple, `3C_273` est une `Quasar`, `Galaxy` est une super-classe de `Quasar` donc `3C_273` est aussi une `Galaxy`.

Hiérarchie source dans le domaine de microbiologie. Faute de hiérarchies manuelles des gènes et des antibiotiques nous n'avons pris en considération que la hiérarchie source des bactéries. C'est la base ncbi Taxonomy qui joue le rôle de hiérarchie source. ncbi Taxonomy regroupe les classes prédéfinies du domaine et assigne chaque objet à une classe et à toutes ses super-classes dans la hiérarchie. Par exemple, la bactérie `Helicobacter_Pylori` est une

`Proteobacteria`, `GammaProteobacteria` est une super-classe de la classe `Proteobacteria` et donc `Helicobacter_Pylori` est une `GammaProteobacteria`.

En revanche, les hiérarchies sources ne donnent pas de définitions aux classes d'objets, elles ne font qu'affecter une classe à un objet. Ainsi, pour définir ces classes, nous extrayons un deuxième type d'éléments : les attributs binaires notés (*DO2*).

4.2.2 Descripteur d'objets 2 : Les attributs binaires

Dans un autre type de ressources, les objets du domaine sont décrits par des attributs binaires qui décrivent des caractéristiques propres aux objets. Ces attributs peuvent être extraits d'une base de données ou d'un corpus de textes. Ces attributs binaires composent l'ensemble Ref_A qui sert à regrouper des instances de l'ensemble Ref_C dans un même concept.

Attributs binaires dans le domaine de l'astronomie. Les attributs binaires dans le domaine de l'astronomie ont été détectés à partir de corpus de textes [Bendaoud *et al.*, 2007a]. Cette méthode d'identification sera détaillée dans la sous-section 4.6.2.

Attributs binaires dans le domaine de la microbiologie. Trois ensembles disjoints d'attributs binaires ont été extraits dans le domaine de la microbiologie, un pour chaque ensemble d'objets. Ces attributs binaires ont été extraits de bases de données. Cette méthode d'identification ainsi que le choix de cette ressource sont présentés dans la sous-section 4.6.3.

Enfin, comme une ontologie est aussi constituée de relations transversales entre concepts, nous avons besoin d'attributs relationnels (*DO3*). Les attributs relationnels sont des instances de relations.

4.2.3 Descripteur d'objets 3 : Les attributs relationnels

Les objets du domaine peuvent être reliés à d'autres objets. Ces relations sont identifiées à partir d'un corpus de textes ou de bases de données. Ces attributs relationnels constituent l'ensemble Ref_R qui représente les instances de relations entre concepts dans une ontologie de domaine.

Attributs relationnels dans le domaine de l'astronomie. Nous avons proposé plusieurs relations dans le domaine de l'astronomie, mais les astronomes ne les ont pas considérées comme des descripteurs de domaine intéressants. Nous avons publié une expérimentation dans [Bendaoud *et al.*, 2007b; 2009] qui prend en compte la relation « isObservedBy » entre les objets célestes et les télescopes. Néanmoins, la classification des objets célestes relativement au type de télescopes permettant de les observer ne s'est pas avérée très intéressante par les astronomes. Nous avons seulement extrait 10 télescopes reliés à 60 objets célestes.

Attributs relationnels dans le domaine de la microbiologie. Dans le domaine de la microbiologie, deux relations ont été prises en compte (choisies également par les microbiologistes) : la relation « isPartOfGenomeOf » entre l'ensemble des gènes et l'ensemble des bactéries et la relation « isResisting » entre l'ensemble des bactéries et l'ensemble des antibiotiques. La méthode d'identification de ces relations est détaillée dans la sous-section 4.7.

Avant de présenter les différentes méthodes d'identification de ces descripteurs d'objets, nous devons appliquer un prétraitement aux corpus de textes car, comme présenté dans la sous-section 3.1.2, le processus d'extraction de connaissances à partir de corpus de textes possède une étape supplémentaire par rapport à ECBD qui est le prétraitement des corpus.

4.3 Prétraitement des corpus de textes

La figure 4.3 présente les différentes étapes du prétraitement que nous appliquons pour identifier les éléments de l'ABOX à partir de corpus de textes. Ce prétraitement se compose de trois étapes : la tokenisation et normalisation, la catégorisation des mots et l'analyse morphologique.

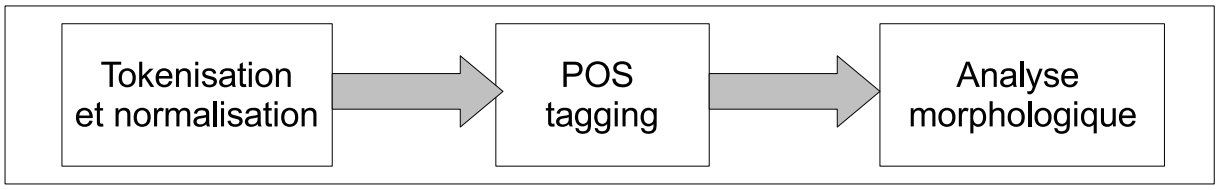


FIG. 4.3 – Prétraitement des corpus de textes

Tokenisation et normalisation. Cette étape consiste à découper les textes en phrases, puis en mots et ensuite à normaliser certains types de mots. Le découpage présente parfois des difficultés comme la différenciation entre un signe de ponctuation qui représente la fin d'une phrase et celui qui est placé après une abréviation telle que : « *i.e* »., ou encore la détection des dates, des horaires, de numéros de téléphone, etc. La normalisation consiste à retrouver des dates ou des horaires par exemple et de les transformer en un format standard. Cette étape de normalisation et de tokenisation est détaillée dans [Klein et Manning, 2002; Feldman et Sanger, 2007].

Catégorisation des mots. Appelée en anglais « *Part-of-speech tagging (POS Tagging)* ». Cette étape permet d'annoter chaque mot dans une phrase avec sa catégorie grammaticale. L'annotation est faite d'après le rôle du mot dans la phrase. Les annotations les plus communes sont : les articles, les verbes, les noms, les adjectifs, les adverbes, les prépositions, ... La plupart des processus de *POS Tagging* s'appuient sur des méthodes statistiques ou probabilistes pour annoter les mots comme dans [Church, 1988; Charniak, 1995]. Dans cette thèse, nous utilisons le « Stanford POS Tagger ». Cet outil a un pourcentage de 96.72 % à 96.86 % de mots correctement étiquetés [Toutanova et Manning, 2000].

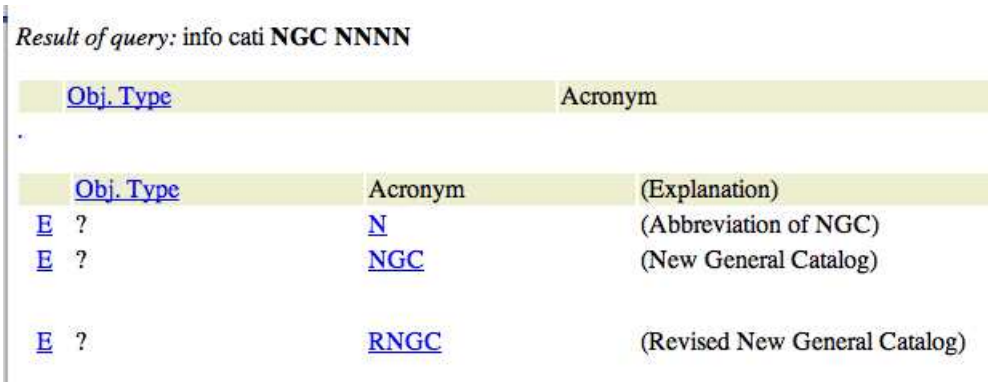
Analyse morphologique. L'analyse morphologique consiste à trouver le lemme d'un mot. Dans cette thèse, nous n'avons pas utilisé d'analyse morphologique de manière très approfondie. Seuls les pluriels des mots ont été gérés.

4.4 Détection des instances

4.4.1 Détection des instances dans le domaine de l'astronomie

Dans le domaine de l'astronomie, les instances sont représentées par les objets célestes. Ces objets ont une forme très particulière et pour les identifier, nous ne pouvons pas utiliser les

méthodes de détection des termes appliquant des critères morpho-syntaxiques ou des critères statistiques comme dans les travaux présentés dans la section 3.1.2. Nous utilisons donc deux stratégies complémentaires suggérées par la base SimBad. La première stratégie consiste à se servir des noms d'objets déjà répertoriés dans la base SimBad (par exemple `Orion`). Ainsi une simple recherche de la chaîne de caractères permet de les localiser dans les textes. La deuxième stratégie consiste à utiliser un dictionnaire de nomenclatures (voir figure 4.4) pour repérer les noms d'objets, par exemple, l'objet `NGC_6994` est détecté par la nomenclature `NGC_NNNN` où `N` est un chiffre.


Result of query: info cati **NGC NNNN**

	Obj. Type	Acronym	(Explanation)
E	?	N	(Abbreviation of NGC)
E	?	NGC	(New General Catalog)
E	?	RNGC	(Revised New General Catalog)

Fig. 4.4 – Dictionnaire de Nomenclatures des objets célestes

Le processus Pactole a identifiés 1382 objets célestes à partir du corpus de textes. Afin d'évaluer cette méthode de détection, les astronomes ont cherché manuellement les objets célestes dans les textes. Les objets que Pactole a identifié représentent 90 % des objets des textes. En revanche, quelques objets détectés ne sont pas des objets célestes. Ces erreurs peuvent être expliquées comme suit :

- Des nomenclatures non discriminantes : quelques objets dans le domaine de l'astronomie possèdent les mêmes nomenclatures que les objets célestes, par exemple, la nomenclature `IRA_X` de la base SimBad détecte l'objet céleste `IRAS_16293` mais aussi le télescope `IRAM_30`,
- Des abréviations dans les textes : quelques auteurs utilisent des abréviations dans les textes, par exemple, ils écrivent `S_180` au lieu de `Sand_180`,
- Des erreurs de typographie dans la base SimBad : quelques noms d'objets sont mal écrits comme, par exemple, `Name_Lupus_2` au lieu de `Lupus_2`.

Découverte d'unités de connaissances. Grâce à notre méthode de détection d'objets qui utilise deux stratégies complémentaires, nous avons réussi à identifier trois nouveaux objets `HH_24MMS, S140_IRS3, M33_X-9` qui n'étaient pas dans la base SimBad. Ces trois objets ont été jugés par les astronomes comme étant des **unités de connaissances** et ils ont été ajoutés à la base SimBad.

Prise en compte des synonymes dans le domaine de l'astronomie. Dans le domaine de l'astronomie, la base SimBad nous donne pour chaque objet céleste les différents identifiants qu'il peut avoir. La figure 4.5 présente tous les identifiants que peut prendre l'objet céleste `NGC_1864`.

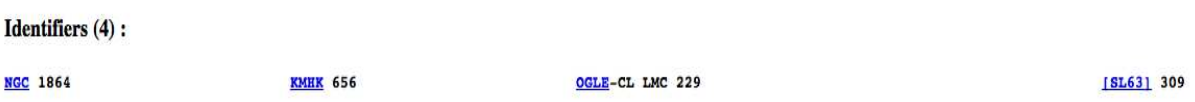


FIG. 4.5 – L'ensemble des identifiants de objet `NGC_1864`

4.4.2 Détection des instances dans le domaine de la microbiologie

La détection des entités nommées dans le domaine de la microbiologie est faite avec le logiciel GATE qui sera présenté dans la sous-section 4.7.1. Le principe de cette méthode étant le même que dans le domaine de l'astronomie, nous utilisons une liste d'objets (thésaurus) ainsi que des dictionnaires de nomenclatures. Par exemple, les bactéries sont identifiées à partir de thésaurus de la base NCBI[20]. Un extrait de ce thésaurus est présenté dans la figure 4.6. Cette liste présente des bactéries ainsi que leurs souches. Par exemple, `Helicobacter_Pylori` est le nom de la bactérie et `Helicobacter_Pylori_Shi470` est le nom d'une de ses souches. Une souche est une variante, ce qui signifie la même bactérie mais avec un génome différent. Les antibiotiques ont été identifiés grâce à une liste d'antibiotiques mise en ligne[21]. Les gènes quant à eux sont détectés par des nomenclatures car généralement un gène est de la forme suivante : `abcD`, c'est-à-dire, trois lettres en minuscules suivies d'une lettre en majuscule. La gestion de ces listes d'objets est expliquée dans la sous-section 4.7.2. La figure 4.7 présente un exemple de textes où tous les objets du domaine sont détectés.

Helicobacter pylori Shi470	chromosome
Heliobacterium modesticaldum Ice1	chromosome
Herminiimonas arsenicoxydans	chromosome
Herpetosiphon aurantiacus ATCC 23779	chromosome
Herpetosiphon aurantiacus ATCC 23779	plasmid pHAU01
Herpetosiphon aurantiacus ATCC 23779	plasmid pHAU02
Histophilus somni	plasmid p57/98
Histophilus somni	plasmid p9L
Hydrogenobaculum sp. Y04AAS1	chromosome
Hyphomonas neptunium ATCC 15444	chromosome
Idiomarina loihiensis L2TR	chromosome
Jannaschia sp. CCS1	chromosome
Jannaschia sp. CCS1	plasmid plasmid1
Janthinobacterium sp. Marseille	chromosome
Kineococcus radiotolerans SRS30216	chromosome
Kineococcus radiotolerans SRS30216	plasmid pKRAD01
Kineococcus radiotolerans SRS30216	plasmid pKRAD02
Klebsiella pneumoniae	plasmid 12

FIG. 4.6 – Un extrait de la classification des bactéries dans la base NCBI

Quelques erreurs ont été détectées dans l'identification des objets du domaine qui peuvent être expliquées comme suit :

– des erreurs dans le dictionnaire des nomenclatures : dans le dictionnaire de nomenclature, les gènes et les protéines possèdent les mêmes noms. Ils sont distingués par le fait que les

[20] http ://www.ncbi.nlm.nih.gov/genomes/genlist.cgi ?taxid=2&type=0&name=Complete%20Bacteria
[21] http ://www.emedexpert.com/lists/antibiotics.shtml

gènes commencent tous par des minuscules et les protéines par des majuscules. Toutefois, certains gènes comme par exemple `Locus_nfxD` commencent par une majuscule ;
- manque de nomenclatures pour retrouver les gènes mutants, par exemple les gènes mutants `A2143C, A2143G, A2144G` auraient pu être retrouvés avec la nomenclature `ANNNC` avec `A` et `C` deux majuscules et `N` un chiffre ;
- des erreurs du système :
  - confusion entre les codons et les aminoacides car leurs noms sont très semblables ;
  - confusion entre les protéines et antibiotiques. Par exemple avec `Penicillin-binding protein (PBP)`, le système a détecté `Penicillin` comme antibiotique et `Penicillin-binding protein (PBP)` comme protéine, au lieu de ne détecter que la protéine ;
  - mauvaises abréviations : le système Pactole a confondu la bactérie `Mycobacterium_Tuberculosis` et la maladie `tuberculosis`.

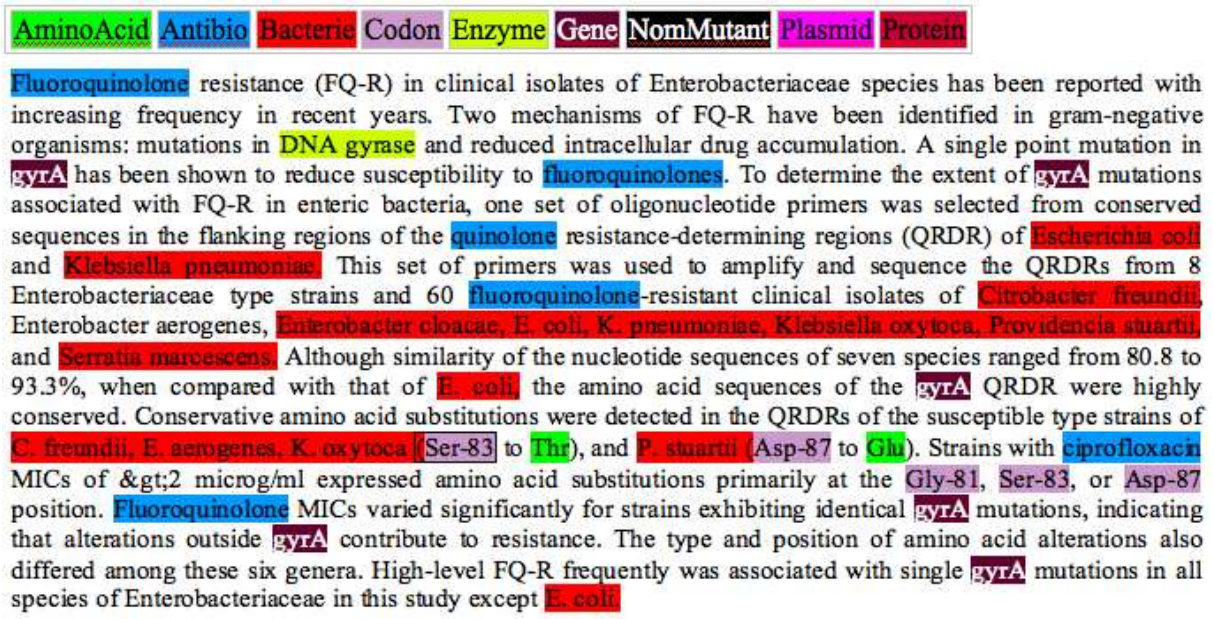


Fig. 4.7 – Identification de neuf types d'objets du domaine de la microbiologie à partir du corpus de textes

Prise en compte des synonymes dans le domaine de la microbiologie. Nous n'avons pas trouvé de ressources dans le domaine de la microbiologie qui nous permettent de prendre en compte des synonymes. Par contre, les experts nous ont conseillé de repérer les abréviations et de les regrouper. Par exemple, les termes « `Streptococcus_Pneumoniae` », « `S._Pneumoniae` » ou encore « `Streptococcus_P.` » représentent la même bactérie écrite de différentes manières et donc la même instance.

4.5 Identification des classes d'objets

Dans cette section, nous présentons notre méthode d'identification des classes d'objets. Cette méthode est composée de deux étapes. La première étape consiste à trouver pour chaque objet

du domaine sa classe. Généralement, la classe est retrouvée en exécutant une requête dans la base des experts. Si aucune classe n'est attribuée à un objet dans cette base alors la classe `Object_of_unknown_nature` lui est attribuée. La deuxième étape consiste à identifier la partie de la « hiérarchie source » qui contient les classes extraites de la première étape, c'est-à-dire, les classes qui sont reliées à au moins un objet. Dans les sous-sections suivantes, nous appliquons cette méthode dans les deux domaines spécifiques, l'astronomie et la microbiologie.

4.5.1 Identification des classes d'objets dans le domaine de l'astronomie

Dans le domaine de l'astronomie, la base SimBad est la plus grande base pour la classification des objets célestes hors du système solaire. Elle possède :

- 4 288 748 objets,
- 12 550 032 identifiants,
- 225 017 références bibliographiques,
- 6 042 878 citations d'objets dans les articles.

SimBad nous permet d'instancier les objets célestes identifiés dans des textes avec des classes prédéfinies par les astronomes. Un exemple de cette instanciation est présenté dans la figure 4.8 qui montre la classification de l'objet `SMC` dans la classe `Galaxy`. Les nouveaux objets `HH_24MMS, S140_IRS3, M33_X-9` identifiés des textes se voient affecter la classe `Object_of_unknown_nature`.

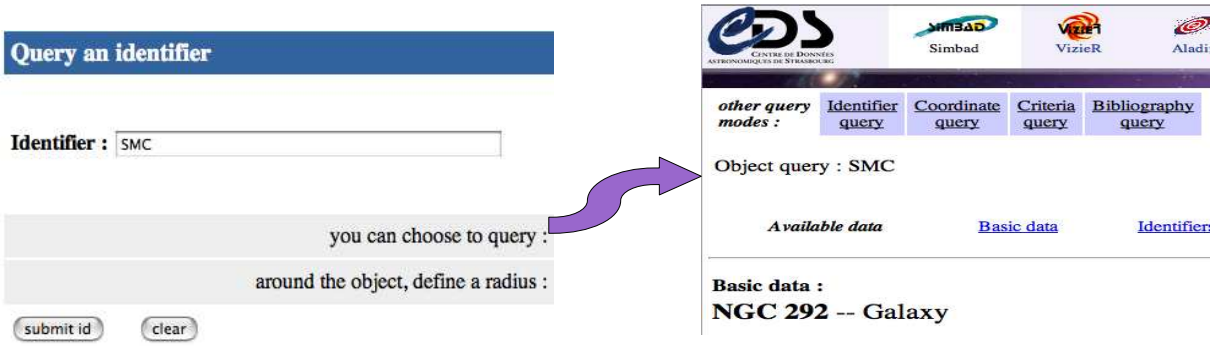


Fig. 4.8 – Classification de l'objet `SMC` dans la classe `Galaxy` par la base SimBad

La figure 4.9 présente une partie de cette classification telle que la classe `Region_defined_in_the_sky` est une super-classe de la classe `Underdense_region_of_the_univers` ou encore la classe `Cluster_of_Stars` est une super-classe des classes `Globular_Cluster` et `open_(galactic)_Cluster`... Comme défini précédemment, si un objet est dans l'extension d'une classe alors il appartient aussi à toutes ses super-classes.

4.5.2 Identification des classes d'objets dans le domaine de le microbiologie

Dans le domaine de la microbiologie, la base ncbi est la plus grande base pour la classification des bactéries dans des classes prédéfinies. Elle regroupe plus de 13890 bactéries classifiées dans 5929 classes. Pour extraire la « hiérarchie source » des bactéries, nous utilisons une partie de la base ncbi nommée ncbi Taxonomy. La figure 4.10 présente une partie de cette « hiérarchie source ». Par exemple, la classe `Bacilli` est une sous-classe de la classe `Firmicutes` qui elle même est une sous-classe de la classe `Bacteria` (voir en haut de la figure 4.10). La classe

multiple_object	mul	Composite object
· Region	reg	Region defined in the sky
· · Void	vid	Underdense region of the Univ
· SuperClG	SCG	Supercluster of Galaxies
· ClG	ClG	Cluster of Galaxies
· GroupG	GrG	Group of Galaxies
· · Compact_Gr_G	CGG	Compact Group of Galaxies
· PairG	PaG	Pair of Galaxies
· · IG	IG	Interacting Galaxies
· GlCl?	Gl?	Possible Globular Cluster
· Cl*	Cl*	Cluster of Stars
· · GlCl	GlC	Globular Cluster
· · OpCl	OpC	Open (galactic) Cluster
· Assoc*	As*	Association of Stars
· **	**	Double or multiple star

FIG. 4.9 – Une partie de la hiérarchie source SIMBAD dans le domaine de l'astronomie

`Bacilli` est aussi super-classe de la classe `Bacillales` qui est elle même super-classe de la classe `Alicyclobacillacae` et ainsi de suite.

Lineage (full): root; cellular organisms; Bacteria; Firmicutes

- **Bacilli** *Click on organism name to get more information.*
  - **Bacillales**
    - **Alicyclobacillaceae**
      - **Alicyclobacillus**
      - **Caldibacillus**
      - **unclassified Alicyclobacillaceae**
      - **environmental samples**
    - **Bacillaceae**
      - **Alkalibacillus**
      - **Amphibacillus**
      - **Amylobacillus**
      - **Anoxybacillus**
      - **Aquisalibacillus**
      - **Bacillus**
      - **Caldalkalibacillus**
      - **Caldaterra**

FIG. 4.10 – Une partie de la hiérarchie source NCBI TAXONOMY dans le domaine de la microbiologie

4.6 Identification des attributs binaires

Pour identifier les attributs binaires des objets, nous utilisons deux méthodes différentes d'identification. Lorsque les attributs binaires sont extraits d'un corpus de textes, nous nous appuyons sur l'hypothèse de Harris présentée dans la sous-section 3.2.3 pour regrouper ensemble les termes se trouvant dans les mêmes expressions syntaxiques et possédant des constituants supportant le même sens. Par exemple, un adjectif peut être le constituant porteur de sens pour les noms. Un autre exemple serait de prendre les verbes comme les constituants porteurs de sens. Il existe plusieurs possibilités, comme dans la phrase : *« Une très belle et grande maison construite en bois »*, dans laquelle le terme principal est `« maison »` et tous les autres mots sont des modifieurs de sens. Pour extraire ces constituants porteurs de sens, il faut tout d'abord trouver la catégorie grammaticale de chaque mot dans les phrases des textes, puis les dépendances qui les relient. Nous choisissons d'utiliser un analyseur partiel et robuste qui permet d'extraire l'analyse syntaxique de chaque phrase du corpus.

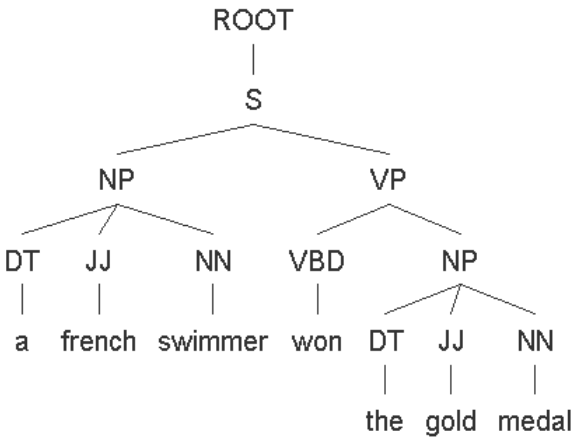


FIG. 4.11 – Analyse syntaxique de la phrase *« a french swimmer won the gold medal »*

Contrairement à une analyse complète d'une phrase, un analyseur de textes partiel n'extrait qu'une partie de l'analyse d'une phrase : celle qui est simple et qui n'est pas très ambiguë. Typiquement, seules les petits et simples syntagmes nominaux et verbaux sont générés. De même les dépendances syntaxiques ne sont extraites que s'il n'y a pas d'ambiguïté [Feldman et Sanger, 2007]. Prenons l'exemple de la phrase : *« a french swimmer won the gold medal »*, l'analyseur syntaxique de cette phrase produit l'arbre syntaxique correct de la figure 4.11. Maintenant, nous prenons une phrase plus compliquée, par exemple : *« a german won the gold medal with his horse »*. Normalement, il n'y a pas d'ambiguïté dans cette phrase car le pronom `his` désigne logiquement `a german` et non la `gold medal`. Malheureusement, l'analyseur syntaxique produit un mauvais arbre syntaxique (voir la figure 4.12), car il attribue `his horse` à la `gold medal` et non au `german`. Mais, certaines phrases sont ambiguës, par exemple : *« The otary plays ball with its nose »*, cette phrase possède une ambiguïté, le pronom `its` désigne-t-il `the otary` ou le nom `ball` ? L'arbre syntaxique extrait dans la figure 4.13 est syntaxiquement juste, mais sémantiquement faux, car le pronom `its` ne peut désigner le nom `ball`. Néanmoins, l'analyse complète d'une phrase demande beaucoup de mémoire et de temps. Pour un corpus de plusieurs milliers de textes, il vaut mieux utiliser l'analyse partielle. Il existe plusieurs travaux sur l'intérêt d'utiliser des analyseurs syntaxiques partiels [Tzoukermann *et al.*, 1997; Lager, 1998; Daelemans

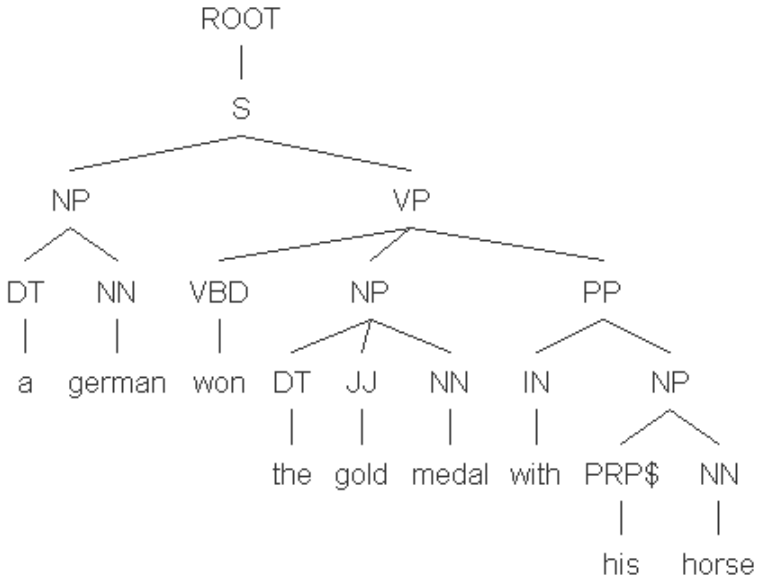


FIG. 4.12 – Analyse syntaxique de la phrase *« a german won the gold medal with his horse »*

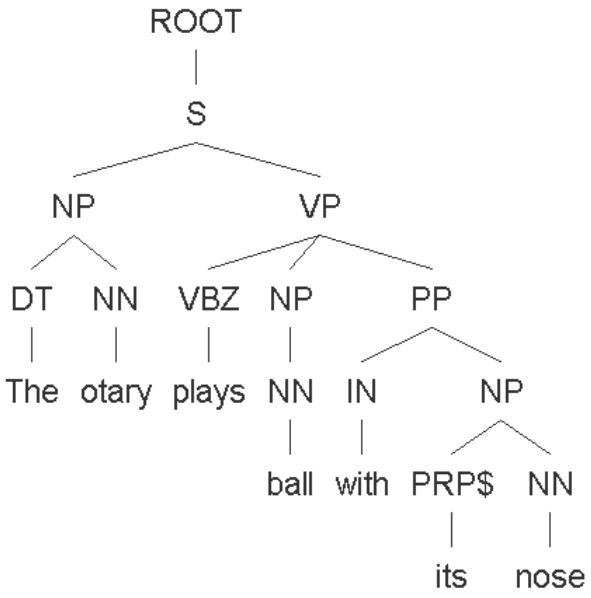


FIG. 4.13 – Analyse syntaxique de la phrase *« The otary plays ball with its nose »*

et al., 1999; Punyakanok et Roth, 2000].

4.6.1 L'analyseur syntaxique STANFORD PARSER

Dans cette thèse, nous choisissons d'utiliser l'analyseur syntaxique STANFORD PARSER[22] [Levy et Manning, 2004; de Marneffe *et al.*, 2006]. Cet analyseur s'appuie sur une méthode

[22] http ://nlp.stanford.edu/software/lex-parser.shtml

statistique pour extraire tout d'abord la catégorie grammaticale de chaque mot, puis l'arbre syntaxique de chaque phrase dans un corpus de textes. Ensuite, il permet d'extraire des dépendances syntaxiques entre les termes. Ces dépendances sont identifiées après la construction de l'arbre syntaxique de la phrase. Par exemple de la phrase *« a french swimmer won the gold medal »*, il extrait les catégories grammaticales : *« a/DT french/JJ swimmer/NN won/VBD the/DT gold/JJ medal/NN »*, puis l'arbre présenté dans la figure 4.11. Ensuite, pour chaque dépendance syntaxique, il utilise l'outil « Stanford Parser grammatical relation browser »[23] qui permet d'extraire le nom de la relation. Voici les dépendances extraites de la phrase *« a french swimmer won the gold medal »* :

- det(swimmer-3,a-1) : cette relation veut dire que le mot `a` qui se trouve à la 1ère position de la phrase est le « déterminant » du mot `swimmer` qui se trouve à la 3ème position de la phrase,
- amod(swimmer-3,french-2) : cette relation veut dire que le mot `french` qui se trouve à la 2ème position de la phrase est un « modifieur adverbial » du mot `swimmer` qui se trouve à la 3ème position de la phrase,
- nsubj(won-4,swimmer-3) : cette relation veut dire que le mot `swimmer` qui se trouve à la 3ème position de la phrase est le « sujet » du verbe `won` qui se trouve à la 4ème position de la phrase,
- det(medal-7,the-5) : cette relation veut dire que le mot `the` qui se trouve à la 5ème position de la phrase est le « déterminant » du mot `medal` qui se trouve à la 7ème position de la phrase,
- amod(medal-7,gold-6) : cette relation veut dire que le mot `gold` qui se trouve à la 6ème position de la phrase est un « modifieur adverbial » du mot `medal` qui se trouve à la 7ème position de la phrase,
- dobj(won-4,medal-7) : cette relation veut dire que le mot `medal` qui se trouve à la 7ème position de la phrase est le « complément d'objet direct » du verbe `won` qui se trouve à la 4ème position de la phrase.

La deuxième méthode d'identification des attributs binaires se fait à partir d'une base de données. Elle consiste à interroger différentes bases de données pour identifier les attributs de ces bases.

4.6.2 Identification des attributs binaires en astronomie

Dans le domaine de l'astronomie, les objets célestes sont définis par des attributs binaires extraits de corpus de textes. Mais, il faut définir ce qui peut être considéré comme un attribut d'objet. Nous choisissons (comme dit précédemment) d'utiliser une approche s'appuyant sur l'hypothèse de Harris, mais nous devons choisir les constituants porteurs de sens qui seront considérés comme attributs binaires des objets célestes. Prenons par exemple des phrases présentées dans la figure 4.14.

Dans la phrase *« We report the discovery of strong flaring of the object `HR2517` »*, le fait que l'objet `HR2517` « *can flare* » (*i.e.* `flare` veut dire avoir une éruption de plasma à la surface de l'objet) fait que cet objet est un type particulier de la classe `Star`.

Dans la deuxième phrase *« We report results from two COMPTEL observations, in June and October 1991, of the quasars 3C_273 containing binary star M_83. »* Puisque l'objet `3C_273` « *can contain* », cela veut dire que l'objet `3C_273` ne pas être une `Star`.

Les attributs sont extraits en effectuant une analyse syntaxique du corpus de textes avec le

[23] http ://nlp.stanford.edu/software/lex-parser.shtml

We report the discovery of strong flaring of the star HR2517. The evidence is based on Stroemgren differential uvby photometry spanning more than a decade. We discuss the behaviour of HR2517 in terms of two models, viz. a rooted bright spot suddenly appearing and developing with modulation in photospheric temperature producing periodic light and profile variations as the star rotates, or that HR2517 is an eccentric (P~33-34d) high-mass X-ray binary (HMXB) with neutron star companion--one of the brightest such systems known.

We report results from two COMPTEL observations, in June and October 1991, of the quasars 3C 273 containing binary star M 83. The COMPTEL instrument, on board the Compton Gamma Ray Observatory, is sensitive over the energy range 0.75-30.0MeV. Our data show that at MeV energies 3C 273 is variable on a time scale of months.

FIG. 4.14 – Deux extraits de textes de notre corpus de textes dans le domaine de l'astronomie

STANFORD PARSER (présenté dans la sous-section 4.6.1). Pour présenter les étapes d'identification des attributs binaires des objets, nous prenons l'exemple de la phrase :

« We report results from two COMPTEL observations, in June and October 1991, of the quasars 3C_273 containing binary star M_83. »

Application du parseur. Le STANFORD PARSER extrait, à partir de chaque phrase du corpus de textes, son arbre syntaxique, puis les dépendances entre les verbes et leurs sujets, leurs objets, leurs compléments et leurs compléments prépositionnels.

Le STANFORD PARSER affecte à chaque mot sa catégorie grammaticale : « We/PRP report/VBP results/NNS from/IN two/CD COMPTEL/JJ observations/NNS ,/, in/IN June/NNP and/CC October/NNP 1991/CD ,/, of/IN the/DT quasars/NN 3C_273/CD containing/VBG binary/JJ star/NN M_83/CD./. » Puis, l'analyseur construit l'arbre syntaxique de la phrase :

```
(ROOT
  (S
  (NP (PRP We)) (VP (VBP report) (NP (NNS results))
    (PP (IN from) (NP (CD two) (JJ COMPTEL) (NNS observations)))
      (, ,)
      (PP (IN in)
        (NP (NP (NP (NNP June)) (CC and) (NP (NNP October) (CD 1991)))
      (, ,)
    (PP (IN of) (NP (NP (DT the) (NN quasars) (QP (CD 3C_273))
     (VP (VBG containing) (NP (DT the) (JJ binary) (NN stars) (QP (CD M_83)))))))))
  (. .)))
```

Ensuite, l'analyseur extrait les dépendances entrel' mots à partir de l'arbre syntaxique. Le processus PACTOLE filtre les dépendances extraites par l'analyseur syntaxique pour ne garder que les dépendances du type `(objet_céleste,attribut_binaire)` :

```
det(3C_273-24,the-22)
amod(3C_273-18,quasars-17)
nsubj(containing-20,3C_273-18)
dobj(containing-20,M_83-23)
```

Ensuite, le système ne sélectionne que les dépendances les plus significatives (ces dépendances sont choisies par les astronomes) :

```
subject(containing-20,3C_273-18)
direct_object(containing-20,M_83-23)
```

Enfin, le processus PACTOLE transforme les dépendances en paires. Ainsi, la paire `(isContaining,3C_273)` est dérivée de la dépendance `subject(containing-20,3C_273-18)` et la paire `(isContained,M_83)` est dérivée de la dépendance `direct_object(containing-20,M_83-23)`. A noter que le verbe apparaît de deux façons différentes, la première lorsqu'il est associé à son sujet et la seconde lorsqu'il est associé à son complément d'objet.

La plupart des dépendances `(objet_céleste,attribut_binaire)` ne sont que des artefacts linguistiques et ne permettent pas réellement de décrire l'objet. Pour ne sélectionner que les dépendances pertinentes, le système définit des filtres avant l'étape de classification. Le premier filtre ne retient que les attributs qui apparaissent au moins deux fois avec un objet (ceci réduit aussi le bruit introduit par les erreurs de l'analyseur syntaxique). Le deuxième filtre regroupe les synonymes. Par exemple, les attributs `isConsisting`, `isContaining`, `isIncluding`... sont regroupés dans un seul attribut noté `isIncluding`. Le troisième filtre qui est un filtre manuel consiste à présenter les attributs aux experts afin qu'ils sélectionnent les attributs les plus significatifs dans le domaine. Cette dernière étape peut être revue tout au long du processus, car les astronomes peuvent considérer un attribut comme étant intéressant et s'apercevoir qu'il ne l'est pas (par exemple les attributs `isPerforming` ou `isOscillating`) ou au contraire découvrir qu'un attribut est intéressant après l'étape de construction de hiérarchie (par exemple l'attribut `isRotating`). Nous présenterons dans la section 6.3 les différentes opérations d'interaction entre le système PACTOLE et les experts du domaines.

Découverte d'unités de connaissances. Cette étape a permis au système de découvrir certaines corrélations entre les objets célestes et les attributs binaires qui n'existaient pas dans la base SIMBAD. Les astronomes ont défini ces corrélations comme étant de « nouvelles unités de connaissances » du domaine. Nous donnons quelques exemples de ces corrélations :

- les objets `59_Aurigae` et `V1208_Aql` sont associés à l'attribut `isPulsing`,
- l'objet `MM_Herculis` est associé à l'attribut `isEclipsing`,
- les objets `AB_Dor` et `OJ_287` sont associés à l'attribut `isFlaring`.

4.6.3 Identification des attributs binaires en microbiologie

Les objets dans le domaine de la microbiologie peuvent être décrits par des attributs binaires. Mais, contrairement au domaine de l'astronomie, nous n'avons pas pu identifier ces attributs binaires à partir du corpus de textes donné par les experts. Les experts nous ont conseillé de les identifier à partir de bases de données, telles que NCBI et « The pathogenic bacteria database »[24]. Les trois types d'attributs sur lesquels nous avons travaillé sont :

Les attributs des bactéries. Les microbiologistes nous ont expliqué que la résistance des bactéries dépendait de leurs formes, de leur gram, de leur interaction avec l'oxygène, de leur mobilité, ... Afin d'extraire ces attributs binaires, nous utilisons la base NCBI. Nous avons sélectionné 13 attributs binaires pour les bactéries. Par exemple, la bactérie `Helicobacter_Pylori` possède l'attribut binaire `hasNegativeGram`, ou encore la bactérie `Klebsiella_Pneumoniae` possède l'attribut binaire `isAnaerobic`.

Les attributs des gènes. Les microbiologistes ont choisi de représenter les gènes par leurs fonctions dans les bactéries. Ce sont ces fonctions qui peuvent expliquer la résistance des bac-

[24] http ://bac.hs.med.kyoto-u.ac.jp/

téries aux antibiotiques. Nous avons utilisé la base GENE ONTOLOGY[25], afin d'extraire 12 attributs binaires pour les gènes. Par exemple, le gène `23S_rRNA` possède l'attribut binaire `isNucleicAcidBinding`. L'attribut `isNucleicAcidBinding` veut dire que le gène produit une séquence qui s'attache à un acide nucléique.

Les attributs des antibiotiques. Des chimistes nous ont aidé à extraire les attributs binaires des antibiotiques. Les antibiotiques sont des ligands, ce qui signifie des molécules capables d'interagir avec une protéine au sein d'un organisme biologique. Les chimistes possèdent une classification appelée classification des ligands que nous utilisons dans cette application. Nous extrayons 24 attributs pour les antibiotiques. Par exemple, l'antibiotique `Ciprofloxacin` possède l'attribut binaire `Fraction of Rotatable bonds =1`. L'attribut `Fraction of Rotatable bonds =1` veut dire que l'antibiotique `Ciprofloxacin` possède une fraction de liaisons capables de rotation. Ces attributs peuvent servir à expliquer les interactions entre les bactéries et les antibiotiques.

4.7 Identification des attributs relationnels

Prenons la phrase de la figure 4.7 : « *To determine the extent of gyrA mutations associated with FQ-R in enteric bacteria, one set of oligonucleotide primers was selected from conserved sequences in the flanking regions of the quinolone resistance-determining regions (QRDR) of Escherichia coli and Klebsiella pneumoniae.* » Cette phrase ne comporte pas d'attributs binaires. En revanche elle comporte des relations entre les différents objets du domaine. Elle explique la résistance des deux bactéries `Escherichia_Coli` et `Klebsiella_Pneumoniae` à l'antibiotique `Quinolone` par mutation du gène `gyrA`. Ainsi, nous pouvons identifier la relation `isResisting` entre bactéries et antibiotiques et la relation `isPartOfGenomeOf` entre les gènes et les bactéries. Cette identification est faite par un logiciel de Traitement Automatique de la Langue Naturelle (TALN) et d'Extraction d'information (EI), le logiciel GATE.

4.7.1 Le logiciel GATE

GATE (General Architecture for Text Engineering) [Crane *et al.*, 2005; Davies *et al.*, 2005] a été développé à l'université de Sheffield depuis 1995 et a été utilisé dans plusieurs projets de recherche. GATE peut être considéré comme une architecture de logiciel pour le traitement automatique de la langue. Le terme « architecture de logiciel » est utilisé ici pour désigner une l'infrastructure pour le développement de logiciel, en incluant des environnements et des cadres de développement.

Le logiciel GATE est une architecture qui contient plusieurs composants qui sont successivement appliqués aux textes. Ces composants sont de six types différents :

1. Tokeniser : c'est un outil qui découpe le texte simplement en repérant les numéros, les mots et les ponctuations.
2. Stemmer : c'est un outil qui extrait les lemmes des mots pour 11 langues européennes, dont l'anglais, le français, le néerlandais, ...
3. Gazetteer : c'est un outil qui recherche les entités nommées. Il utilise un ensemble de listes avec un fichier index nommé `lists.def` qui permet d'y accéder. Chaque liste possède une

[25] http ://www.gèneontology.org/

entrée par ligne, cette entrée représente un terme. Comme par exemple, la liste des « mountain.lst » possède les entrées : `Alps, andes, Himalayas, Pyrenees, Snowdonia,...` Ces termes peuvent être très différents, c'est-à-dire de plusieurs types, par exemple, des montagnes, des villes... Pour les distinguer, il faut définir pour chaque liste, un type principal et optionnellement un type secondaire. Chaque texte est annoté avec d'abord le type principal puis le type secondaire. Des règles de grammaire déterminent quel type est utilisé dans les cas particuliers.

4. Sentence Splitter : c'est un outil qui découpe le texte en phrases. Il utilise l'outil « gazetteer » pour reconnaître les abréviations et distinguer entre un point de fin de phrase et un point d'abréviation par exemple.
5. POS Tagger : c'est un outil qui annote chaque mot du texte par sa catégorie grammaticale, il utilise pour cela une version modifiée de « the Brill tagger » [Brill, 1995],
6. Transducer : c'est un outil sémantique qui s'appuie sur la grammaire « Java Annotation Pattern Engine » (JAPE). Il contient des règles pour annoter des relations dans les phrases des corpus de textes.

Pour notre travail, les deux outils les plus importants dans le logiciel GATE sont l'outil « gazetteer » qui permet d'extraire le vocabulaire du domaine et, le « transducer » qui extrait les relations à partir des textes.

4.7.2 Identification des attributs relationnels dans le domaine de la microbiologie

L'identification des relations transversales est faite en utilisant l'outil GATE (présenté dans la sous-section précédente).

Détection d'entités nommées. La détection d'entités nommées dans le domaine de la microbiologie a été présentée dans la sous-section 4.4.2. Dans cette partie, nous expliquons la gestion des listes d'objets. Le « gazetteer » dans le logiciel GATE possède toutes les listes des objets qu'il doit extraire dans le fichier `lists.def`. Le fichier `lists.def` est de la forme :

```
aminoacid.lst :aminoacid
antibio.lst :antibio
bacteries.lst :bacterie
codons.lst :codon
diseases.lst :disease
enzymes.lst :enzyme
genes.lst :gene :protein
mutants.lst :mutant
plasmides.lst :plasmide
```

L'application du « gazetteer » sur la phrase *« The genes conferring resistance to doxorubicin and daunorubicin in S. peucetius have been sequenced. »* donne le texte annoté suivant :

<paragraph><Token category=DT kind=word length=3 orth=upperInitail stem=the>**The**</Token> <Token category=NNS kind=word lengt=5 orth=lowercase stem=gene>**genes**</Token><Token category=VBG kind=word lengt=10 orth=lowercase stem=confer>**conferring**</Token> <Token category=NN kind=word lengt=10 orth=lowercase stem=resist>**resistance**</Token><Token category=TO

kind=word length=2 orth=lowercase stem=to>**to**</Token> <Token category=NN kind=word lengt=11 orth=lowercase stem=doxorubicin><Lookup majorType=antibio>**doxorubicin**</Lookup></Token> <Token category=CC kind=word lengt=3 orth=lowercase stem=and>**and**</Token><Token category=NN kind=word length=12 orth=lowercase stem=daunorubicin><Lookup majorType=antibio>**daunorubicin**</Lookup></Token> <Token category=IN kind=word length=2 stem=in>**in**</Token><Lookup majorType=bacterie><Token category==NNP kind=word length=1 orth=upperInitial stem=s>**S**</Token><Token category=. kind=punctuation length=1 stem=.>.</Token><Token category=JJ kind=word length=9 orth=lowercase stem=peucetius>**peucetius**</Token> </Lookup> <Token category=VBP kind=word length=4 orth=lowercase stem=have>**have**</Token> <Token category=VBN kind=word length=4 stem=been>**been**</Token> <Token category=JJ kind=word length=9 orth=lowercase stem=sequenc>**sequenced**</Token><Token category=. kind=punctuation length=1 stem=.>**.**</Token></paragraph>

Identification d'une relation entre deux entités nommées. Cette étape consiste à identifier des expressions régulières en utilisant la grammaire « Java Annotation Pattern Engine » (Jape). Cette grammaire est constituée d'un ensemble de règles du type `patrons/action`. Chaque règle est composée de deux parties : la Partie Gauche (pg) et la Partie Droite (pd) séparées par la chaîne de caractères « `->` ». La pg de la règle fournit l'expression régulière qui doit être retrouvée et peut contenir des opérateurs d'expressions régulières du type (*, ?, +). La pd se compose des déclarations de manipulation d'annotation et fournit l'étiquette qui doit étiqueter la phrase si la pg est vérifiée.

```
Rule :Bacteries
(
{Lookup.majorType == "bacterie"}
) :bacterie
-> :bacterie.Bacterie = {rule="bacterie1"}
Rule :Antibio
(
{Lookup.majorType == "antibio"}
) :antibio
-> :antibio.Antibio = {rule="antibio1"}
Rule :resistant
(
{Token.string == "resistance"}
({Token.category == IN}|{Token.category == TO})
{Antibio}
({Token.kind == punctuation}{Antibio}({Token.kind==punctuation}) ?)*
({Token.category == AND}{Antibio}) ?
{Token.category==IN}
{Bacterie}
) :resistant_label
-> :resistant_label.Resistant = {rule="resistant3"}
```

Tab. 4.2 – Exemple de règles d'identification. La première règle identifie des bactéries, la deuxième des antibiotiques et la troisième extrait la relation de résistance

Dans l'exemple précédent, les règles de la table 4.2 permettent de détecter la relation de résistance entre les deux antibiotiques `doxorubicin` et `daunorubicin` et la bactérie `S. peucetius`. Dans cet exemple, la première règle « `Rule :Bacteries` » permet d'annoter un texte avec l'annotation « `Bacterie` » si ce texte contient la balise `{Lookup.majorType == "bacterie"}`. La deuxième règle « `Rule :Antibio` » permet d'annoter un texte avec l'annotation « `Antibio` » si ce texte contient la balise `{Lookup.majorType == "antibio"}`. Et enfin, la troisième règle « `Rule :resistant` » permet d'annoter un texte avec l'annotation « `Resistant` » si dans ce texte, le patron syntaxique suivant est retrouvé : d'abord le terme « `resistance` » (`{Token.string == "resistance"}`), suivi d'un terme des catégories « `IN` » ou « `TO` » (`{Token.category == IN}|{Token.category == TO}`), suivie d'une balise « `Antibio` » (`{Antibio}`), suivie peut-être d'une ou de plusieurs ponctuations et balises « `Antibio` » (`({Token.kind == punctuation}{Antibio}({Token.kind==punctuation})?)*`), suivi d'un terme de la catégorie « `AND` » et d'une balise « `Antibio` » (`({Token.category == AND}{Antibio})?`), suivi d'une balise « `Bacterie` » (`{Bacterie}`).

Chacune de ces règles est stockée dans un fichier « NomFichier.jape ». En microbiologie, nous avons défini plusieurs fichiers. Par exemple le fichier « MicroBioObjects.jape » contient les règles pour retrouver les objets du domaine. Les deux premières règles de la table 4.2 appartiennent à ce fichier. Le fichier « MicroBioMutation.jape » se compose des règles pour retrouver la relation de mutation. Le fichier « MicroBioResistant.jape » contient les règles pour retrouver la relation de résistance. C'est dans ce fichier que se trouve la dernière règle de la table 4.2. Enfin, le fichier « MicroBioResist-Mutat.jape » contient les règles pour retrouver les relations de résistance et de mutation,... Le logiciel dispose aussi d'un fichier nommé « main.jape » qui définit un ordre d'exécution entre les différents fichiers de règles (les « NomFichier.jape »). La table 4.3 présente un exemple d'un fichier « main.jape » qui définit l'ordre entre les fichiers que nous avons cité en exemple.

```
// Niveau 0
MicroBioObjects
// Niveau 1
MicroBioMutation
MicroBioResistant
// Niveau 2
MicroBioResist-Mutat
```

TAB. 4.3 – Exemple d'un fichier « main.jape » qui définit que le programme doit d'abord exécuter toutes les règles du fichier « MicroBioObjects.jape » puis celles des deux fichiers « MicroBioMutation.jape » et « MicroBioResistant.jape » et enfin celles du fichier « MicroBioResist-Mutat.jape »

L'application de toutes les règles définies dans la table 4.2 et ordonnées par le fichier « main.jape » (présenté dans la table 4.3) donne l'annotation suivante :

```
   The genes conferring
<Resistant>
resistance to
<Antibio>doxorubicin</Antibio>
and
<Antibio>daunorubicin</Antibio>
in
<Bacterie>S. peucetius</Bacterie>
</Resistant>
```

have been sequenced.

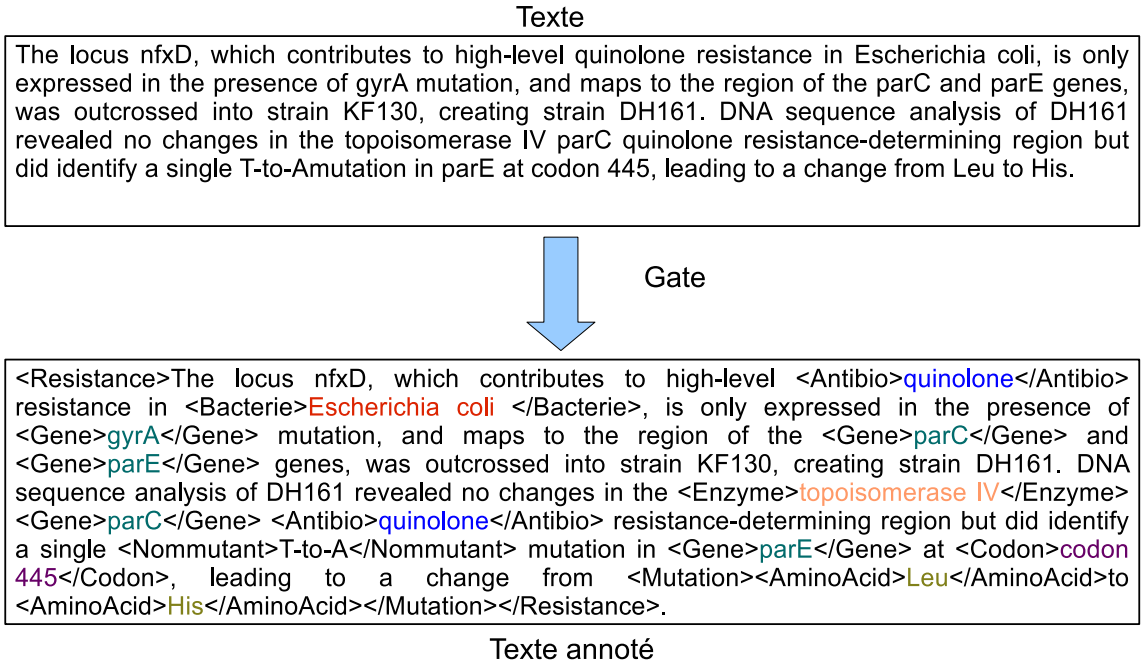


FIG. 4.15 – Détection des objets du domaine et des relations de résistance (entre les bactéries et les antibiotiques) et de mutation (des gènes) avec le logiciel GATE

La figure 4.15 présente un exemple de l'utilisation du logiciel GATE sur un texte pour l'identification des objets et des relations entre ces objets. Le système doit d'abord identifier tous les types d'objets du domaine. Ici nous avons identifié les objets suivants : les antibiotiques avec la balise « `Antibio` », les bactéries avec la balise « `Bacterie` », les gènes avec la balise « `Gene` », les enzymes avec la balise « `Enzyme` », les gènes mutants avec la balise « `NomMutant` », les codons avec la balise « `Codon` » et enfin les acides aminés avec la balise « `AminoAcid` ». Le système a aussi extrait plusieurs types de relations, telles que la relation de mutation avec la balise « `Mutation` » et la relation de résistance avec la balise « `Resistance` ». Dans cette thèse nous nous sommes limités à l'identification des trois types d'objets (bactéries, antibiotiques et gènes) et aux relations « `isPartOfGenomeOf` » entre les gènes et les bactéries et « `isResisting` » entre les bactéries et les antibiotiques.

L'ensemble des règles a été défini manuellement en étudiant le corpus de textes afin de comprendre les différents types d'occurrences des relations étudiées. Puis, pour chaque type d'objets et pour chaque relation nous avons défini des règles d'identification. Nous donnons ici le nombre de règles qui ont été définies en plus des règles déjà existantes dans le logiciel GATE :

- 30 règles pour extraire les différents types d'objets et pour regrouper un objet avec toutes ses abréviations,
- 37 règles pour détecter la relation `isPartOfGenomeOf` entre les gènes et bactéries,
- 41 règles pour extraire la relation `isResisting` entre les bactéries et les antibiotiques,
- 2 règles pour détecter l'appartenance et la résistance

– et 1 règle « not resistance » pour éviter de prendre en compte des phrases du type « *the bacteria B doesn't resist to the antibiotic A.* »

4.8 Conclusion

Dans ce chapitre, nous avons extrait de plusieurs ressources hétérogènes un (ou des) ensemble(s) d'objets ainsi que différents types de descripteurs d'objets. Nous avons expliqué que les descripteurs d'objets sont choisis à l'aide des experts du domaine, d'après les ressources dont on dispose, mais aussi en fonction de notre méthode de fouille : l'Analyse Formelle de Concepts AFC. Nous avons appliqué nos méthodes de détection et d'identification de ces différents descripteurs d'objets dans deux domaines d'application : l'astronomie et la microbiologie. Dans le domaine de l'astronomie, nous avons défini l'ensemble des objets célestes comme étant l'ensemble des objets du domaine. Puis, nous avons extrait une hiérarchie source du domaine où les objets célestes sont affectés manuellement par les astronomes à une hiérarchie de classes prédéfinies ; ce descripteur est noté *DO1*. Ensuite, nous avons extrait des attributs binaires à partir du corpus de textes à l'aide d'un analyseur syntaxique ; ce descripteur est noté *DO2*. Dans le domaine de la microbiologie, nous avons défini trois types d'objets du domaine : les antibiotiques, les gènes et les bactéries. Nous avons également extrait une hiérarchie source où les bactéries ont été affectées manuellement à une hiérarchie de classes prédéfinies ; ce descripteur est noté *DO1*. Puis, pour chaque type d'objets, nous avons extrait des attributs binaires à partir de différentes bases de données, ce descripteur est noté *DO2*. Enfin, nous avons extrait deux types de relations transversales entre les objets, la relation `isResisting` entre les bactéries et les antibiotiques et la relation `isPartOfGenomeOf` entre les gènes et bactéries à l'aide du logiciel GATE ; ce descripteur est noté *DO3*. Dans le chapitre suivant, ces différents descripteurs serviront à définir les classes des experts, à enrichir leurs hiérarchies sources, puis à construire une ontologie du domaine représentée en LD et implémentée en OWL.

Chapitre 5

Méthodologie Pactole : Extraction de connaissances à partir de ressources

Sommaire

Introduction

Ce chapitre est découpé comme suit : la première section 5.1 présente l'application de PACTOLE dans le domaine de l'astronomie. Ensuite, la section 5.2 détaille l'application de PACTOLE dans le domaine de la microbiologie. Puis, la section 5.4 cite des travaux utilisant aussi l'AFC pour construire une ontologie à partir de ressources textuelles. Et enfin, la section 5.3 explique le passage du schéma d'ontologie à une ontologie avec le langage des logiques de descriptions $\mathcal{FLE}$. Cette section présente aussi les types de questions que les experts peuvent poser ainsi que les raisonnements appliqués pour y répondre.

5.1 Construction du schéma d'ontologie dans le domaine de l'astronomie avec Pactole

Nous choisissons d'utiliser l'Analyse Formelle de Concept (AFC) pour la construction des hiérarchies d'objets, car elle nous offre différents bénéfices. Premièrement, cette technique présente plusieurs opérations de maintenance et de mise à jour de la hiérarchie de concepts résultante en ajoutant des objets ou des attributs du contexte. Deuxièmement, si la hiérarchie de concepts change (parce que par exemple, le corpus de textes a changé), le schéma de l'ontologie évoluera de façon correcte et consistante. Ainsi, la représentation du schéma de l'ontologie en logique de descriptions est faite sans risque d'inconsistance.

Afin de pouvoir construire une hiérarchie d'objets qui prend en compte toutes les descriptions des objets extraits des différentes ressources, il faut construire des contextes formels à partir de chaque descripteur d'objets.

5.1.1 Construction d'un treillis de concepts à partir des classes d'objets

Le premier type de Descripteur d'Objets (*DO1*) est constitué de la hiérarchie source $(M_1,\sqsubseteq)$. Les objets sont classifiés dans des classes prédéfinies et ces classes sont organisées dans un arbre, ce qui signifie qu'il n'y a pas d'héritage multiple dans cette hiérarchie. Nous souhaitons transformer cet arbre en un treillis de concepts et pour cela, il faut construire un contexte formel.

Soit G l'ensemble des objets et $\subseteq$ la relation d'inclusion sur l'ensemble des parties de G. Le couple $(G,\sqsubseteq)$ dénote un ensemble ordonné ainsi que la hiérarchie source $(M_1,\sqsubseteq)$. Ainsi nous pouvons transformer la hiérarchie source en un contexte formel $\mathbb{K}_1 := (G, M_1, I_1)$ défini comme suit : G est l'ensemble des objets du domaine, M_1 est l'ensemble des classes des objets et I_1 est la relation qui assigne à chaque objet sa classe et toutes les super-classes de cette classe dans la hiérarchie. Un exemple du contexte $\mathbb{K}_1$ d'objets célestes et de leurs classes extraites de la base SIMBAD est présenté dans le tableau 5.1, la figure 5.1 montre le treillis de concepts correspondant.

TAB. 5.1 – Le contexte $\mathbb{K}_1=(G, M_1, I_1)$ décrivant les objets et leurs classes extraites de la base SIMBAD. `Asso._of_Stars` est une abréviation de `Association_of_Stars`

	Classes SIMBAD					
	Quasar	Galaxy	Asso._of_Stars	T_Tau_type_Star	Eclipsing_Binary	Star
Cen_A		×				
3C_273	×	×				
TWA			×			
Per_OB2			×			
T_Tauri				×		×
Y_Cygni					×	×
V773_Tau				×		×
Algol					×	×

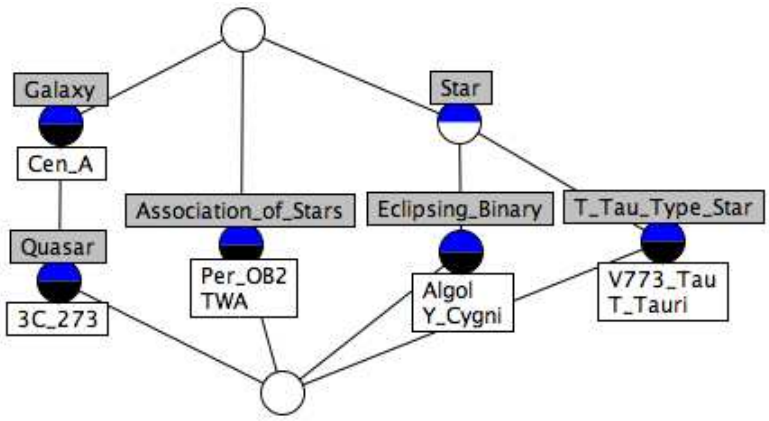


FIG. 5.1 – Le treillis correspondant au contexte $\mathbb{K}_1=(G, M_1, I_1)$

5.1.2 Construction d'un treillis à partir des attributs binaires

Le deuxième Descripteur d'Objets *(DO2)* est constitué des attributs extraits des textes. Ce descripteur peut être utilisé pour la construction du contexte formel $\mathbb{K}_2 := (G, M_2, I_2)$ tel que : G est l'ensemble des objets célestes, M_2 l'ensemble des attributs extraits des textes et $I_2 \subseteq G \times M_2$ où $I_2(g, m_2)$ veut dire que l'objet g possède l'attribut m_2 (ici l'ensemble G des objets est le même ensemble dans les deux contextes $\mathbb{K}_1$ et $\mathbb{K}_2$). Nous donnons un exemple du contexte $\mathbb{K}_2$ d'objets célestes et de leurs attributs extraits des textes. Il est présenté dans le tableau 5.2 ainsi que le treillis de concepts correspondant dans à la figure 5.2.

	Attributs binaires						
	isObserved	isIncluding	isEmitting	isEclipsing	isExpanding	isIncluded	isFlaring
Cen_A	×	×					
3C_273	×	×	×				
TWA	×	×			×		
Per_OB2	×	×					
T_Tauri	×		×			×	×
Y_Cygni	×		×	×		×	
V773_Tau	×		×			×	×
Algol	×		×	×		×	

TAB. 5.2 – Le contexte $\mathbb{K}_2=(G, M_2, I_2)$ décrivant les objets et leurs attributs binaires extraits des textes

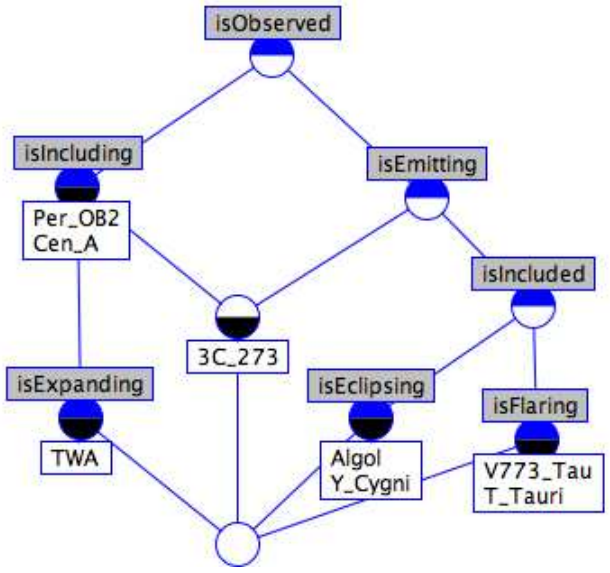


FIG. 5.2 – Le treillis correspondant au contexte $\mathbb{K}_2$=(G, M_2, I_2)

5.1.3 Affectation d'attributs binaires à des classes d'objets

Nous proposons une méthode d'affectation des attributs binaires extraits du corpus de textes aux classes de la hiérarchie source de la base SIMBAD. Cette méthode utilise une propriété de l'AFC qui est l'« apposition de contextes » définie par [Ganter et Wille, 1999] et présentée dans la section 3.3.4.

L'apposition de contextes $\mathbb{K}_o = (G_o, M_o, I_o)$ des deux contextes $\mathbb{K}_1 = (G, M_1, I_1)$ et $\mathbb{K}_2 = (G, M_2, I_2)$ peut se voir comme suit : G_o est l'ensemble des objets (le même ensemble pour $\mathbb{K}_1$ et $\mathbb{K}_2$), $M_o := M_1 \cup M_2$ où $M_1 \cap M_2 = \emptyset$ et où M_1 est l'ensemble des classes du contexte $\mathbb{K}_1$ (extraits de la hiérarchie de la base SIMBAD) et M_2 est l'ensemble des attributs du contexte $\mathbb{K}_2$ (extraits du corpus de textes) et $I_o := I_1 \cup I_2$. L'apposition de contextes $\mathbb{K}_o$ est présentée dans le tableau 5.3 et le treillis de concepts résultant est présenté à la figure 5.3.

Dans le treillis $\underline{\mathfrak{B}}(\mathbb{K}_o)$, les classes se voient associer des attributs binaires. Par exemple le concept {`Star, Eclipsing_Binary, isObserved, isEmitting, isIncluded, isEclipsing`} {`Algol, Y_Cygni`} permet d'associer à la classe `Eclipsing_Binary`, de la base SIMBAD, les attributs binaires {`isObserved, isEmitting, isIncluded, isEclipsing`}. Les experts peuvent interpréter ces éléments de deux façons différentes. La première est de considérer que les attributs binaires {`isObserved, isEmitting, isIncluded, isEclipsing`} définissent la classe `Eclipsing_Binary` qui est une sous-classe de la classe `Star`. La deuxième est de considérer que les attributs binaires {`isObserved, isEmitting, isIncluded, isEclipsing`} définissent une sous-classe de la classe `Eclipsing_Binary` et ainsi créent une nouvelle classe dans la hiérarchie source. L'apposition de contextes permet donc, non seulement, d'affecter des attributs binaires aux classes prédéfinies de la hiérarchie source, mais aussi d'enrichir cette hiérarchie source avec de nouvelles classes.

Le treillis de la figure 5.3 constitue le schéma d'ontologie dans le domaine de l'astronomie. Ce schéma d'ontologie regroupe un ensemble d'objets d'après des classes prédéfinies et des attributs binaires. Nous verrons dans la section 5.3 comment formaliser ce schéma d'ontologie. Le treillis complet du contexte $\mathbb{K}_o = (G_o, M_o, I_o)$ est présenté dans l'annexe B à la figure B.1.

TAB. 5.3 – Le contexte d'apposition $\mathbb{K} = (G, M, I)$

	Attributs binaires							Classes SIMBAD					
	isObserved	isIncluding	isEmitting	isEclipsing	isExpanding	isIncluded	isFlaring	Quasar	Galaxy	Asso._of_Stars	T_Tau_type_Star	Eclipsing_Binary	Star
Cen_A	×	×							×				
3C_273	×	×	×					×	×				
TWA	×	×			×					×			
Per_OB2	×	×								×			
T_Tauri	×		×			×	×				×		×
Y_Cygni	×		×	×		×						×	×
V773_Tau	×		×			×	×				×		×
Algol	×		×	×		×						×	×

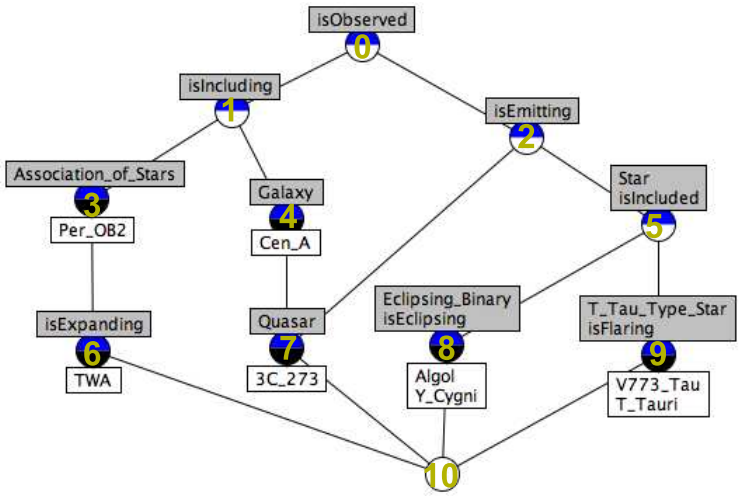


FIG. 5.3 – Le treillis de l'apposition de contextes $\mathbb{K}_o = (G_o, M_o, I_o)$

5.2 Construction du schéma d'ontologie dans le domaine de la microbiologie avec Pactole

Nous appliquons la même méthode de fouille pour regrouper les objets d'après leurs attributs communs et leurs relations avec d'autres objets. La construction des deux treillis des liens hiérarchiques et des attributs binaires, ainsi que celle du contexte d'apposition ne sont pas très détaillées, car le traitement est le même que dans le cas d'application au domaine de l'astronomie.

5.2.1 Construction d'un treillis à partir des classes d'objets

La transformation de cette hiérarchie en un treillis de concepts est faite par la construction du contexte $\mathbb{K}_1 := (G, M_1, I_1)$ défini comme suit :

- G est l'ensemble des objets du domaine (l'ensemble des bactéries),
- M_1 est l'ensemble des classes des objets extraites de NCBI
- I_1 est la relation qui assigne à chaque objet sa classe et toutes les super-classes de cette classe dans la hiérarchie.

	Classes NCBI			
	Proteobacteria	GammaProteobacteria	Actinobacteria	Bacilli
Helicobacter_Pylori	×			
Klebsiella_Pneumoniae	×	×		
Mycobacterium_Smegmatis			×	
Streptococcus_Pneumoniae				×
Klebsiella_Oxytoca	×	×		

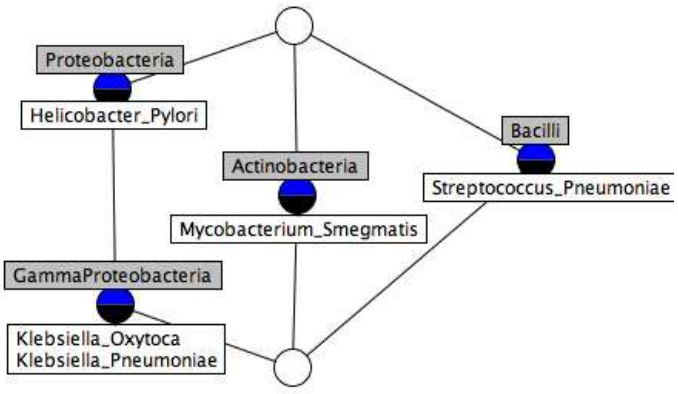


FIG. 5.4 – Contexte $\mathbb{K}_1 := (G, M_1, I_1)$ des bactéries avec leurs classes extraites de la base NCBI et son treillis de concepts associé.

La figure 5.4 présente un exemple du contexte $\mathbb{K}_1$ et son treillis de concepts correspondant. Dans ce treillis de concepts, deux bactéries sont dans le même concept si et seulement si elles appartiennent aux mêmes classes dans la NCBI. Par exemple, les deux bactéries {`Klebsiella_Pneumoniae`, `Klebsiella_Oxytoca`} sont regroupées par les classes {`GammaProteobacteria`, `Proteobacteria`} dans le même concept.

5.2.2 Construction d'un treillis à partir des attributs binaires

Nous construisons le contexte formel $\mathbb{K}_2 := (G, M_2, I_2)$ tel que :

- G est l'ensemble des objets du domaine (l'ensemble des bactéries, le même que dans $\mathbb{K}_1$),
- M_2 l'ensemble des attributs extraits des bases de données,
- $I_2(g, m)$ veut dire que l'objet g possède l'attribut m_2.

La figure 5.5 présente un exemple du contexte $\mathbb{K}_2$ et son treillis de concepts correspondant. Dans ce treillis de concepts, deux bactéries sont dans le même concept si et seulement si elles possèdent les mêmes attributs binaires. Par exemple, les deux bactéries {`Klebsiella_Pneumoniae`, `Klebsiella_Oxytoca`} sont regroupées par les attributs binaires {`isSticks`, `hasNegativeGram`, `isAnaerobic`} dans le même concept. Le treillis complet des bactéries est donné dans l'annexe B à la figure B.2.

5.2.3 Affectation d'attributs binaires à des classes d'objets

L'apposition est appliquée de la même façon que dans le domaine de l'astronomie, les deux contextes $\mathbb{K}_1 = (G, M_1, I_1)$ et $\mathbb{K}_2 = (G, M_2, I_2)$ sont fusionnés dans le contexte d'apposition $\mathbb{K}_B = (G, M, I)$

	Attributs binaires					
	isSpherical	isSticks	hasNegativeGram	hasPositiveGram	isAerobic	isAnaerobic
Helicobacter_Pylori	×		×		×	
Klebsiella_Pneumoniae		×	×			×
Streptococcus_Pneumoniae		×		×	×	
Mycobacterium_Smegmatis		×		×	×	
Klebsiella_Oxytoca		×	×			×

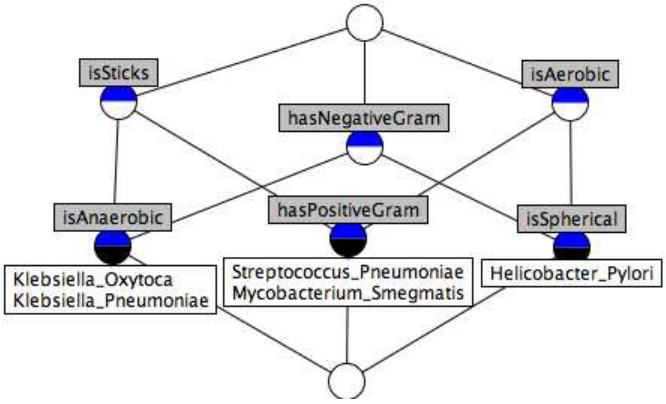


FIG. 5.5 – Contexte $\mathbb{K}_2 = (G, M_2, I_2)$ des bactéries avec leurs attributs binaires et son treillis de concepts correspondant

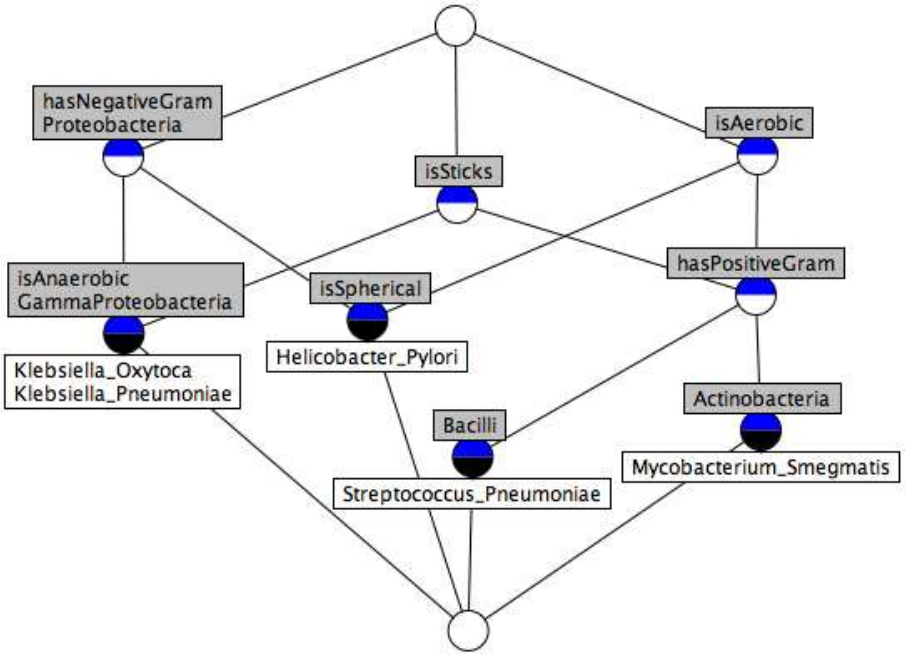


FIG. 5.6 – Le treillis de concepts résultant du contexte l'apposition des contextes $\mathbb{K}_1$ and $\mathbb{K}_2$

Dans le treillis $\underline{\mathfrak{B}}(\mathbb{K}_B)$ (figure 5.6), les classes se voient associer des attributs binaires. Par exemple, le concept B({`GammaProteobacteria`, `Proteobacteria`, `isSticks`, `hasNegativeGram`, `isAnaerobic`}, {`Klebsiella_Pneumoniae`, `Klebsiella_Oxytoca`}) permet d'associer à la classe `GammaProteobacteria` (sous-classe de la classe `Proteobacteria` dans la hiérarchie de la base NCBI) les attributs binaires {`isSticks`, `hasNegativeGram`, `isAnaerobic`}. Les experts peuvent interpréter ces éléments de deux façons différentes. La première est de considérer que les attributs binaires {`isSticks`, `hasNegativeGram`, `isAnaerobic`} définissent la classe `GammaProteobacteria`. La deuxième est de considérer que les attributs binaires {`isSticks`, `hasNegativeGram`, `isAnaerobic`} définissent une sous-classe de la classe `GammaProteobacteria` et ainsi créent une nouvelle classe dans la hiérarchie source. L'apposition de contextes permet

donc non seulement d'affecter des attributs binaires aux classes prédéfinies de la hiérarchie source, mais aussi d'enrichir cette hiérarchie source avec de nouvelles classes.

5.2.4 Construction du treillis relationnel

Le troisième Descripteur d'Objets (*DO3*) est composé des relations entre les objets. Ces relations ne peuvent être prises en compte en utilisant l'AFC traditionnelle, mais son extension l'Analyse Relationnelle de Concepts (ARC) qui a été introduite par [Rouane-Hacene *et al.*, 2007]. Avec ARC (Présentée dans la section 3.3.5) un objet n'est pas seulement décrit par des attributs binaires, mais aussi par les relations qu'il entretient avec d'autres objets (attributs relationnels). Ainsi, nous pouvons définir une classe de bactéries non seulement par l'ensemble des attributs qu'elles partagent, mais aussi par leur résistance aux antibiotiques.

Les données en ARC sont organisées dans une Famille de Contextes Relationnels (FCR) composé d'un ensemble de contextes $\mathbb{K}_i = (G_i, M_i, I_i)$ et d'un ensemble de relations $r_k \subseteq G_i \times G_j$.

Nous considérons les deux relations `isPartOfGenomeOf` entre les gènes et les bactéries et la relation `isResisting` entre les bactéries et les antibiotiques. Ainsi, la FCR $(\mathbf{K}, \mathbf{R})$ est composée de trois contextes et de deux relations :

- $\mathbb{K}_A = (G_A, M_A, I_A)$ le contexte des antibiotiques,
- $\mathbb{K}_B = (G_B, M_B, I_B)$ le contexte des bactéries,
- $\mathbb{K}_G = (G_G, M_G, I_G)$ le contexte des gènes.
- $r_1 \subseteq G_G \times G_B$ la relation `isPartOfGenomeOf` entre l'ensemble des gènes et l'ensemble des bactéries,
- $r_2 \subseteq G_B \times G_A$ la relation `isResisting` entre l'ensemble des bactéries et l'ensemble des antibiotiques.

La première étape consiste à construire les trois contextes formels $\mathbb{K}_A$, $\mathbb{K}_B$, $\mathbb{K}_G$. Nous avons déjà le contexte des bactéries $\mathbb{K}_B$ et son treillis correspondant dans la figure 5.6, reste à construire les contextes et les treillis des gènes et des antibiotiques.

Construction du treillis des gènes

Le contexte formel $\mathbb{K}_G = (G_G, M_G, I_G)$ se compose d'un ensemble de gènes G_G (ces gènes sont extraits du corpus de textes à l'aide de dictionnaire de nomenclatures), un ensemble d'attributs binaires M_G (extraits de bases de données) et une relation binaire $I_G \subseteq G_G \times M_G$ où $I_G(g_G, m_G)$ veut dire que le gène g_G possède l'attribut m_G.

La figure 5.7 présente un exemple du contexte $\mathbb{K}_G$ et son treillis de concepts correspondant. Dans ce treillis de concepts, deux gènes sont dans le même concept si et seulement si ils possèdent les mêmes attributs binaires. Par exemple, les deux gènes {`pbp`, `EmbB`} sont regroupés par les attributs binaires {`isProteinBinding`, `isInIntracellularRegion`} dans le même concept. Nous donnons le treillis complet des gènes l'annexe B à la figure B.4.

Construction du contexte des antibiotiques

De même manière, le contexte formel $\mathbb{K}_A = (G_A, M_A, I_A)$ se compose d'un ensemble des antibiotiques G_A (cet ensemble est aussi extrait des textes), un ensemble d'attributs binaires M_A et d'une relation binaire $I_A \subseteq G_A \times M_A$ où $I_A(g_A, m_A)$ veut dire que l'objet g_A possède l'attribut m_A. Les attributs M_A, comme présenté précédemment, sont extraits de la classification des ligands. Ils calculent : le nombre d'`accepteurs de liaisons hydrogène` (en anglais `Hydrogen Bond Acceptors (HBA)`), le `nombre de liaisons aromatiques` (en anglais `number of ARomatic bonds (ARB)`) et le `nombre de fractions de liaisons capables de rotation`

Contexte des gènes	isProteinBinding	isDrugBinging	isNucleicAcidBinding	isInIntraCellularRegion	isInExtraCellularRegion
23S_rRNA			×	×	
grlA					×
pbp	×	×		×	
EmbB	×			×	

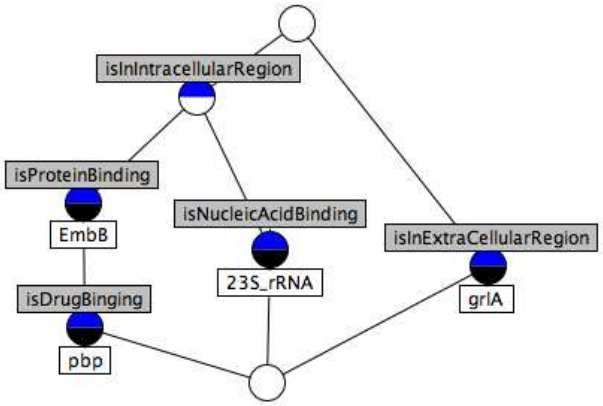


FIG. 5.7 – Le contexte des gènes $\mathbb{K}_G = (G_G, M_G, I_G)$ et son treillis de concepts correspondant

(en anglais `Fraction of Rotatable Bonds (FRB)`). Ainsi, l'attribut `FRB1` veut dire que l'antibiotique possède une seule fraction de liaisons capables de rotation. Le treillis complet des antibiotiques est présenté dans l'annexe B à la figure B.3.

Contexte des antibiotiques	FRB1	FRB3	ARB1	ARB2	HBA5	HBA10
Clarithromycin		×				×
Ciprofloxacin	×			×	×	
Cefotaxim		×	×			×
Macrolid			×			×

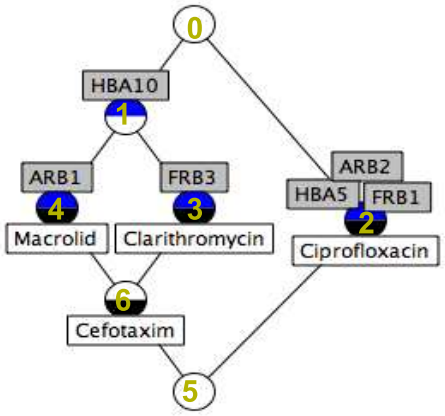


FIG. 5.8 – Le contexte des antibiotiques $\mathbb{K}_A = (G_A, M_A, I_A)$ et son treillis de concepts correspondant

La figure 5.8 présente un exemple du contexte $\mathbb{K}_A$ et son treillis de concepts correspondant. Dans ce treillis de concepts, deux antibiotiques sont dans le même concept si et seulement si ils possèdent les mêmes attributs binaires. Par exemple, les deux antibiotiques {`Macrolid`, `Cefotaxim`} sont regroupés par les attributs binaires {`ARB1`, `HBA10`} dans le même concept.

Tableaux des relations

La figure 5.9 présente les deux relations inter-contextes `isPartOfGenomeOf` et `isResisting`. La relation `isPartOfGenomeOf` (présentée à gauche de la figure) est la relation qui relie l'ensemble des gènes G_G (domaine de la relation) à l'ensemble des bactéries G_B (co-domaine de la relation). La relation `isResisting` (présentée à droite de la figure) est la relation qui relie l'ensemble des bactéries G_B (domaine de la relation) à l'ensemble des antibiotiques G_A (co-domaine de la relation).

La relation `isPartOfGenomeOf`					
	Helicobacter_Pylori	Klebsiella_Pneumoniae	Klebsiella_Oxytoca	Mycobacterium_Smegmatis	Streptococcus_Pneumoniae
23S_rRNA	×				
grlA		×	×		
pbp					×
EmbB				×	

La relation `isResisting`				
	Clarithromycin	Ciprofloxacin	Cefotaxim	Macrolid
Helicobacter_Pylori	×			
Klebsiella_Pneumoniae		×		
Klebsiella_Oxytoca		×		
Mycobacterium_Smegmatis				×
Streptococcus_Pneumoniae			×	

FIG. 5.9 – Les deux relations inter-contextes `isPartOfGenomeOf` (à gauche) et `isResisting` (à droite)

Échelonnage des contextes

L'application de l'ARC sur la famille de contextes relationnels $(\mathbf{K}, \mathbf{R})$ utilise l'échelonnage sur les deux ensembles qui sont domaines des deux relations `isPartOfGenomeOf` et `isResisting`. Nous rappelons que dans cette thèse, nous avons choisi d'utiliser l'échelonnage existentiel. Nous donnons ici des exemples de quantification universelle et existentielle :

`B1` := ∀ `isResisting:A1` et `B2` := ∃ `isResisting:A2`.

La première proposition est vraie si tous les objets de l'extension du concept `B1` ne résistent qu'aux instances du concept `A1`. La deuxième proposition est vraie si pour toute instance x de `B2`, il existe au moins une instance y de `A2` tel que y est en relation avec x.

	Clarithromycin	Ciprofloxacin	Cefotaxim	Macrolid
Helicobacter_Pylori	X			
Klebsiella_Pneumoniae		X		
Mycobacterium_Smegmatis				X
Streptococcus_Pneumoniae			X	
Klebsiella_Oxytoca		X		

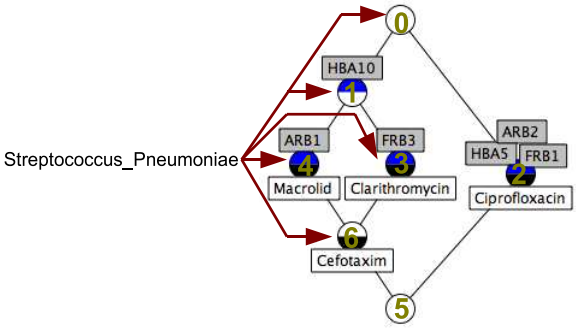


FIG. 5.10 – Exemple de l'application de l'échelonnage sur la bactérie `Streptococcus_Pneumoniae` en relation `isResisting` avec l'antibiotique `Cefotaxim`

Le premier treillis à être construit en ARC est le treillis des antibiotiques (comme il n'est domaine d'aucune relation, il ne changera pas). Puis, nous construisons le treillis des bactéries.

La figure 5.10 présente un exemple du traitement par graduation de la relation `isResisting` entre la bactérie `Streptococcus_Pneumoniae` et l'antibiotique `Cefotaxim`, ce que signifie que la bactérie `Streptococcus_Pneumoniae` possédera un attribut relationnel avec tous les concepts où l'antibiotique `Cefotaxim` apparaît en tant qu'instance. Ainsi, `Streptococcus_Pneumoniae` pos-

sède les attributs relationnels `isResisting:A0`, `isResisting:A1` `isResisting:A3` `isResisting:A4` et `isResisting:A6`. Le tableau 5.4 est le résultat de l'application de cette méthode de graduation à toutes les bactéries de l'ensemble G_B.

TAB. 5.4 – Contexte relationnel des bactéries

	Attributs binaires						Classes NCBI				Attributs relationnels					
	isSpherical	isSticks	hasNegativeGram	hasPositiveGram	isAerobic	isAnaerobic	Proteobacteria	GammaProteobacteria	Actinobacteria	Bacilli	isResisting:A0	isResisting:A1	isResisting:A3	isResisting:A2	isResisting:A4	isResisting:A6
Helicobacter_Pylori	×		×		×		×				×	×	×			
Klebsiella_Pneumoniae		×	×			×	×	×			×				×	
Mycobacterium_Smegmatis		×		×	×				×		×	×		×		
Streptococcus_Pneumoniae		×		×	×					×	×	×	×	×		×
Klebsiella_Oxytoca		×	×			×	×	×			×				×	

Le troisième treillis à être construit en ARC est le treillis des gènes. Le même principe de graduation est utilisé pour relier les gènes avec les concepts de bactéries. Par exemple, puisque le gène `grlA` est en relation `isPartOfGenomeOf` avec la bactérie `Klebsiella_pneumoniae`, alors ce gène possède un attribut relationnel avec tous les concepts dont `Klebsiella_Pneumoniae` est instance. Bien sûr les gènes sont reliés au treillis relationnel résultant de la graduation du contexte des bactéries. Ainsi, le gène `grlA` possède les attributs relationnels `isPartOfGenomeOf:B5`, `isPartOfGenomeOf:B1`, `isPartOfGenomeOf:B3` et `isPartOfGenomeOf:B0`. Le tableau 5.5 est le résultat de l'application de cette méthode de graduation à tous les gènes de l'ensemble G_G.

TAB. 5.5 – Le contexte relationnel des gènes

	Attributs binaires					Attributs relationnels									
	isProteinBinding	isDrugBinging	isNucleicAcidBinding	isInIntracellularRegion	isInExtraCellularRegion	isPartOfGenomeOf:B0	isPartOfGenomeOf:B2	isPartOfGenomeOf:B3	isPartOfGenomeOf:B1	isPartOfGenomeOf:B6	isPartOfGenomeOf:B4	isPartOfGenomeOf:B8	isPartOfGenomeOf:B9	isPartOfGenomeOf:B5	isPartOfGenomeOf:B10
23S_rRNA			×	×		×	×		×		×				×
grlA					×	×		×	×					×	
pbp	×	×		×		×	×	×		×			×		×
EmbB	×			×		×	×	×		×		×			

Treillis relationnels

Les figures 5.11 et 5.12 présentent les treillis relationnels résultant de la famille de contextes formels $(\mathbf{K}, \mathbf{R})$, ces trois treillis sont connectés entre eux (le treillis de la figure 5.11 est présenté seul par manque de place). Dans ces treillis, les objets ne sont pas seulement regroupés par des attributs binaires mais aussi par des attributs relationnels. Par exemple, les deux gènes {`EmbB`,`pbp`} (concept `G6` dans la figure 5.11) possèdent en commun les attributs binaires {`isProteinBinding, isInIntracellularRegion`} mais aussi la relation `isPartOfGenomeOf` avec les concepts des bactéries `B0, B3, B6`. Le concept des bactéries `B6` regroupe les bactéries {`Streptococcus_Pneumoniae`, `Mycobacterium_Smegmatis`} avec les attributs binaires {`hasPositiveGram, isSticks, isAerobic`}, mais aussi la relation `isResisting` avec le concept des antibiotiques `A4`. Cette méthode a aussi été appliquée et détaillée dans [Bendaoud *et al.*, 2007b; 2008a].

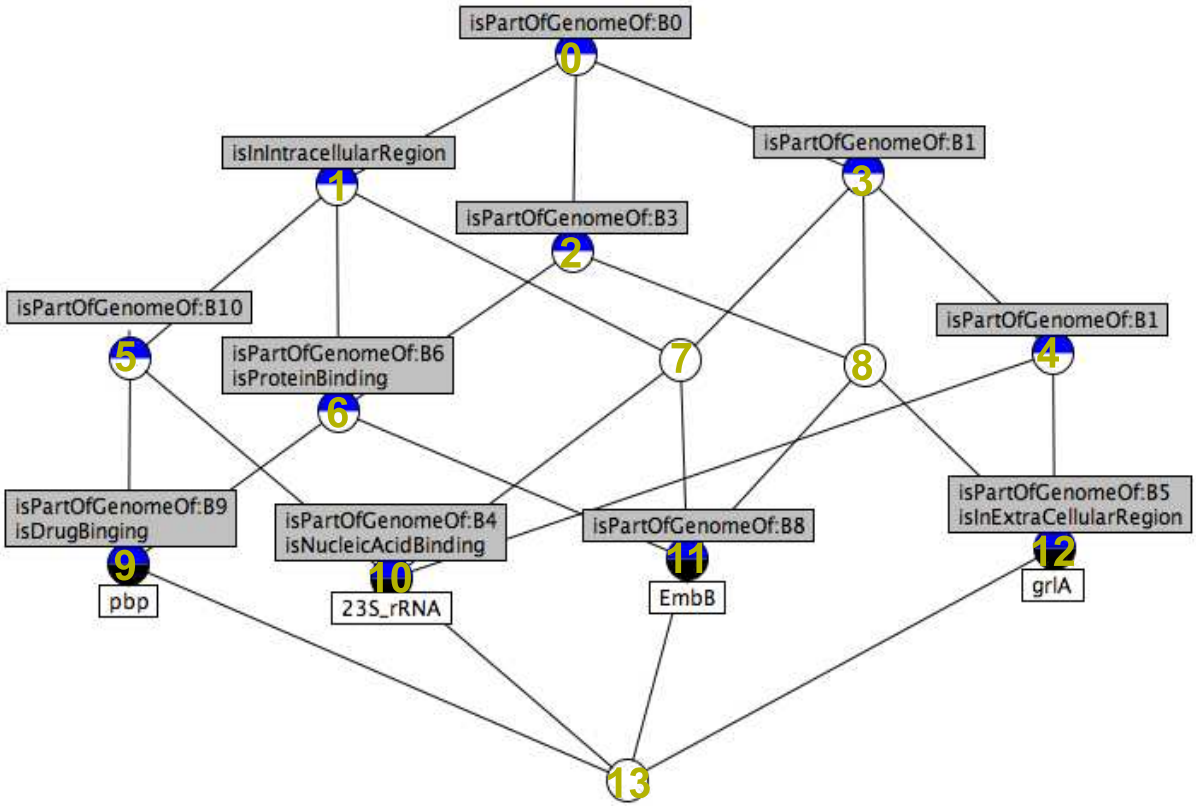


FIG. 5.11 – Treillis relationnel des gènes résultant de l'application de l'ARC sur la famille de contextes relationnels $(\mathbf{K}, \mathbf{R})$

5.2.5 Extraction d'unités de connaissances en microbiologie

Proposition de nouvelles classes

L'exemple de la figure 5.13 présente l'interaction entre une classe de gènes ayant la relation `isPartOfGenomeOf` avec une classe de bactéries qui a une relation `isResisting` avec une classe d'antibiotiques. Les gènes `{gyrA, parC}` ne possèdent aucun attribut en commun, mais les experts du domaine ont trouvé ce concept très intéressant, car les deux gènes sont fortement liés :

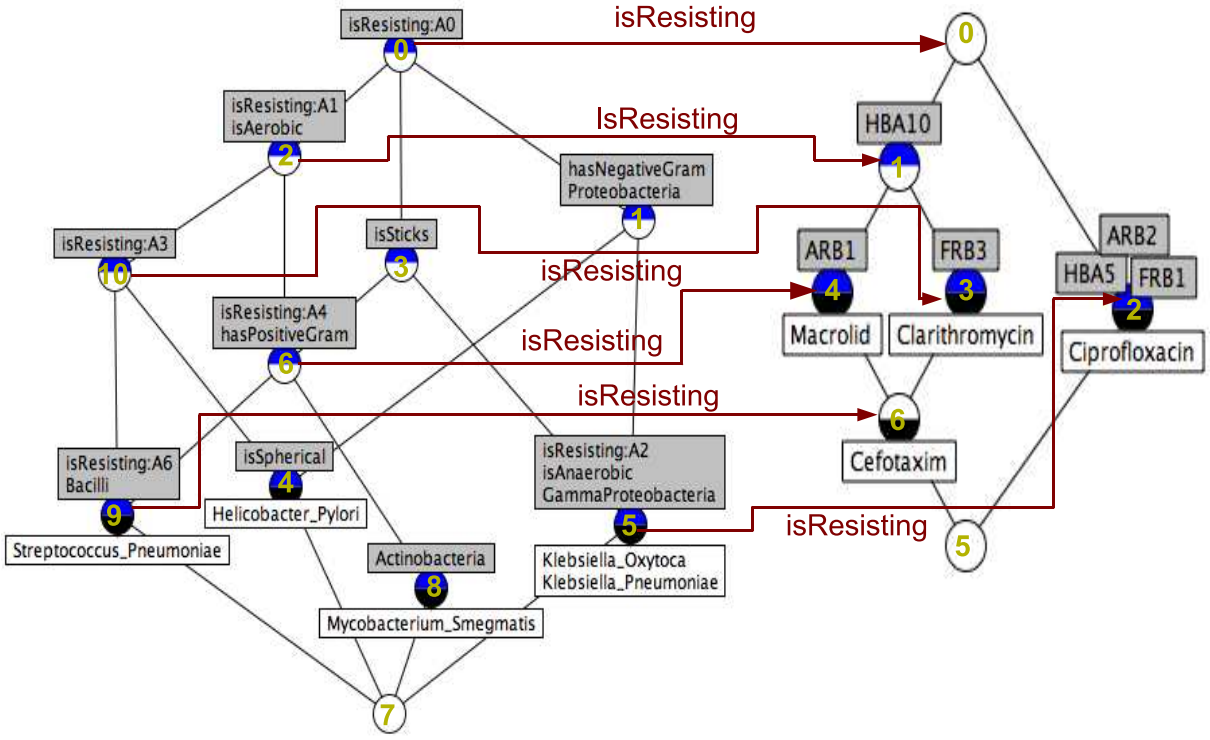


FIG. 5.12 – Treillis relationnels des bactéries (à gauche) et des antibiotiques (à droite) résultant de l'application de ARC sur la famille de contextes relationnels $(\mathbf{K}, \mathbf{R})$

l'un ne peut être trouvé sans l'autre. Cette classe n'a pas encore été « étiquetée » par les experts, mais elle pourrait être une nouvelle classe dans le thésaurus des gènes.

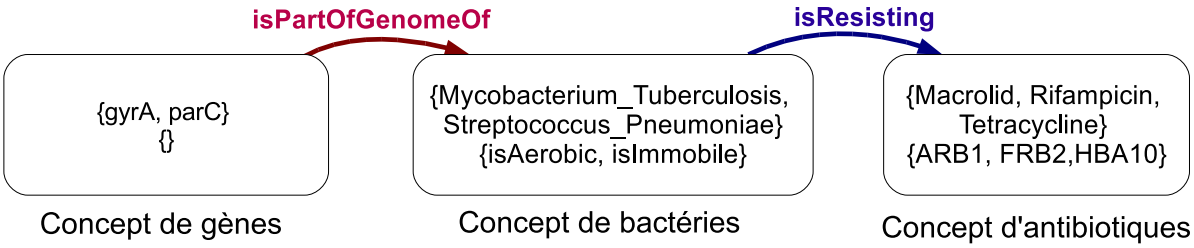


FIG. 5.13 – Exemple de mise en évidence de nouvelles classes

Explication du phénomène de résistance

Dans la figure 5.14, les experts ont trouvé une explication au phénomène de résistance de la classe des bactéries `{Mycobacterium_Smegmatis, Mycobacterium_Tuberculosis, Neisseria_Gonorrhoeae}` à l'ensemble des antibiotiques `{Macrolide, Rifampicin, Tetracycline}`. L'explication est : l'ensemble des antibiotiques `{Macrolide, Rifampicin, Tetracycline}` tue les bactéries en détruisant leur ADN. Mais, le fait que l'ensemble des

gènes {gyrA, gyrB, inhA, parE, rrs} possède l'attribut isBindingDNA (s'attacher à l'ADN) permet à l'ensemble des bactéries de résister à l'ensemble des antibiotiques.

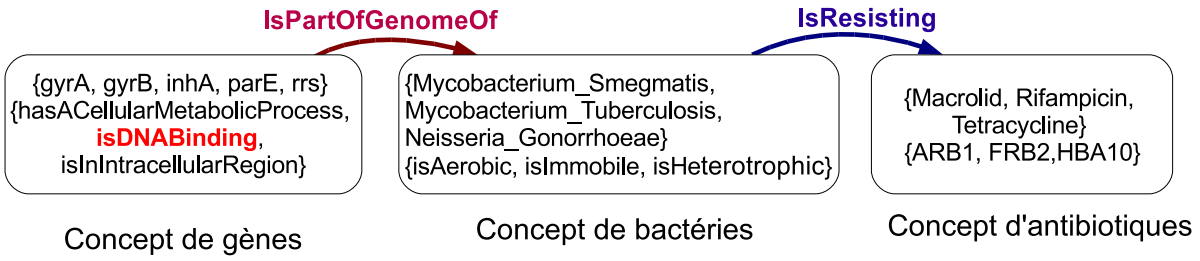


FIG. 5.14 – Exemple d'une interprétation du phénomène de résistance entre trois concepts : gènes-bacteries-antibiotiques

Les treillis des figures 5.11 et 5.12 constituent le schéma de l'ontologie dans le domaine de la microbiologie. Dans ce schéma d'ontologie, les concepts sont hiérarchisés et reliés entre eux par des relations binaires. Néanmoins, ils ne sont pas formellement définis. Pour pouvoir effectuer des raisonnements tels que l'instanciation automatique des concepts, la comparaison entre concepts ou encore la détection du domaine ou du co-domaine d'une relation ; il faut que ce schéma d'ontologie soit représenté en un langage formel.

5.3 Passage du schéma d'ontologie à une ontologie formelle

5.3.1 La représentation des concepts formels en logique de descriptions $\mathcal{FLE}$

Pour représenter les concepts formels de l'ARC, nous devons choisir un langage de représentation des connaissances. Ici nous avons fait le choix de la LD $\mathcal{FLE}$ comme suggéré dans les travaux de [Rouane-Hacene *et al.*, 2007]. Le langage des LD $\mathcal{FLE}$ [Baader *et al.*, 2003] inclut les constructeurs $\top$ (top), $\bot$ (bottom), $C \sqcap D$ (conjonction de concepts), $\forall r.C$ et $\exists r.C$ (quantificateurs universel et existentiel). Cet ensemble de constructeurs constitue l'ensemble minimal permettant de représenter les concepts formels du treillis en particulier : les objets, les attributs binaires, les attributs relationnels, les concepts primitifs, les concepts définis et la relation de subsomption entre concepts.

La transformation des treillis de l'ARC en une base de connaissances en $\mathcal{FLE}$ se schématise par une transformation appelée τ. Cette représentation en $\mathcal{FLE}$ permet d'introduire des concepts primitifs et des concepts définis, afin d'appliquer les mécanismes de raisonnement en LD et de pouvoir répondre automatiquement à un certain type de questions.

$\tau : \underline{\mathfrak{B}}(G_f, M_f, I_f) \longrightarrow TBox \cup ABox$, où $\underline{\mathfrak{B}}(G_f, M_f, I_f)$ sont les treillis de l'ARC (figure **??**). La TBOX et l'ABOX sont les composants en LD de l'ontologie finale. Le détail de τ est donné avec des exemples pour chaque type de transformation :

- un attribut $m_1 \in M_1$, où $\underline{\mathfrak{B}}(G, M_1, I_1)$ dénote que le treillis associé à la hiérarchie source est transformé en un concept atomique dans la TBOX. L'application de cette transformation sur le domaine de l'astronomie donne par exemple τ(`Quasar`) = `Quasar`, où `Quasar` est un concept primitif. L'application de cette transformation sur le domaine de la microbiologie donne par exemple τ(`Proteobacteria`) = `Proteobacteria`, où `Proteobacteria` est un concept primitif,

- un attribut du contexte $\underline{\mathfrak{B}}(G, M_2, I_2)$ qui dénote le contexte des objets et de leurs attributs binaires est transformé en une expression conceptuelle : $\tau(\mathtt{m_2}) = \exists \mathtt{m_2}.\top$. L'application de cette transformation sur le domaine de l'astronomie donne par exemple $\tau(\mathtt{isEmitting}) = \exists \mathtt{isEmitting}.\top$. L'application de cette transformation sur le domaine de la microbiologie donne par exemple $\tau(\mathtt{hasNegativeGram}) = \exists \mathtt{hasNegativeGram}.\top$,
- une relation $r \in R$ est transformée dans la TBox en un rôle atomique $\tau(r)$. Cette transformation n'est pas appliquée dans le domaine de l'astronomie, car il n'y a pas de relation entre les objets. Dans le domaine de la microbiologie, cette transformation donne par exemple : $\tau(\mathtt{isResisting}) = \mathtt{isResisting}$,
- un attribut relationnel r.A_1 est transformé dans la TBox en un concept défini $\tau(r.T_1)$. Cette transformation n'est pas appliquée dans le domaine de l'astronomie, car il n'y a pas de relation entre les objets. Dans le domaine de la microbiologie, cette transformation donne par exemple : $\tau(\mathtt{isResisting:A1}) = \mathtt{isResisting:A1}$
- un concept formel $C = (X, Y)$ est transformé en un concept défini par une conjonction de concepts primitifs et de quantificateurs existentiels de relations. L'application de cette transformation dans le domaine de l'astronomie donne par exemple $\tau(C_9) := \mathtt{Star} \sqcap \mathtt{T\_Tau\_Star} \sqcap \exists \mathtt{isObserved}.\top \sqcap \exists \mathtt{isEmitting}.\top \sqcap \exists \mathtt{isFlaring}.\top$ (figure 5.3). L'application de cette transformation dans le domaine de la microbiologie donne par exemple $\tau(B6) := \exists \mathtt{hasPositiveGram}.\top \sqcap \exists \mathtt{isSticks}.\top \sqcap \exists \mathtt{isAerobic}.\top \sqcap \exists\ \mathtt{isResisting:A0} \sqcap \exists\ \mathtt{isResisting:A1} \sqcap \exists\ \mathtt{isResisting:A4}$.
- la relation de subsomption entre concepts (du treillis) est transformée en une inclusion générale de concepts. L'application de cette transformation dans le domaine de l'astronomie donne, par exemple, la relation de subsomption $C_6 \sqsubseteq C_3$ dans le treillis devient $\tau(C_6) \sqsubseteq \tau(C_3)$ dans l'ontologie. L'application de cette transformation dans le domaine de la microbiologie donne, par exemple, la relation de subsomption $C_6 \sqsubseteq C_2$ dans le treillis qui devient $\tau(C_6) \sqsubseteq \tau(C_2)$ dans l'ontologie,
- un objet $g \in G$ est transformé en un individu $\tau(g)$ dans la ABox. L'application de cette transformation dans le domaine de l'astronomie donne par exemple l'objet céleste `3C_273` qui devient l'instance $\tau(\mathtt{3C\_273})$. L'application de cette transformation sur le domaine de la microbiologie donne par exemple la bactérie `Helicobacter_Pylori` qui devient l'instance $\tau(\mathtt{Helicobacter\_Pylori})$.

Le passage des treillis résultant de l'AFC/ARC à une ontologie formelle n'est pas effectué à la main, mais à l'aide de l'API Java « Jena » (présentée dans la sous-section 2.3.4). Nous donnons ici quelques exemples de concepts définis extraits de la représentation du treillis des objets célestes (figure 5.3) notés $\mathtt{C}_i$ dans le domaine de l'astronomie. Nous présentons des exemples, dans le domaine de la microbiologie, de concepts définis extraits de la représentation des treillis des bactéries notés $\mathtt{B}_i$, des antibiotiques notés $\mathtt{A}_i$ et des gènes notés $\mathtt{G}_i$ résultant de l'ARC (figures 5.11 et 5.12) :

$\mathtt{C}_2 = \exists \mathtt{isObserved}.\top \sqcap \exists \mathtt{isEmitting}.\top$

$\mathtt{C}_4 = \mathtt{Galaxy} \sqcap \exists \mathtt{isObserved}.\top \sqcap \exists \mathtt{isIncluding}.\top$

$\mathtt{G}_6 = \exists \mathtt{isProteinBinding}.\top \sqcap \exists \mathtt{isInIntracellularRegion}.\top \sqcap \exists\ \mathtt{isPartOfGenomeOf:B0} \sqcap \exists\ \mathtt{isPartOfGenomeOf:B3} \sqcap \exists\ \mathtt{isPartOfGenomeOf:B6}$

$\mathtt{B6} = \exists \mathtt{hasPositiveGram}.\top \sqcap \exists \mathtt{isSticks}.\top \sqcap \exists \mathtt{isAerobic}.\top \sqcap \exists \mathtt{isResisting:A0} \sqcap \exists \mathtt{isResisting:A1} \sqcap \exists \mathtt{isResisting:A4}$

$\mathtt{A}_4 = \exists \mathtt{ARB1}.\top \sqcap \exists \mathtt{HBA10}.\top$

5.3.2 Implémentation de la représentation des concepts formels en OWL

Après la représentation de chaque élément de l'ontologie dans le langage formel $\mathcal{FLE}$, il faut l'implémenter en un langage du Web sémantique pour pouvoir éditer l'ontologie résultante avec un éditeur tel que le logiciel PROTÉGÉ et enfin, effectuer des raisonnements, avec des moteurs d'inférences tels que RACER ou PELLET (présentés dans la sous-section 2.3.4). Ces moteurs d'inférences permettent de répondre automatiquement à des questions complexes des experts. L'implémentation de la représentation des concepts formels est faite avec le langage du Web sémantique OWL. Cette implémentation est définie dans le tableau 2.1. La figure 5.15 présente un exemple de cette implémentation dans le domaine de l'astronomie. La première partie définit `Eclipsing_Binary` en tant que concept primitif (`owl:Class`) et sous-concept (`rdfs:subClassOf`) du concept `Star`. La deuxième partie définit le concept `C2` en tant que concept (`owl:Class`) défini (`owl:equivalentClass`) par une restriction (`owl:Restriction`) existentielle (`owl:someValuesFrom`) de l'attribut binaire `isEmitting` et sous-concept (`rdfs:subClassOf`) du concept `C0`. La troisième partie définit l'attribut (attribut binaire) `isEmitting` (`owl:ObjectProperty`) avec le domaine `C2` (`rdfs:domain rdfs:resource="♯C2"`) et un co-domaine `Thing` (le top $\top$) (`rdfs:range rdfs:resource="&owl;Thing"`). Enfin, la quatrième partie définit l'objet `Algol` en tant qu'instance du concept `C8`. Un autre exemple d'un concept de microbiologie est présenté dans l'annexe A à la figure A.1.

```
<owl:Class rdf:about="#Eclipsing_Binary">
   <rdfs:subClassOf rdf:resource="#Star"/>
</owl:Class>
```

```
<owl:Class rdf:about="#C2">
   <owl:equivalentClass>
      <owl:Restriction>
         <owl:onProperty rdf:resource="#isEmitting"/>
         <owl:someValuesFrom rdf:resource="&owl;Thing"/>
      </owl:Restriction>
   </owl:equivalentClass>
   <rdfs:subClassOf rdf:resource="#C0"/>
</owl:Class>
```

```
<owl:ObjectProperty rdf:about="#isEmitting">
   <rdfs:domain rdf:resource="#C2"/>
   <rdfs:range rdf:resource="&owl;Thing"/>
</owl:ObjectProperty>
```

```
<C8 rdf:about="#Algol"/>
```

FIG. 5.15 – Représentation des concepts `Eclipsing_Binary` et `C2`, de l'attribut binaire `isEmitting` ainsi que de l'objet `Algol` en OWL

Les figures 5.16 et 5.17 présentent l'édition des deux ontologies résultant de la représentation des treillis des objets célestes, des bactéries, des gènes et des antibiotiques dans le langage formel $\mathcal{FLE}$ et implémentée dans le langage du Web sémantique OWL. Les relations ne sont pas visibles, car le plugin de PROTÉGÉ « OWLViz » ne permet pas de visualiser les relations.

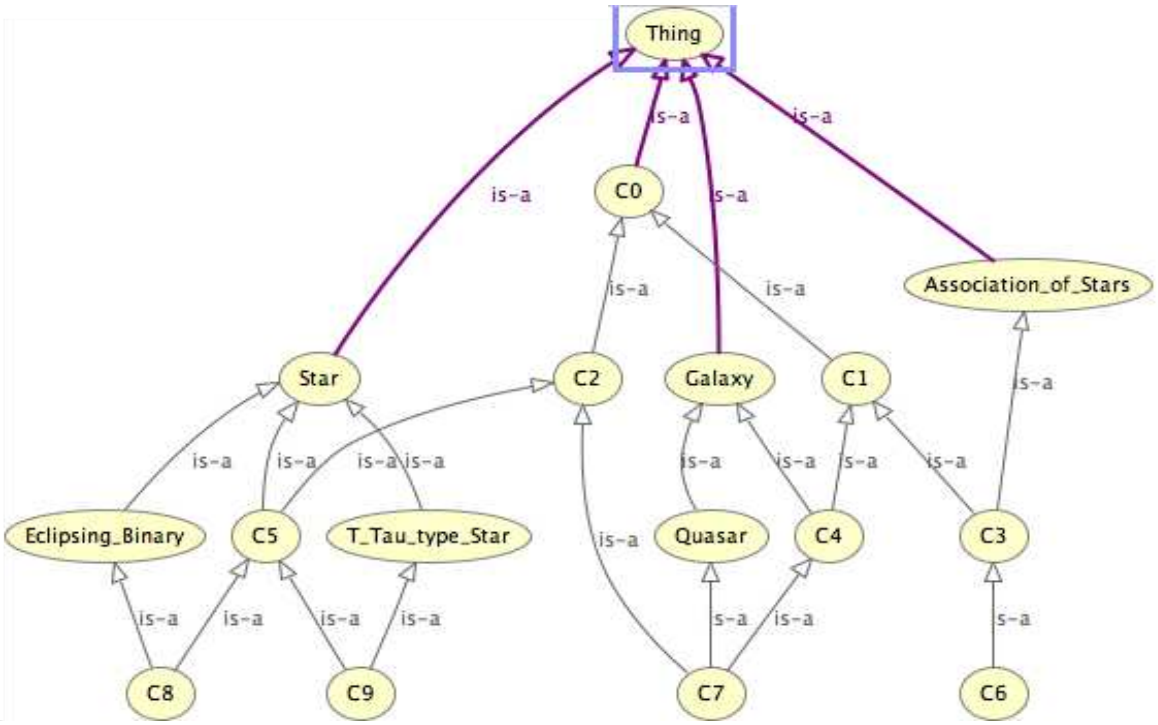


FIG. 5.16 – L'ontologie résultant de la représentation des treillis de l'AFC dans le domaine de l'astronomie. L'ontologie est éditée par le logiciel PROTÉGÉ.

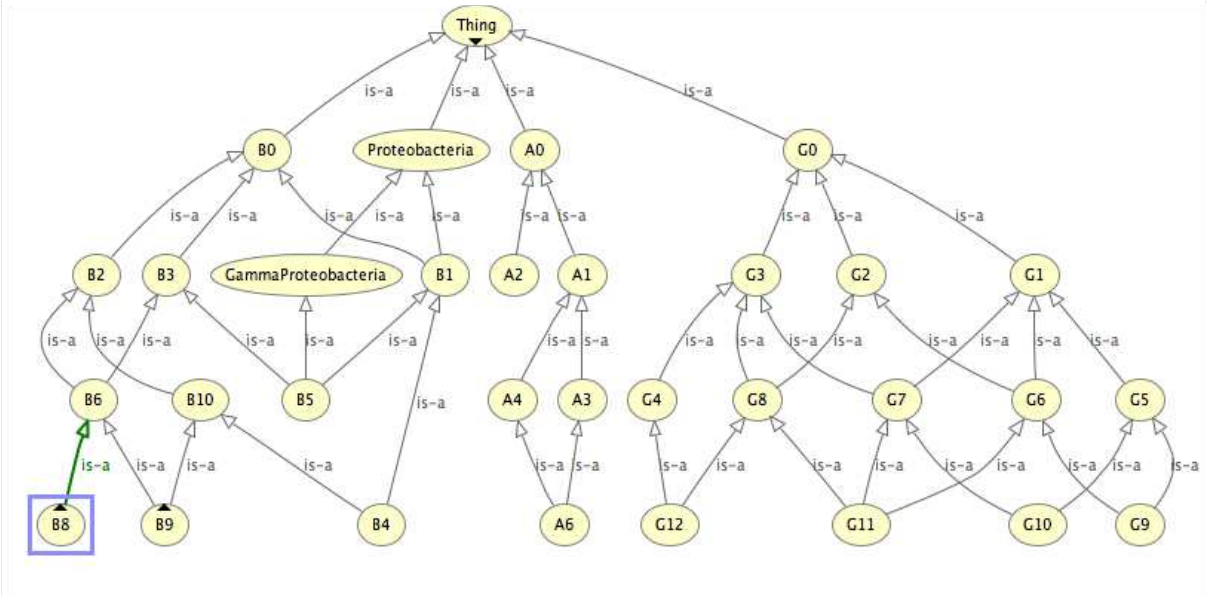


FIG. 5.17 – Une partie de l'ontologie résultant de la représentation des treillis de l'ARC dans le domaine de la microbiologie. L'ontologie est éditée par le logiciel PROTÉGÉ.

5.3.3 Raisonnement avec les concepts de l'ontologie

Les opérations de raisonnement associées à l'ontologie sont : l'instanciation, la subsomption de concepts, la comparaison de concepts et la détection du domaine d'une relation. Les détails de ces opérations de raisonnement sont donnés dans des exemples. Nous utilisons le raisonneur PELLET (voir la sous-section 2.3.4) pour effectuer ces raisonnements et le langage d'interrogation SPARQL pour interroger l'ontologie. Ce langage est intégré dans l'outil KOWL (présenté dans la sous-section 2.3.4). Les détails de ces opérations de raisonnement sont donnés dans le contexte des deux domaines précédents : l'astronomie et la microbiologie.

Instanciation de concepts.

L'instanciation de concepts consiste à trouver le concept d'un objet dans la hiérarchie des concepts. Cette opération permet de répondre à des questions comme « *trouver le concept* $C(o_1)$ *de l'objet* o_1 *dans l'ontologie* ». La réponse à cette question utilise la *classification progressive* [Langley, 1996] qui consiste à partir du $\top$ dans la hiérarchie et à descendre progressivement dans la hiérarchie jusqu'à arriver au concept de l'objet o_1. Nous prenons par exemple l'instance o_1 ayant les attributs $\{p_1, p_2\}$ et appartenant à la classe C_3 dans la hiérarchie source (*i.e.* C_3 est un concept atomique dans l'ontologie). Le concept de o_1 est défini par $C(o_1) \sqsupseteq C_3 \sqcap \exists p_1.\top \sqcap \exists p_2.\top$:

1. Si $C(o_1) = C_3 \sqcap \exists p_1.\top \sqcap \exists p_2.\top$ existe, alors l'objet o_1 est instance du concept $C(o_1)$. Cet objet est alors hérité par tout les ancêtres du concepts $C(o_1)$,
2. Si $C(o_1) = C_3 \sqcap \exists p_1.\top \sqcap \exists p_2.\top$ n'existe pas, alors l'objet o_1 est instance des concepts C appartenant à l'ensemble des parties de l'ensemble $E = \{C3, \exists p_1.\top, \exists p_2.\top\}$ noté $P(E)$.

Instanciation de concepts dans le domaine de l'astronomie. L'application de cette opération sur un exemple d'astronomie consiste à répondre à des questions telles que « Quel est le concept de l'objet `Angel` qui possède les attributs `{isObserved,isExpanding}` et appartient à la classe `{Association_of_Stars}` dans la hiérarchie de la base SIMBAD ». La réponse est le concept $X \equiv$ `Association_of_Stars` $\sqcap \exists$ `isObserved`.$\top \sqcap \exists$ `isExpanding`.$\top$. Pour trouver ce concept, il faut répondre à quatre questions telles que `c` $\neq \bot$:

- Quels sont les concepts `c` subsumés par le concept `a` domaine du rôle `isObserved` ? Traduit en SPARQL par la recherche de deux triplets RDF :
  1. le triplet « `astro :isObserved rdfs :domaine ?a` » recherche les concepts `a` domaine du rôle `isObserved` dans l'ontologie `astro`. La réponse est que `a` est le concept `C0`,
  2. le triplet « `?c rdfs :subClassOf ?a` » qui recherche les concepts `c` subsumés par les concepts `a` résultant de la première requête. Les réponses sont que `c` $\in$ {`C0,C1,C2,C3,C4,C5,C6,C7,C8,C9`}.
- Quels sont les concepts `c` subsumés par le concept `b` domaine du rôle `isExpanding` ? Traduit en SPARQL par la recherche de deux triplets RDF :
  1. le triplet « `astro :isExpanding rdfs :domaine ?b` » recherche les concepts `b` domaine du rôle `isExpanding` dans l'ontologie `astro`. La réponse est : `b` est le concept `C6`,
  2. le triplet « `?c rdfs :subClassOf ?b` » qui recherche les concepts `c` subsumés par les concepts `b` résultant de la première requête. Deux réponses données sont `C6 et` $\bot$ et comme `c` $\neq \bot$ alors la réponse est `C6`.
- Quels sont les concepts `c` subsumés par le concept `Association_of_Stars` ? Traduit en SPARQL par la recherche du triplet RDF :

Ontology OWL :

```
http://www.loria.fr/~bendaoud/Astro.owl
```

SPARQL query :

```
PREFIX astro:
<http://www.semanticweb.org/ontologies/2009/1/OntoAstro4.owl#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs:
<http://www.w3.org/2000/01/rdf-schema#>
SELECT ?a ?b ?c
WHERE
{
   astro:isObserved rdfs:domain ?a.
   ?c rdfs:subClassOf ?a.
   astro:isExpanding rdfs:domain ?b.
   ?c rdfs:subClassOf ?b.
   ?c rdfs:subClassOf
astro:Association_of_Stars.
}
```

Résultat de la requête

```
- <results>
  - <result>
    - <binding name="a">
      - <uri>
          http://www.semanticweb.org/ontologies/2009/1/OntoAstro4.owl#C0
        </uri>
      </binding>
    - <binding name="b">
      - <uri>
          http://www.semanticweb.org/ontologies/2009/1/OntoAstro4.owl#C6
        </uri>
      </binding>
    - <binding name="c">
      - <uri>
          http://www.semanticweb.org/ontologies/2009/1/OntoAstro4.owl#C6
        </uri>
      </binding>
  </result>
```

FIG. 5.18 – Requête SPARQL d'instanciation de l'objet `Angel` dans l'ontologie résultant du domaine de l'astronomie et une partie de la réponse à la requête

1. le triplet « `?c rdfs :subClassOf astro :Association_of_Stars` » recherche les concepts `c` subsumés par le concept `Association_of_Stars` dans l'ontologie `astro`. La réponse est : `c` est le concept `Association_of_Stars,C3,C6`,

– Et enfin le concept `c` est le concept vérifiant toutes les requêtes précédentes, ce qui revient à dire le concept `C6`. La réponse complète à cette requête est donnée dans l'annexe A à la figure A.2.

La figure 5.18 présente la requête d'instanciation de l'objet `Angel` dans l'ontologie présentée dans la figure 5.16. `Angel` est instance du concept `C6`. `C6` $\equiv$ `Association_of_Stars` $\sqcap$ $\exists$`isObserved.`$\top$ $\sqcap\exists$`isExpanding.`$\top$ $\neq \bot$.

Si le seul résultat de la requête est le $\bot$ alors on doit rechercher les concepts appartenant à l'ensemble des parties de l'ensemble `E` = {`Association_of_Stars,` $\exists$`isObserved.`$\top$, $\exists$`isExpanding.`$\top$} noté `P(E)` et poser les requêtes présentées dans la figure 5.19. Les réponses à cette requête sont tous les concepts vérifiant `?c, ?d et ?e`.

Cette opération d'instanciation nous permet automatiquement de peupler notre ontologie (ajouter des instances) sans avoir ni à ré-appliquer les méthodes de fouille, ni à re-calculer les

Ontology OWL : http://www.loria.fr/~bendaoud/Astro.owl

SPARQL query :

```
PREFIX astro:
<http://www.semanticweb.org/ontologies/2009/1/OntoAstro4.owl#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs:
<http://www.w3.org/2000/01/rdf-schema#>
SELECT ?a ?b ?c ?d ?e
WHERE
{
   astro:isObserved rdfs:domain ?a.
   ?a rdfs:subClassOf ?c.
   astro:isExpanding rdfs:domain ?b.
   ?b rdfs:subClassOf ?d.
   astro:Association_of_Stars rdfs:subClassOf ?e.
}
```

FIG. 5.19 – Requête SPARQL de recherche de tous les concepts de l'ensemble des parties `P(E)`

treillis, ni à représenter les éléments des treillis.

Ontology OWL : http://www.loria.fr/~bendaoud/MicroBio4.owl

SPARQL query :

```
PREFIX MicroBio:
<http://www.semanticweb.org/ontologies/2008/11/MicroBio1.owl#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs:  <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?a ?b ?c ?d
WHERE
{
    MicroBio:hasPositiveGram rdfs:domain ?a.
    ?d rdfs:subClassOf ?a.
    MicroBio:isAerobic rdfs:domain ?b.
    ?d rdfs:subClassOf ?b.
    MicroBio:isSticks rdfs:domain ?c.
    ?d rdfs:subClassOf ?c.
    ?d rdfs:subClassOf _:R.
    _:R  a owl:Restriction .
    _:R owl:onProperty MicroBio:isResisting.
    _:R owl:someValuesFrom MicroBio:A4.
}
```

FIG. 5.20 – Requête SPARQL d'instanciation de l'objet `Staphylococcus_Aureus` dans l'ontologie résultant du domaine de la microbiologie

Instanciation de concepts dans le domaine de la microbiologie. L'application de cette opération dans un exemple de microbiologie consiste à répondre à des questions telles que « *Quelle est la classe de l'objet* `Staphylococcus_Aureus` *qui possède les attributs* `{isSticks, isAerobic, hasPositiveGram}` *et la relation* `isResisting` *avec le concept d'antibiotiques* `A4` » La réponse est le concept $X \equiv \exists$ `isSticks`.$\top \sqcap \exists$ `isAerobic`.$\top \sqcap \exists$ `hasPositiveGram`.$\top \sqcap \exists$ `isResisting:A4`.

La requête SPARQL correspondant à cette question est donnée dans la figure 5.20. La réponse est le concept `B6` (voir A à la figure A.3).

Comparaison de concepts de l'astronomie. Dans les domaines scientifiques tels que l'astronomie ou la microbiologie, les hiérarchies sources sont généralement des arbres. Ce type de structure est insuffisant pour classifier des objets aussi complexes que les objets célestes ou les bactéries.

Par exemple, dans le domaine de l'astronomie, nous avons les deux objets `Loop_1` et `Honeycomb_nebula`. L'objet céleste `Loop_1` est classifié en tant que `SuperNova_Remnant_Candidate` dans la base SimBad (`SuperNova_Remnant_Candidate` est la classe des objets dont les experts supposent que ce sont des `SuperNova_Remnant`, mais ils n'en sont pas sûrs) et `Honeycomb_nebula` est classifié en tant que `SuperNova_Remnant` dans la base SimBad. Les deux objets ne se retrouvent qu'à la racine ($\top$) de SimBad alors qu'ils sont supposés être dans des classes presque identiques.

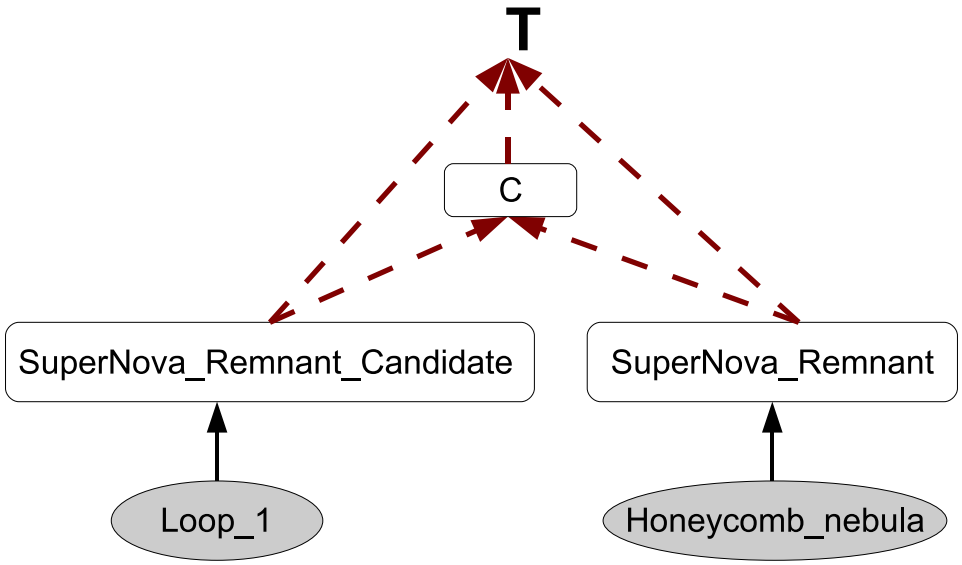


Fig. 5.21 – Une partie de l'arbre de la base SimBad

La figure 5.21 décrit l'arbre de la base SimBad. La deuxième opération de raisonnement consiste à rechercher un concept `C` dont deux objets sont des instances. Cette opération est aussi faite par classification progressive, car pour trouver le concept des deux instances o_1 et o_2, il faut trouver les concepts `C(`o_1`)` et `C(`o_2`)`, puis chercher le plus petit subsumant commun (`LCS`) de `C(`o_1`)` et `C(`o_2`)`. Pour trouver les concepts `C(`o_1`)` et `C(`o_2`)`, nous utilisons les requêtes d'instanciation. Pour l'exemple des deux objets `Loop_1` et `Honeycomb_nebula`, `Loop_1` est instancié dans le concept `C(Loop_1) = C367` et `Honeycomb_nebula` est instancié dans le concept `C(Honeycomb_nebula) = C368`.

Plusieurs cas sont possibles pour `LCS(CPS(`o_1`),CPS(`o_2`))` avec les conditions que `CPS(`o_1`)` $\neq \top$ et `CPS(`o_2`)` $\neq \top$:

1. Si `C(`o_1`)` $\sqsubseteq$ `C(`o_2`)` alors `LCS(C(`o_1`),C(`o_2`)) = C(`o_2`)`,
2. Si `C(`o_2`)` $\sqsubseteq$ `C(`o_1`)` alors `LCS(C(`o_1`),C(`o_2`)) = C(`o_1`)`,
3. Si `C(`o_1`)` $=$ `C(`o_2`)` alors `LCS(C(`o_1`),C(`o_2`)) = C(`o_1`) = C(`o_2`)`,
4. Si `C(`o_1`)` $\sqcup$ `C(`o_2`)` $\neq \top$ alors `LCS(C(`o_1`),C(`o_2`))` existe,
5. Si `C(`o_1`)` $\sqcup$ `C(`o_2`)` $= \top$ alors `LCS(C(`o_1`),C(`o_2`))` n'existe pas,

Dans les quatre premiers cas, les deux objets sont instances d'un même concept `LCS(C(`o_1`),C(`o_2`))` qui sera présenté aux experts, potentiellement comme une nouvelle classe d'objets. Dans le cinquième et dernier cas, il n'existe pas de concept dont les deux objets sont des instances.

Ontology OWL : http://www.loria.fr/~bendaoud/Astro.owl

```
PREFIX astro:
<http://www.semanticweb.org/ontologies/2009/1/OntoAstro4.owl#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs:
<http://www.w3.org/2000/01/rdf-schema#>
SELECT ?a
WHERE
{
   astro:C367 rdfs:subClassOf ?a.
   astro:C368 rdfs:subClassOf ?a.
}
```

FIG. 5.22 – Requête SPARQL recherchant le plus petit subsument des concepts C367 et C368

Nous appliquons cette méthode sur l'exemple des deux objets `Loop_1` et `Honeycomb_nebula`. La figure 5.22 présente la requête SPARQL pour trouver le `LCS(C(Loop_1),C(Honeycomb_nebula))`. `C367` $\sqcup$ `C368` $\neq \top$. Ainsi nous sommes dans le quatrième cas où `LCS(C(`o_1`),C(`o_2`))` existe et est égal à `C267` dans l'ontologie :
`C267` = `Stars` $\sqcap \exists$ `isObserved`.$\top \sqcap \exists$ `isIncluding`.$\top \sqcap \exists$ `isEmitting`.$\top$. Le concept `C267` est présenté aux experts comme une nouvelle classe potentielle d'objets.

Ontology OWL : http://www.loria.fr/~bendaoud/MicroBio4.owl

SPARQL query :

```
PREFIX MicroBio:
<http://www.semanticweb.org/ontologies/2008/11/MicroBio1.owl#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs:  <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?a ?b ?c ?d ?e ?y
WHERE
{
    MicroBio:HBA5 rdfs:domain ?a.
    ?d rdfs:subClassOf ?a.
   MicroBio:ARB2 rdfs:domain ?b.
    ?d rdfs:subClassOf ?b.
    MicroBio:FRB1 rdfs:domain ?c.
    ?d rdfs:subClassOf ?c.
    ?e rdfs:subClassOf _:R.
    _:R  a owl:Restriction .
    _:R owl:onProperty MicroBio:isResisting.
    _:R owl:someValuesFrom ?d.
  ?y rdf:type ?e
}
```

FIG. 5.23 – Requête SPARQL recherchant le domaine de la relation `isResisting` donc le co-domaine est le concept instanciant l'objet `Norfloxacin`

Détection du domaine d'une relation dans le domaine de la microbiologie. La troisième opération nous permet de détecter le domaine d'une relation. Nous prenons, par exemple, la question « *Quelles bactéries résistent au concept d'antibiotique dont l'objet* `Norfloxacin` *est instance ? L'objet* `Norfloxacin` *possède les attributs* `{ARB2,HBA5,FRB1}`. ».

Dans ce cas, la première étape consiste à instancier l'objet `Norfloxacin`, en utilisant la première opération d'instanciation. Le résultat est `A(Norfloxacin)` = `A2`. Ensuite, il faut trouver

les concepts des bactéries qui sont en relation `isResisting` avec le concept des antibiotiques `A2`. Pour cela, il faut rechercher les triplets RDF contenant la restriction qui relie le concept des antibiotiques `A2` à un autre concept `?d` par la quantification existentielle de la relation `isResisting`. Ceci se traduit en SPARQL par la recherche de trois triplets RDF :

- « `_ :R a owl :Restriction.` » Définition de `_ :R` en tant que restriction `a owl :Restriction.`,
- « `_ :R owl :onProperty MicroBio :isResisting.` » Définit `_ :R` comme étant la restriction de la relation `isResisting` de l'ontologie `MicroBio`,
- « `_ :R owl :someValuesFrom MicroBio :A4.` » Définit `_ :R` comme étant une quantification existentielle avec comme co-domaine le concept `A4` de l'ontologie `MicroBio`.

Une seule réponse est donnée, c'est le concept `B5`. Puis, il faut rechercher les instances du concept `B5` pour trouver les bactéries qui résistent à l'antibiotique `Norfloxacin`. Ce qui se traduit en SPARQL par la recherche du triplet RDF « `?y rdf :type ?e.` ». Les réponses sont les bactéries `Klebsiella_Oxytoca` et `Klebsiella_Pneumoniae` (voir l'annexe A, figures A.4 et A.5).

5.4 Travaux similaires utilisant l'AFC

Nous avons présenté dans la section 3.2, des travaux similaires proposant des méthodes d'extraction de connaissances à partir de ressources textuelles. Dans cette sextion, nous nous intéressons aux travaux de construction d'ontologies avec la méthode symbolique AFC. Cette classification regroupe un ensemble d'objets d'après les attributs binaires qu'ils partagent en s'appuyant sur la théorie de Galois. Haav [2004] a été la première à avoir eu l'idée d'utiliser l'AFC pour construire une hiérarchie de concepts. Elle recherche la cooccurrence des termes dans les textes avec comme contexte formel $\mathbb{K}=(\mathtt{G},\mathtt{M},\mathtt{I})$:

- G : ensemble d'objets, ici les objets sont les textes du corpus étudié,
- M : ensemble d'attributs binaires, ici l'ensemble d'attributs est composé par un ensemble de syntagmes nominals extraits des textes,
- $I(g,m)$ si et seulement si le terme (le syntagme nominal) m est présent dans le texte t.

Nous reprenons l'exemple de l'article de [Haav, 2004]. Cette méthode construit un contexte à partir d'un corpus de textes dans le domaine de l'immobilier (voir figure 5.24). Après la construction du treillis de concepts, cette méthode utilise les clauses de Horn [Cornuéjols A., 2003] pour représenter formellement le treillis résultant en une ontologie de domaine.

Objects	Attributes (noun-phrases)					
	Real estate	Family house	Country house	Summer house	Blockhouse	Skyscraper
A1	1	1				
A2	1	1	1	1		
A3	1		1			
A4	1				1	1
A5	1				1	
A6	1			1		

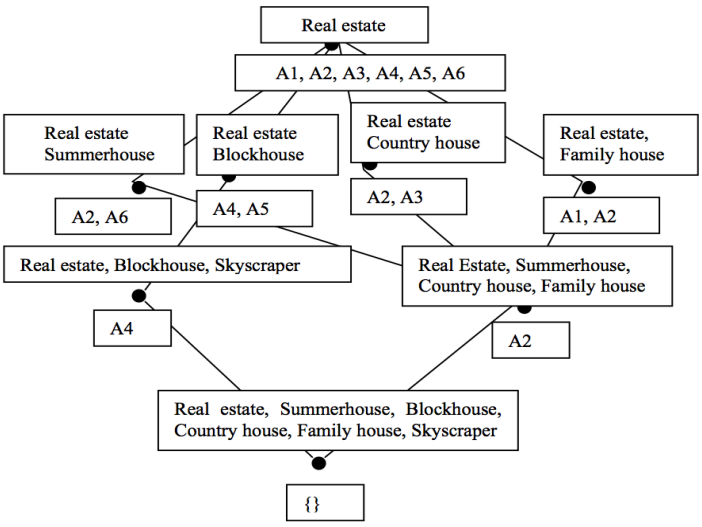


FIG. 5.24 – Exemple de la méthode de Haav, la figure à gauche décrit le contexte formel de cette méthode et la figure à droite décrit le treillis de concepts correspondant

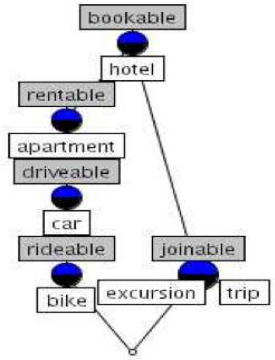
Cette méthode présente aussi quelques désavantages, tels que le fait qu'elle n'a pas été appliquée à un grand corpus de textes, la hiérarchie de termes résultante ne repose que sur leurs cooccurrences et enfin, elle ne propose aucune évaluation de la hiérarchie résultante.

Dans [Cimiano *et al.*, 2005; Cimiano, 2006], les auteurs s'appuient sur l'hypothèse de distribution de Harris [1968] pour extraire à partir d'un corpus de textes, des objets et leurs attributs puis ils créent une hiérarchie de concepts en utilisant l'AFC. Ensuite, ils transforment le treillis en une ontologie de concepts. Le contexte formel utilisé est $\mathbb{K}=(\mathtt{G},\mathtt{M},\mathtt{I})$:

- G : ensemble d'objets,
- M : ensemble des verbes,
- $I(g,m)$ si et seulement si le l'objet g est présent comme argument du verbe m (un argument peut être : le sujet, le complément d'objet direct ou indirect, ...).

Dans leur article Cimiano et al. [2005], construisent un contexte à partir d'un corpus de textes sur le domaine du tourisme (voir figure 5.25). Puis, ils construisent à l'aide de l'AFC le treillis de concepts correspondant et enfin ils transforment le treillis en une hiérarchie de concepts dans le schéma de l'ontologie.

	bookable	rentable	driveable	rideable	joinable
hotel	x				
apartment	x	x			
car	x	x	x		
bike	x	x	x	x	
excursion	x				x
trip	x				x



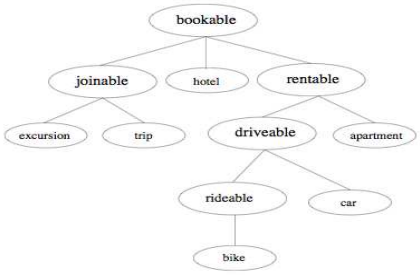


FIG. 5.25 – Exemple de la méthode de Cimiano, la figure à gauche décrit le contexte formel de cette méthode, la figure du milieu décrit le treillis de concepts correspondant et la figure à droite représente l'ontologie résultant du treillis de concepts

Cette méthode propose un ensemble d'attributs pour chaque ensemble d'objets, puis transforme le treillis $(\mathfrak{B}(G, M, I), \sqsubseteq)$ résultant en une ontologie $(C, \sqsubseteq')$. Cette transformation est faite comme suit :

$C := G \cup \{B \mid (A, B) \in \mathfrak{B}\}$

$\sqsubseteq' := \{(g, B_1) \mid g \text{ est l'objet propre du concept } (A_1, B_1)\} \cup \{(A_1, B_1) \sqsubseteq (A_2, B_2)\}$

Cependant, cette méthode ne définit que des concepts (les objets du treillis de concepts sont aussi considérés comme étant des concepts) et elle se limite à la hiérarchie de concepts sans prendre en compte les relations entre concepts. De plus, elle n'utilise que les corpus de textes comme ressource pour construire le hiérarchie de concepts.

5.5 Discussion

L'application du processus PACTOLE sur deux domaines spécifiques nous a permis de comprendre que dans certains corpus un type de descripteurs d'objets pouvait dominer. Dans le domaine de l'astronomie, les textes ne contenaient pratiquement que des descripteurs binaires, alors que dans le domaine de la microbiologie, nous avons l'effet inverse. Ce qui signifie que le corpus ne contient que des descripteurs d'objets relationnels et presque pas de descripteurs d'objets de type : attributs binaires.

Nous avons aussi choisi l'AFC qui offre plusieurs avantages, tels que le fait d'avoir des hiérarchies de concepts incrémentales, bien fondées mathématiquement et assez faciles à représenter dans une logique de descriptions comme $\mathcal{FLE}$. L'AFC nous a permis d'associer des définitions (ensemble d'attributs binaires) à des classes (ensemble d'objets) prédéfinies par les experts, ainsi que d'enrichir la hiérarchie des classes en proposant de nouvelles classes. L'Analyse Relationnelle de Concepts (ARC) nous permet de proposer des définitions incluant des relations avec d'autres types d'objets (attributs relationnels).

Néanmoins, l'AFC et l'ARC présentent aussi quelques désavantages, comme la production d'un

très grand nombre de concepts. La taille du treillis peut atteindre ($2^{min\{n,m\}}$) concepts (n est le nombre d'objets et m le nombre d'attributs du contexte) [Ganter et Wille, 1999]. Pour traiter ce problème d'explosion combinatoire, certains travaux ont proposé des solutions pour contrôler le nombre de concepts. Par exemple les travaux de [Stumme *et al.*, 2002] ou ceux de [Messai *et al.*, 2008]. [Stumme *et al.*, 2002] proposent de limiter la construction du treillis de concepts aux concepts les plus généraux. Il s'agit alors de fixer un seuil minimal contrôlant la taille de l'extension des concepts extraits (treillis d'iceberg). [Messai *et al.*, 2008] dans le cadre des treillis de concepts multivalués, apportent une solution double permettant de contrôler l'évolution du nombre de concepts. D'une part, les treillis de concepts multivalués sont directement déduits des contextes multivalués sans avoir à recourir à l'échelonnage ce qui réduit la taille du contexte. D'autre part, la génération des concepts est fonction de la similarité entre les données dans le contexte. La variation de la similarité entraîne la variation du nombre de concepts dans le treillis obtenu.

Chapitre 6

Expérimentations et évaluation

Sommaire

Introduction

Dans ce chapitre, nous présentons dans les deux premières sections les expérimentations effectuées sur les deux domaines d'application de notre processus ; l'astronomie et la microbiologie. Puis, dans la troisième section, nous détaillons les interactions possibles entre les experts du domaine et le processus PACTOLE. Enfin, la dernière section de ce chapitre donne les différents types d'unités de connaissances que le processus PACTOLE a pu extraire dans les deux domaines d'application.

6.1 Evaluation du processus PACTOLE dans le domaine de l'astronomie

Dans le domaine de l'astronomie, l'enjeu est de pouvoir proposer des définitions aux classes d'objets prédéfinies de la base SIMBAD. Le domaine dispose de plusieurs ressources textuelles : bases de données, thésaurus et corpus de textes. Le processus PACTOLE a été appliqué sur un corpus de 11591 résumés (2 069 062 mots) du journal A&A « Astronomy and Astrophysics » de 1994 à 2002 et sur la base de données SIMBAD.

6.1.1 Construction de treillis de concepts à partir de corpus de textes

La construction de treillis de concepts à partir de corpus de textes a été faite à partir du contexte formel $\mathbb{K}_2 := (G, M_2, I_2)$ tel que : G est l'ensemble des objets célestes, M_2 l'ensemble des attributs extraits des textes et $I_2 \subseteq G \times M_2$ où $I_2(g, m_2)$ veut dire que l'objet g possède l'attribut m_2. L'extraction des attributs binaires a été faite par l'analyse syntaxique STANFORD PARSER. Cet analyseur a extrait l'arbre syntaxique de chaque phrase des textes ainsi que les dépendances syntaxiques entre les termes. Le STANFORD PARSER a analysé 68.5 % des phrases du corpus (ce qui représente 60026 phrases). Plusieurs raisons pourraient expliquer pourquoi le reste des phrases n'a pas été analysé :

- les phrases étaient trop longues (la taille maximale des phrases analysées se situe entre 31 et 36 mots d'après la complexité syntaxique de la phrase). Des phrases très longues nécessitent beaucoup de mémoire vive et l'analyseur syntaxique a été installé sur une machine de 4Go de RAM.
- les phrases complexes contenaient trop de chiffres et de symboles. Par exemple, nous prenons la phrase : *« Radio observations were made at 3.6, 6 and 20 cm. Of the 15 observed* `RS_CVn` *systems, we detected 11 with {sigma} = 4 confidence at one or more wavelengths. »* Celle-ci rend le travail de l'analyseur très difficile, car rappelons le, nous avons choisi un analyseur syntaxique partiel pour sa robustesse.

Afin d'évaluer les dépendances syntaxiques, c'est-à-dire savoir quelles dépendances donnent le plus de paires intéressantes pour l'expert et lesquelles donnent le plus d'attributs aux objets, nous extrayons deux ensembles différents de dépendances syntaxiques entre les verbes et leurs arguments. Ces deux ensembles sont nommés respectivement SO et SOC avec :

- SO : subject(object,verb) + object(object,verb). Par exemple, de la phrase *« `NGC_5195` is connected to `M51` by a bridge of diffuse X-ray emission »* l'analyseur extrait deux dépendances. La première `nsubjpass(connected-3, NGC_5195-1)` veut dire que l'objet `NGC_5195` est le sujet du verbe `to connect` conjugué au passé. La deuxième `dobj(connected-3, M51-5)` veut dire que l'objet `M51` est l'objet direct du verbe `to connect`,
- SOC : SO + complément d'objet indirect(object,verb) + préposition_X(object,verb), où X peut être une préposition (in, of, ...). Par exemple, prenons la phrase : *« We report on the spectroscopic observations of 2 new planetary nebulae (PN), firstly identified by Terzan, one of them probably belonging to the galactic_ bulge. Diameters and radial velocities are estimated. Line intensities are given. »* Cette dernière nous permet d'extraire la dépendance `prep_to(belonging-25, galactic_bulge-28)` qui veut dire que l'objet `galactic_bulge` est lié au verbe `belonging` par la préposition `to`. Nous pouvons en déduire que cet objet peut contenir d'autres objets.

Néanmoins, les prépositions peuvent aussi être liés deux noms et ainsi nous permettre de définir un objet céleste non seulement par des verbes, mais aussi par des noms. Nous prenons les exemples de phrases suivants :

- *« Photometric measurements in four colours and visual observations of the* `AR_Pav` *covering almost 7 cycles of the 604.5 day orbit between 1982 and 1993 are presented. »* Pour celle-ci, l'analyseur extrait la dépendance : `prep_of(observations-8, AR_Pav-14)` qui permet de savoir que l'objet `AR_Pav` est lié au mot `observations`. Nous pouvons déduire que cet objet possède l'attribut `isObserved`,
- *« Direct images taken in 1988 and 1993 and spectra of the bipolar reflection nebula* `RNO_138` *are presented. »* Dans ce cas, l'analyseur extrait la dépendance : `prep_of(spectra-9, RNO_138-15)` qui permet de savoir que l'objet `RNO_138` est lié au mot `spectra`. Nous pouvons déduire que cet objet possède l'attribut `isEmitting`,
- *« We find that the 60* #181;*m emission of* `Vega` *is as extended as 35* #177; *5 arcsec. »* Cette fois, l'analyseur extrait la dépendance : `prep_of(emission-8, Vega-10)` qui permet de savoir que l'objet `Vega` est lié au mot `emission`. Nous pouvons déduire que cet objet possède l'attribut `isEmitting`.

L'ajout de ces dépendances constitue plus de 1793 des paires extraites, ce qui nous permet d'avoir une meilleure description des objets célestes. Nous définissons un nouvel ensemble de dépendance nommé SOCP : SOC + preposition_X(object,noun), où X peut être une préposition (in, of,...). L'ensemble des paires extraites des textes pour chaque type de dépendance est décrit dans le tableau 6.1. Nous remarquons que l'utilisation de l'ensemble SOCP nous permet d'avoir 230 objets et 6 attributs de plus qu'avec les dépendances utilisant le verbe comme argument.

	SO				SOC				SOCP			
Nombre de textes	Nombre de paires	Nombre d'objets	Nombre d'attributs	Nombre de concepts	Nombre de paires	Nombre d'objets	Nombre d'attributs	Nombre de concepts	Nombre de paires	Nombre d'objets	Nombre d'attributs	Nombre de concepts
50	1	1	1	1	4	4	2	4	7	5	2	4
100	6	4	2	4	13	9	2	4	28	15	4	7
500	22	15	3	5	48	28	3	6	148	67	7	12
1000	40	31	4	6	94	49	4	7	282	109	8	14
2000	105	65	8	13	218	93	8	14	563	168	12	28
3000	163	90	9	15	343	124	10	19	840	223	14	36
4000	203	107	10	16	451	151	10	21	1083	263	14	36
5000	268	136	10	19	576	187	10	24	1359	312	15	43
6000	426	224	12	25	887	311	12	29	2023	511	18	56
7000	489	236	12	25	1010	333	12	32	2282	542	19	57
8000	556	256	12	28	1149	357	13	34	2575	567	19	62
9000	612	272	12	29	1272	382	13	35	2830	601	21	63
10000	672	284	12	31	1396	395	15	37	3067	626	21	70
11000	723	299	12	31	1504	413	15	37	3261	644	21	71
11591	740	303	13	34	1537	417	15	37	3330	647	21	70

TAB. 6.1 – Evolution des résultats de l'AFC par rapport aux nombre de textes dans le corpus

L'ensemble de dépendances SOCP permet d'extraire plus de paires, plus d'attributs et plus de concepts que les autres ensembles de dépendances (voir le tableau 6.1), ce qui paraît logique : plus il y a de dépendances prises en compte, plus il y a de paires extraites et par conséquent plus d'objets, d'attributs et de concepts.

Les quatre graphiques des figures 6.1 et 6.2 montrent l'évolution du nombre de paires (voir figure 6.1), d'objets (voir figure 6.1), d'attributs (voir figure 6.2) et de concepts (voir figure 6.2) par rapport au nombre de textes dans le corpus de textes. Ces graphiques nous permettent de proposer une statistique textuelle, des termes, des paires et des attributs que nous recherchons. La statistique textuelle est un outil destiné à parfaire l'analyse, la description, la comparaison, en un mot, le traitement des textes [Lebart et Salem, 1994]. De ces graphiques, nous remarquons

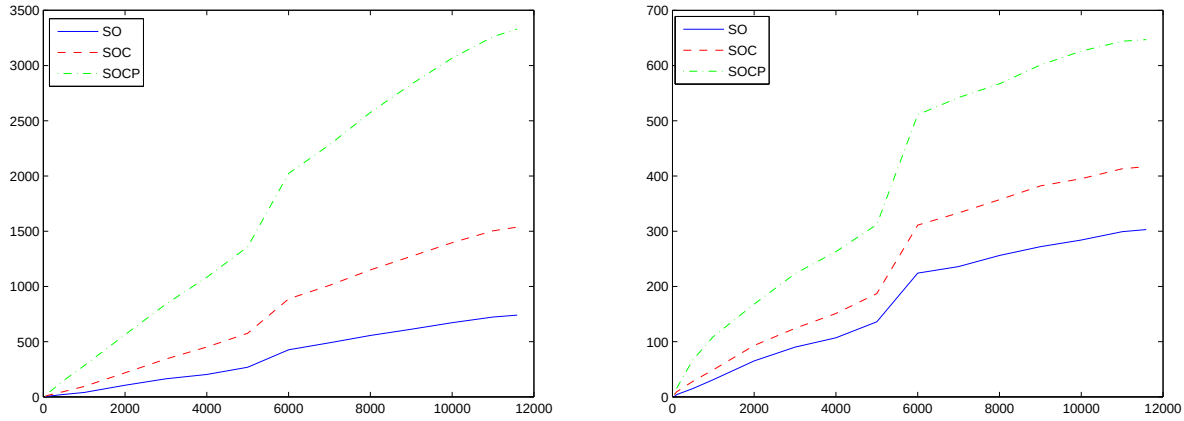


FIG. 6.1 – Graphiques montrant l'évolution du nombre de paires (à gauche) et du nombre d'objets (à droite). L'abcisse indique le nombre de textes

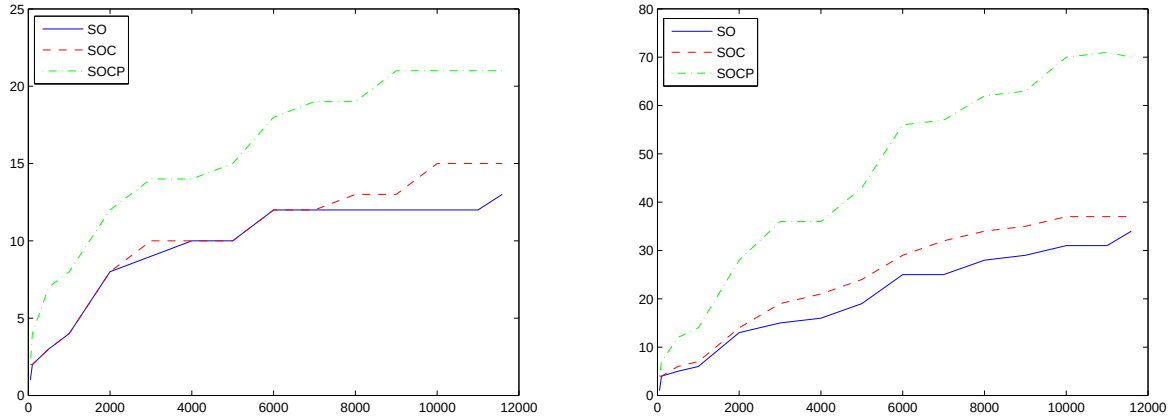


FIG. 6.2 – Graphiques montrant l'évolution du nombre d'attributs (à gauche) et du nombre de concepts (à droite). L'abcisse indique le nombre de textes

que plus il y a de textes, plus le nombre de paires et d'objets augmente. En revanche, le nombre des attributs binaires semble atteindre un certain seuil, ce qui limite aussi le nombre de concepts du treillis de concepts.

L'évolution du graphique des paires augmente pratiquement toujours de façon linéaire, tandis que dans le graphique des objets, nous pouvons observer une inflexion de la courbe vers les 6000 textes. Cette inflexion peut être expliquée par le fait qu'à un certain moment, les mêmes objets apparaissent dans les textes avec d'autres attributs. Nous remarquons aussi que s'il y a une accélération dans le nombre de paires, cette accélération se retrouve dans le nombre d'objets extraits, ce qui nous paraît logique. Le graphique des paires peut être expliqué par le fait que le nombre d'attributs binaires significatifs (les verbes et les noms liés par une préposition) est limité dans les textes. Enfin, le graphique des concepts peut être interprété par le fait que le nombre de concepts dans un treillis de concepts est fonction du minimum entre le nombre d'objets et le nombre d'attributs dans le contexte.

6.1.2 Construction de treillis de concepts à partir de la hiérarchie source

Nous considérons le contexte formel $\mathbb{K}_1 = (G, M_1, I_1)$, où : G est l'ensemble des objets célestes, M_1 est l'ensemble des classes de la base SIMBAD et I_1 est la relation binaire où $I_1(g, m_1)$ veut dire que l'objet g est attribué à la classe m_1 dans la hiérarchie source. A partir de ce contexte, l'AFC produit un treillis de 100 concepts, avec un contexte formel de 647 objets et 98 attributs (nous avons affecté la classe « `Object_of_unknown_nature` » aux objets qui n'apparaissaient pas dans SIMBAD).

6.1.3 Correspondance entres les deux hiérarchies de concepts

La correspondance entre les deux hiérarchies est effectuée pour déterminer si la hiérarchie extraite semi-automatiquement du corpus de textes produit les mêmes regroupements que la hiérarchie source construite manuellement par les experts (la base SIMBAD). En d'autres termes, pouvons nous associer aux classes de la base SIMBAD (classes de validation) les concepts définis par les attributs binaires (classes d'expérimentation) ?

Plusieurs travaux ont proposé des méthodes d'évaluation de correspondance entre deux hiérarchies. Par exemple, les approches de [Hearst, 1992] et de [Carpineto et Romano, 2000] consistent à calculer le nombre de relations hiérarchiques de la hiérarchie source qu'on retrouve dans la hiérarchie construite semi-automatiquement. Mais la méthodologie PACTOLE, comme celle de [Cimiano *et al.*, 2005], ne produit pas de label (nom de concept) pour les concepts générés, ce qui rend impossible l'utilisation de cette approche. Une autre évaluation possible est de mesurer la similarité entre les ensembles d'instances des concepts des deux hiérarchies. Les mesures de précision et de rappel sont utilisées pour évaluer cette similarité. Pour chaque classe de validation, il faut retrouver la classe d'expérimentation, puis une précision et un rappel locaux sont calculés entre ces deux classes. La précision globale (Precision_G) et le rappel global (Rappel_G) représentent la moyenne de toutes les précisions et de tous les rappels locaux respectivement.

Calcul de la précision et du rappel. La précision est le nombre d'instances communes entre C_{E_i} (la i-ème classe d'expérimentation) et C_{V_j} (la j-ème classe de validation) divisé par le nombre d'instances de la classe C_{E_i}. Autrement dit, la précision est le rapport entre le nombre de vrais positifs (les objets bien placés) sur le nombre total des objets de C_{E_i}. Le rappel est le nombre d'instances communes entre C_{E_i} et C_{V_j} divisé par le nombre d'instances de C_{V_j}. En d'autres termes, le rappel est le rapport entre les vrais positifs (les objets bien placés) et le nombre total des objets de C_{E_i}, soit N le nombre de classes de C_E.

$$Precision_i = \frac{|C_{E_i} \cap C_{V_j}|}{|C_{E_i}|}, \quad Rappel_i = \frac{|C_{E_i} \cap C_{V_j}|}{|C_{V_j}|}$$

$$Precision\_G = \frac{\sum_{i=1..N}(Precision_i)}{N}, \quad Rappel\_G = \frac{\sum_{i=1..N}(Rappel_i)}{N}$$

Détection de la classe la plus proche. Pour chaque classe C_{V_j} de la hiérarchie source, on recherche la classe la plus proche C_{E_i} de la hiérarchie extraite des textes qui partage le plus d'instances avec la classe C_{V_j}.

$$\forall C_{E_k} \in C_E, (C_{E_i} \cap C_{V_j}) = max(|C_{E_k} \cap C_{V_j}|) \wedge min(|C_{E_k} \setminus C_{V_j}|)$$

Par exemple, soit G l'ensemble des objets G = {`Cen_A, 3C_273, TWA, Per_OB2, T_Tauri, Y_Cygni, V773_Tau, Algol`} (voir figures 5.1 (page 85) et 5.2 (page 86)). La classe la plus proche de la classe C_{V_1} de SIMBAD avec les instances {`3C_273, Cen_A`} (figure 5.2) est la classe C_{E_1} avec les instances {`Per_OB2, Cen_A, 3C_273, TWA`} (figure 5.1), car le nombre d'instances communes est maximal ($|C_{E_k} \cap C_{V_j}|$ =2) et, le nombre d'instances qui diffère est minimal ($|C_{E_k} \setminus C_{V_j}|$=2).

TAB. 6.2 – Résultats des mesures de précision et de rappel pour l'application de l'AFC

	SO		SOC		SOCP	
	Precision_G	Rappel_G	Precision_G	Rappel_G	Precision_G	Rappel_G
AFC	58.33 %	05.03 %	60.89 %	09.65 %	70.50 %	26.84 %

L'ensemble SOCP donne de meilleurs résultats (voir le tableau 6.2) que les autres dépendances avec une bonne précision (70.50 %), ce qui veut dire que les objets ont été classés convenablement. En revanche, le rappel est très bas (26.84 %), cela peut être expliqué de différentes façons.

Premièrement, le nombre d'attributs associés aux objets n'est pas suffisant. Le processus PACTOLE n'a extrait que 21 attributs, car la plupart des verbes et des noms dans les textes ne sont pas significatifs et il est très difficile de trouver des attributs discriminants dans un domaine spécifique (1600 attributs binaires proposés et seulement 21 validés par les astronomes).

Deuxièmement, les classes ne sont pas seulement définies par les verbes et les noms, mais d'autres attributs binaires pourraient être considérés. Par exemple : les adjectifs, les adverbes, les mesures,... Le processus extrait ces dépendances du corpus de textes, mais ne les traite pas. Par exemple, si nous prenons la phrase « *Analysis of the gas velocity structure within GF 17 and GF 20 reveals evidence for smooth large-scale streaming motions along the filamentary structures with magnitude = 0.5 kmpc.* » La dépendance `conj_and(GF_17-18, GF_20-20)` qui exprime la conjonction pourrait permettre de déduire que les deux objets `GF_17` et `GF_20` ont des attributs communs. Ou encore, la mesure de magnitude « `magnitude = 0.5 kmpc` » pourrait être considérée comme un attribut des deux objets. D'autres dépendances sont intéressantes aussi, comme par exemple dans la phrase « *The highly inclined spiral NGC 4258 has been observed in X-rays with the PSPC of the Roentgen observatory ROSAT.* » La dépendance `nn(NGC_4258-5, spiral-4)`, entre les deux noms `NGC_4258` et `spiral`, nous permet de considérer `spiral` comme attribut de l'objet `NGC_4258`. Le choix de ne pas prendre en compte toutes les dépendances est dû au fait qu'elles introduisent aussi beaucoup de bruit. En effet, passer des dépendances (SO) aux dépendances (SOCP) augmente de façon considérable le nombre de paires (de 303 pour l'ensemble SO à 3330 pour l'ensemble SOCP). La validation de ces paires représente un travail considérable pour les astronomes.

Troisièmement, certains attributs binaires sont implicites et ne peuvent pas être extraits par un analyseur syntaxique, ce qui rend impossible la définition de toutes les classes proposées par les experts du domaine.

Cette expérimentation sur le corpus d'astronomie a montré que l'utilisation de toutes les dépendances syntaxiques (SOCP) donnait de meilleurs résultats que la prise en compte d'une partie de ces dépendances, même si ces dépendances ne sont pas suffisantes pour définir toutes les classes. L'utilisation d'autres dépendances du corpus de textes pourrait définir plus de classes, toutefois, il faut limiter le nombre de dépendances à extraire car plus nous extrayons de dépendances plus le travail de validation et d'interprétation des experts du domaine s'accroît.

6.1.4 Affectation des attributs binaires aux classes d'objets

Le treillis de concepts résultant de l'apposition de contextes a été présenté aux astronomes afin de déterminer si cette opération a permis de définir les classes d'objets ou d'enrichir la hiérarchie des classes avec de nouvelles classes. Cette opération a fait émerger de nouvelles unités de connaissances, comme par exemple, le concept ({`Orion, TWA`}, {`Association_of_stars, isExpanding, isObserved`}). Ce concept représente les « `Association_of_stars` » qui peuvent s'étendre (`isExpanding`). Ce concept a été considéré comme intéressant par les experts et a servi à définir une nouvelle classe « `Association_of_Young_Stars` ».

Extraction de relations dans le domaine de l'astronomie

	Precision_G	Rappel_G
ARC	52,19 %	25,07 %

TAB. 6.3 – Résultats de l'application de l'ARC dans le domaine de l'astronomie avec des mesures de précision et de rappel

Dans le domaine de l'astronomie, les astronomes nous ont conseillé de prendre en compte la relation `isObservedBy` entre les objets célestes et les télescopes. A partir de 11591 textes, GATE a extrait 200 phrases où la relation `isObservedBy` relie 10 télescopes à 64 objets célestes. Après avoir montré ces résultats aux astronomes (voir le tableau 6.3), nous avons décidé de ne plus prendre en compte la partie relationnelle de notre approche et de n'appliquer que la partie qui extrait des descriptions binaires. Néanmoins, l'extraction des attributs relationnels nous a permis d'extraire des « unités de connaissance » validées par les astronomes, comme par exemple : la classe `Binary_Star` qui peut être définie par les « `X-Ray_Telescopes` ».

6.2 Evaluation du processus PACTOLE dans le domaine de la microbiologie

Dans le domaine de la microbiologie, l'objectif de notre application était la classification des bactéries en prenant en compte leurs relations avec les gènes et les antibiotiques. Cette classification doit servir aussi à expliquer les phénomènes de résistance des bactéries aux antibiotiques par mutation de gènes. Le processus PACTOLE a été appliqué sur un corpus de 1244 résumés (213 960 mots) extraits d'une dizaine de journaux de 1997 à 1999. Ces résumés traitent tous de la résistance des bactéries aux antibiotiques par mutation de gènes. Nous pouvons citer quelques journaux de ce corpus tels que : MICROBIOLOGY, BACTERIOLOGY, COMMUNICABLE-DISEASES, MEDICINE, BIOCHEMISTRY ; BIOPHYSICS ... Ce corpus a été constitué par des experts de « l'Institut de l'Information Scientifique et Technique (INIST) »[26]. L'extraction des objets est faite à l'aide du logiciel GATE et d'un ensemble de listes (une pour chaque objet). Les listes des objets sont extraites avec l'aide des experts du domaine à partir de différentes bases de données telles que : la GENE ONTOLOGY ou PubMed,...

TAB. 6.4 – Les résultats de l'évaluation des hiérarchies des bactéries résultant de l'AFC et de l'ARC

	Méthode	Nombre d'objets	Nombre d'attributs binaires	Nombre d'attributs relationnels	Nombre de concepts
Gènes	AFC	15	14	0	32
	ARC	15	14	127	232
Bactéries	AFC	18	13	0	58
	ARC	18	13	55	152
Antibiotiques	AFC	26	23	0	39
	ARC	26	23	0	39

6.2.1 Construction des treillis de concepts à partir des bases de données et des corpus de textes

Dans cette partie, nous donnons dans le tableau 6.4 les résultats de l'application des deux méthodes de fouille, l'AFC et l'ARC, pour chaque contexte créé dans le domaine de la microbiologie (contexte gènes, contexte bactéries et contexte antibiotiques). Ces résultats comprennent le nombre d'objets, d'attributs binaires, d'attributs relationnels et de concepts. Nous remarquons que puisqu'il n'y a pas de relation ayant le contexte des antibiotiques en tant que domaine, le contexte des antibiotiques ne possède pas d'attribut relationnel lors de l'application de l'ARC. Par conséquence, le nombre de concepts résultant de l'ARC est le même que celui résultant de l'AFC. Nous remarquons aussi que le nombre de concepts dans le contexte des gènes augmente de 200 concepts avec l'application de l'ARC, car à chaque fois que le contexte des bactéries change (par l'échelonnage relationnel), le contexte des gènes évolue aussi.

6.2.2 Construction des treillis de concepts à partir de la hiérarchie source

Nous partons du contexte formel $\mathbb{K}_1 = (G, M_1, I_1)$, où : G est l'ensemble des bactéries, M_1 est l'ensemble des classes de la base NCBI et I_1 est la relation binaire où $I_1(g, m_1)$ veut dire que la bactérie g est attribué à la classe m_1 dans la hiérarchie source. A partir de ce contexte, l'AFC produit un treillis de 21 concepts, avec un contexte formel de 18 objets et 19 classes.

6.2.3 Correspondance entre les deux hiérarchies de concepts

Dans cette partie, nous avons effectué deux expérimentations différentes pour comparer les hiérarchies extraites à partir des classes de la base NCBI et celle extraite à partir d'attributs binaires ou relationnels (figure 6.3). La première utilise seulement l'Analyse Formelle de Concepts (AFC) et la deuxième utilise l'AFC et son extension l'Analyse Relationnelle de Concepts (ARC). Le tableau 6.5 présente les mesures de précision et de rappel résultant de cette évaluation. Le rappel est égal à 100 % avec l'AFC et avec l'ARC car toutes les classes de la base NCBI sont ou égales à celle de l'expérimentation, ou incluses dedans. Ce tableau présente aussi le nombre de classes avec 100 % de précision et de rappel que l'AFC ou l'ARC ont défini, ce qui signifie qu'elles donnent les conditions nécessaires et suffisantes pour appartenir à cette classe. L'ARC possède une précision moins élevés que l'AFC, mais elle définit plus de classes, car plusieurs classes sont

[26] http ://www.inist.fr/

TAB. 6.5 – Les résultats de l'évaluation des hiérarchies des bactéries résultant de l'AFC et de l'ARC

	Precision_G	Rappel_G	Classes définies
AFC	76,52 %	100 %	8/19
ARC	66,88 %	100 %	12/19

définies par des relations (voir le tableau 6.5). Nous présentons un exemple d'une classe qui n'a pas pu être définie en AFC, mais que l'ARC a pu définir. La classe `B9 ({Neisseria_Gonorrhoeae}, {Betaproteobacteria, Neisseria, Proteobacteria})` de la base NCBI n'a pas pu être définie par l'AFC car aucune classe résultant de l'AFC n'a le même ensemble d'instances. Néanmoins, avec l'ARC le processus a pu lui affecter le concept `B150` avec l'instance `{Neisseria_Gonorrhoeae}`. Dans notre expérimentation, nous avons remarqué que plus l'ensemble des instances d'une classe est réduit et plus il est difficile de trouver une définition à cette classe.

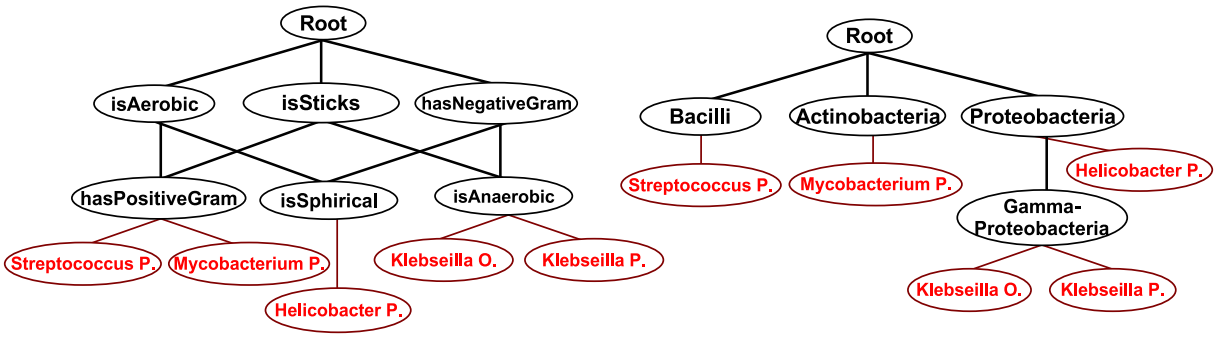


FIG. 6.3 – Un exemple des deux hiérarchies, l'une extraite de la base NCBI (à droite) et l'autre de base de données avec l'AFC (à gauche)

6.3 Interaction entre l'expert et le processus PACTOLE

L'expert est invité à interpréter les résultats de notre processus, à détecter les problèmes dans le schéma d'ontologie résultant avant sa représentation en DL et aider à construire notre ontologie finale [Bendaoud *et al.*, 2008c]. Les raisons de ces problèmes peuvent être de deux natures différentes : (1) les ressources du domaine peuvent contenir du bruit ou alors ce bruit est issu de la première étape d'extraction d'informations et d'analyse des textes, (2) les experts ne sont pas satisfaits par le schéma de l'ontologie résultant et ils veulent l'adapter à leurs besoins. Quelles que soient les raisons de ces problèmes, une étape de post-traitement est mise à disposition des experts. Cette étape définit des filtres qui peuvent être appliqués sur les ressources par les experts afin de ré-appliquer l'AFC/ARC et d'avoir une nouvelle version du schéma de l'ontologie résultant. Cette étape de post-traitement enregistre toutes les opérations effectuées par les experts pour garder une trace de leurs interactions avec notre processus PACTOLE et ne pas les obliger à refaire le même travail pour chaque nouveau corpus, par exemple. Nous définissons différents types d'opérations sur les contextes formels qui dépendent du type de descripteurs d'objets (*DO1, DO2* ou *DO3*). Afin de faciliter le travail des experts du domaine, nous leur proposons qu'au lieu de modifier les contextes formels, ils n'auront qu'à ajouter de simples lignes dans des fichiers

spécifiques. L'ajout de ces lignes représente la trace de la modification.

6.3.1 Les opérations sur la hiérarchie de liens *DO1*

Avant de présenter les opérations sur *DO1*, nous décrivons les fichiers qui servent au processus PACTOLE pour construire à partir de cette ressource un contexte formel. *DO1* est stockée dans le processus sous forme de deux fichiers différents. Le premier, nommé « HierarchieSource », stock des paires du type (`objet`$_i$, `classe`$_j$). Ces paires signifient que l'objet `objet`$_i$ est classé dans la classe `classe`$_j$. Le deuxième, nommé « LienHierarchiques », stocke des paires du type (`classe`$_i$, `classe`$_j$), ces paires signifient que la classe `classe`$_i$ est une sous-classe de la classe `classe`$_j$. Ce sont ces deux fichiers qui garderont les traces des opérations sur *DO1*.

Ajouter une nouvelle classe. Les experts du domaine peuvent insérer une nouvelle classe dans le thésaurus. Cette opération nécessite l'ajout d'une colonne dans le contexte formel représentant la hiérarchie et l'expert doit assigner les objets appropriés à cette classe.

TAB. 6.6 – Ajout de la classe `Association_of_Young_Stars` dans le contexte formel

	Attributs binaires							Classes SIMBAD						
	isObserved	isIncluding	isEmitting	isEclipsing	isExpanding	isIncluded	isFlaring	Quasar	Galaxy	Asso._of_Young_Stars	Asso._of_Stars	T_Tau_type_Star	Eclipsing_Binary	Star
Cen_A	×	×							×					
3C_273	×	×	×					×	×					
TWA	×	×			×					×	×			
Per_OB2	×	×									×			
T_Tauri	×		×			×	×					×		×
Y_Cygni	×		×	×		×							×	×
V773_Tau	×		×			×	×					×		×
Algol	×		×	×		×							×	×

Prenons l'exemple de la figure 5.3, nous avons présenté dans la sous-section 6.1.4 que l'expert peut à l'aide de l'apposition de contextes, définir une nouvelle classe dans le thésaurus. Par exemple, puisque l'objet `TWA` possède l'attribut `isExpanding`, l'expert peut décider de lui affecter la nouvelle classe `Association_of_Young_Stars`. Puisque la classe `Association_of_Young_Stars` est plus spécifique que la classe `Association_of_Stars`, la corrélation entre l'objet `TWA` et la classe `Association_of_Stars` est maintenue. La classe `Association_of_Young_Stars` est ajoutée au contexte dans le tableau 6.6. Le treillis correspondant à ce tableau est présenté dans la figure 6.4.

Trace de l'ajout d'une nouvelle classe dans le processus. Pour ajouter une nouvelle classe dans notre processus, l'expert va simplement ajouter des lignes de la forme (`objet`$_i$,

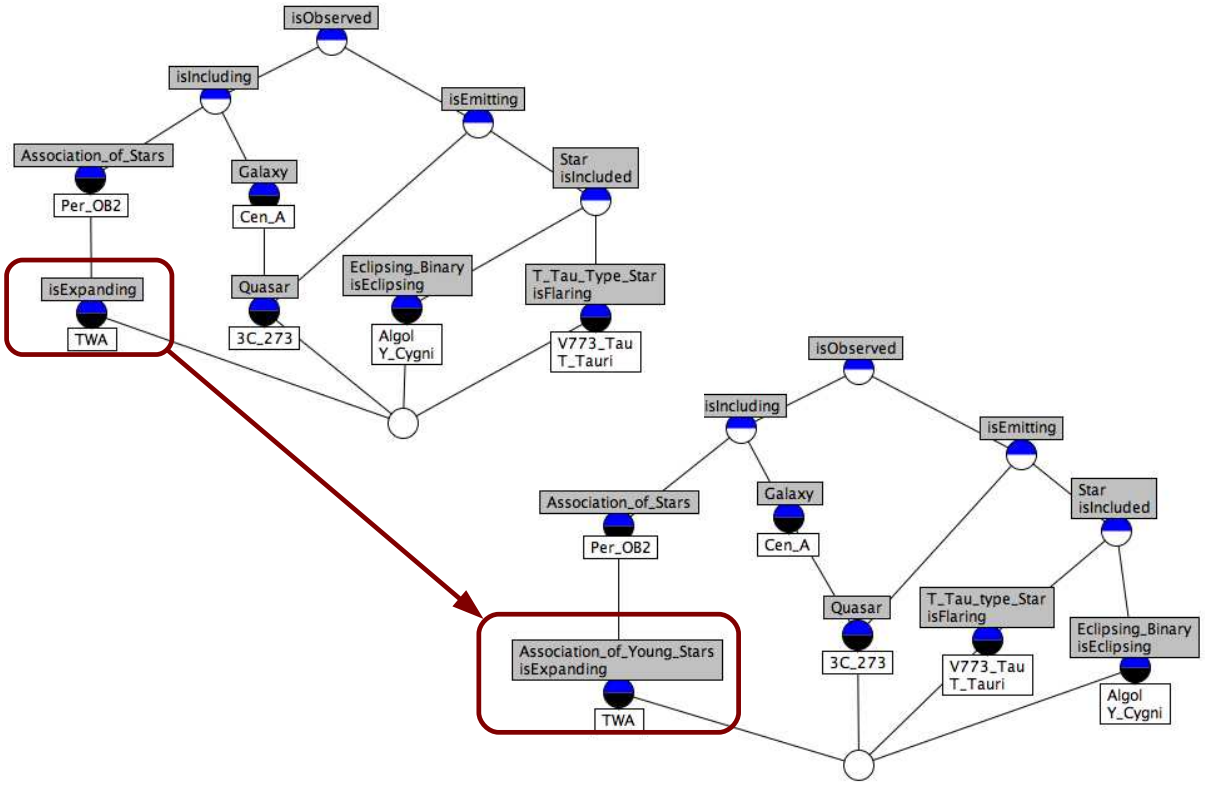


FIG. 6.4 – Transformation du treillis de concepts correspondant à la création d'une nouvelle classe `Association_of_Young_Stars` dans le contexte formel

`classe`$_j$`)` dans le fichier « HierarchieSource » pour affecter la nouvelle classe aux objets qui lui sont associés. Il doit aussi ajouter deux lignes du type `(classe`$_i$`, classe`$_j$`)` et `(classe`$_j$`, classe`$_k$`)` dans le fichier « LienHierarchique ». Ces deux lignes placent la nouvelle classe dans la hiérarchie source, avec `classe`$_i$ une sous-classe et `classe`$_k$ une super-classe de la nouvelle classe `classe`$_j$.

Pour notre exemple de le tableau 6.6, il suffit à l'expert d'ajouter la ligne `(TWA, Association_of_Young_Stars)` dans le fichier « HierarchieSource », ainsi que la ligne `(Association_of_Young_Stars, Association_of_Stars)` dans le fichier « LienHierarchique ». Il ajoute seulement une ligne, car il n'y a pas dans la hiérarchie une sous-classe de la classe `Association_of_Young_Stars`.

Changer la classe d'un objet. Pour changer la classe d'un objet, la ligne décrivant cet objet dans le contexte formel doit être considérée : l'ancienne classe et toutes ses super-classes doivent être supprimées comme attributs de l'objet. La nouvelle classe et toutes ses super-classes doivent être rajoutées comme attributs de l'objet.

Prenons par exemple l'objet `3C_273`. Si les astronomes décident qu'il n'a plus les attributs nécessaires pour être une `Quasar` la corrélation entre l'objet `3C_273` et la classe `Quasar` doit être supprimée. Ainsi, l'objet `3C_273` n'a de corrélations qu'avec la classe `Galaxy`. Le tableau 6.7 présente cet exemple, le treillis correspondant à ce tableau est présenté dans la figure 6.5.

TAB. 6.7 – Changement de classe pour l'objet `3C _273` dans le contexte formel

	Attributs binaires							Classes SIMBAD						
	isObserved	isIncluding	isEmitting	isEclipsing	isExpanding	isIncluded	isFlaring	Quasar	Galaxy	Asso._of_Young_Stars	Asso._of_Stars	T_Tau_type_Star	Eclipsing_Binary	Star
Cen_A	×	×							×					
3C_273	×	×	×						×					
TWA	×	×			×					×	×			
Per_OB2	×	×									×			
T_Tauri	×		×			×	×					×		×
Y_Cygni	×		×	×		×							×	×
V773_Tau	×		×			×	×					×		×
Algol	×		×	×		×							×	×

Trace du changement d'une classe à un objet dans le processus. Pour changer un objet de classe, l'expert doit simplement supprimer la ligne (`objet`$_i$, `classe`$_j$) de cet objet et la remplacer par la ligne (`objet`$_i$, `classe`$_k$) dans le fichier « HierarchieSource ». La même opération est effectuée dans le cas particulier où l'expert ne souhaite pas changer la classe d'un objet, mais seulement affecter cet objet à une super-classe de sa classe. L'exemple du tableau 6.7 présente ce cas. Il suffit à l'expert de supprimer la ligne `(3C_273, Quasar)` et d'ajouter la ligne `(3C_273, Galaxy)` dans le fichier « HierarchieSource ».

Supprimer une classe. La suppression d'une classe revient à supprimer la colonne de cette classe dans le contexte formel de la hiérarchie. Cette opération est définie dans notre processus, mais d'après notre interaction avec les experts des deux domaines d'application, elle n'est jamais utilisée.

Par exemple, nous prenons le contexte du tableau 6.7. La classe `Quasar` ne possède plus aucun objet. Les astronomes peuvent décider de supprimer cette classe. Le tableau 6.8 présente cet exemple, le treillis correspondant à ce tableau est présenté dans la figure 6.6.

Trace de la suppression d'une classe dans le processus. Pour supprimer une classe dans la « hiérarchie source », l'expert doit remplacer toutes les lignes (`objet`$_i$, `classe`$_j$) par les lignes (`objet`$_i$, `classe`$_k$) dans le fichier « HierarchieSource », tel que `classe`$_k$ est la super-classe la plus spécifique de la classe `classe`$_j$.

Pour notre exemple du tableau 6.8, il suffit à l'expert de supprimer la ligne `(Quasar, Galaxy)` dans le fichier « LienHierarchique ». Puisqu'aucun objet n'est affecté à cette classe, l'expert ne touche pas le fichier « LienHierarchique ».

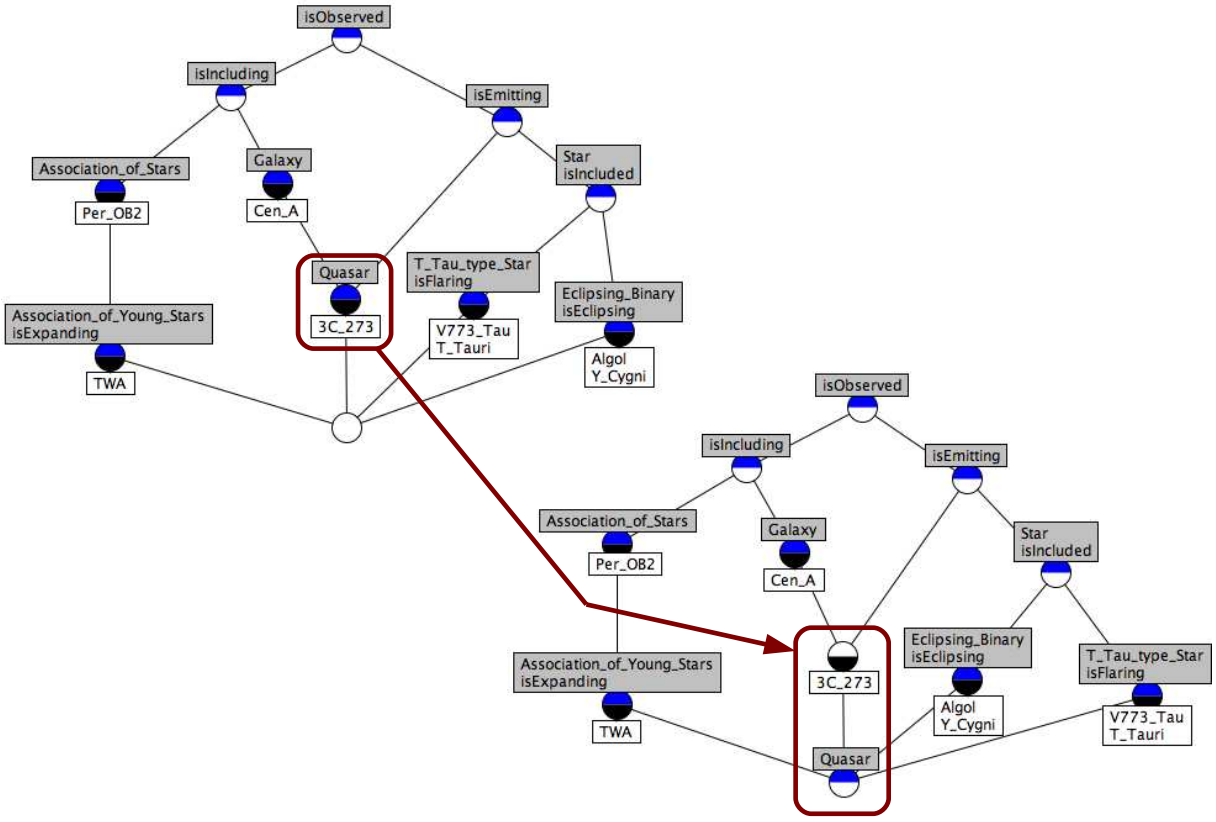


FIG. 6.5 – Transformation du treillis de concepts correspondant au changement de classe de objet `3C_273` dans le contexte formel

6.3.2 Les opérations sur *DO2*.

La qualité des ressources peut varier selon leur forme : bases de données, textes... Par exemple, les outils de TALN ou ceux d'extraction d'informations introduisent beaucoup de bruit. Le niveau linguistique est beaucoup plus détaillé que le niveau de l'ontologie. Les experts doivent enlever tout le bruit introduit par ces outils. Les opérations suivantes vont leurs permettre d'y arriver.

Le fichier qui garde les traces des changements des descripteurs d'objets *DO2* et *DO3* est nommé « AttributsBinaires ».

Fusionner des attributs. Cette opération permet aux experts de fusionner les attributs qu'ils considèrent comme étant des synonymes (dans le domaine de l'astronomie). Par exemple, nous prenons la phrase *« We derive a new mass estimate for the ejecta of the young supernova remnant Cas_ A of M 4M{sun}. »* Dans ce cas, l'analyseur syntaxique extrait la paire `prep_of(ejecta-9, Cas_A-15)` qui est transformée par notre processus en la paire `(isEjecting, Cas_A)`. Si l'expert considère que les deux attributs `isEmitting` et `isEjecting` sont des synonymes, il peut fusionner ces deux attributs en un seul (voir le tableau 6.9). L'opération de fusion des attributs consiste à fusionner les deux colonnes de ces deux attributs. Cette opération est équivalente au constructeur logique « ou » dans les colonnes correspondantes dans le contexte formel. La transformation dans le treillis de concepts est présentée dans la figure 6.9

TAB. 6.8 – Suppression de la classe `Quasar` dans le contexte formel

	Attributs binaires							Classes SIMBAD					
	isObserved	isIncluding	isEmitting	isEclipsing	isExpanding	isIncluded	isFlaring	Galaxy	Asso._of_Young_Stars	Asso._of_Stars	T_Tau_type_Star	Eclipsing_Binary	Star
Cen_A	×	×						×					
3C _273	×	×	×					×					
TWA	×	×			×				×	×			
Per_OB2	×	×								×			
T_Tauri	×		×			×	×				×		×
Y_Cygni	×		×	×		×						×	×
V773_Tau	×		×			×	×				×		×
Algol	×		×	×		×						×	×

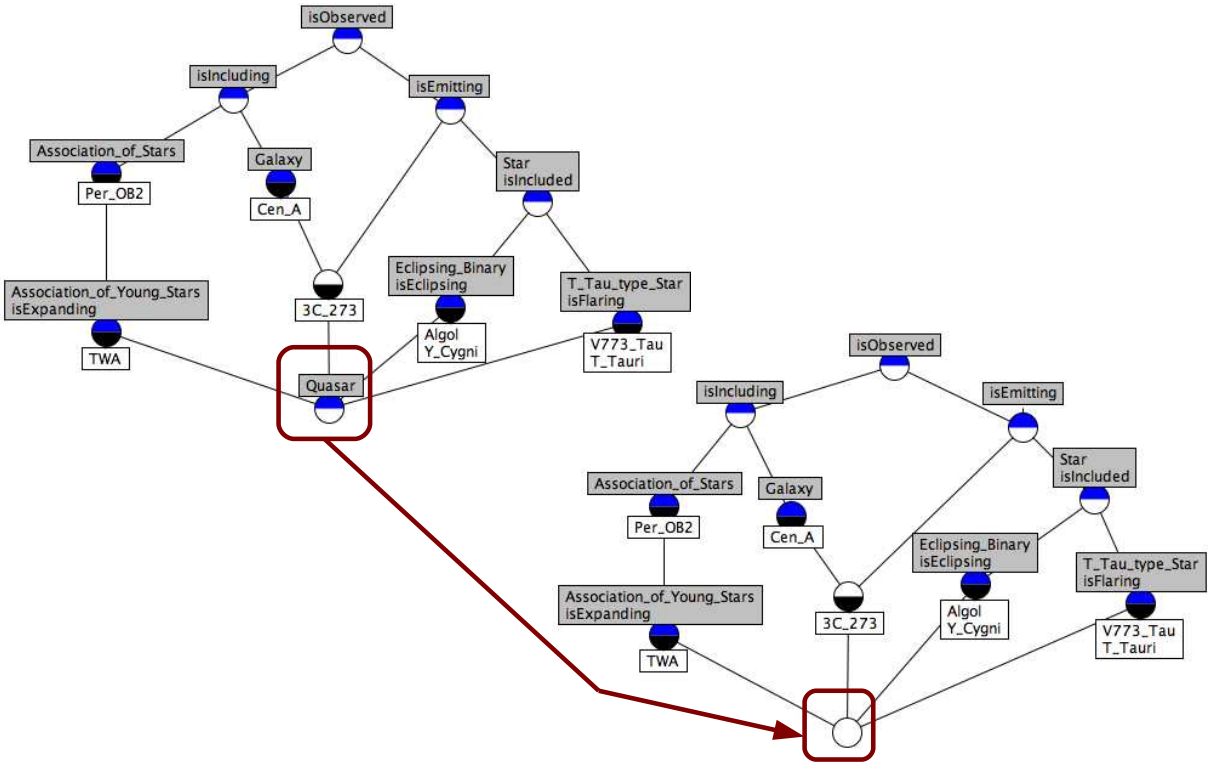


FIG. 6.6 – Transformation du treillis de concepts correspondant à la suppression de la classe `Quasar` dans le contexte formel

TAB. 6.9 – Fusion des deux attributs `isEmitting` et `isEjecting` dans le contexte formel

	Attributs binaires								Classes SIMBAD					
	isObserved	isIncluding	isEjecting	isEmitting	isEclipsing	isExpanding	isIncluded	isFlaring	Galaxy	Asso._of_Young_Stars	Asso._of_Stars	T_Tau_type_Star	Eclipsing_Binary	Star
Cen_A	×	×	×						×					
3C _273	×	×		×					×					
TWA	×	×				×				×	×			
Per_OB2	×	×									×			
T_Tauri	×			×			×	×				×		×
Y_Cygni	×			×	×		×						×	×
V773_Tau	×			×			×	×				×		×
Algol	×			×	×		×						×	×

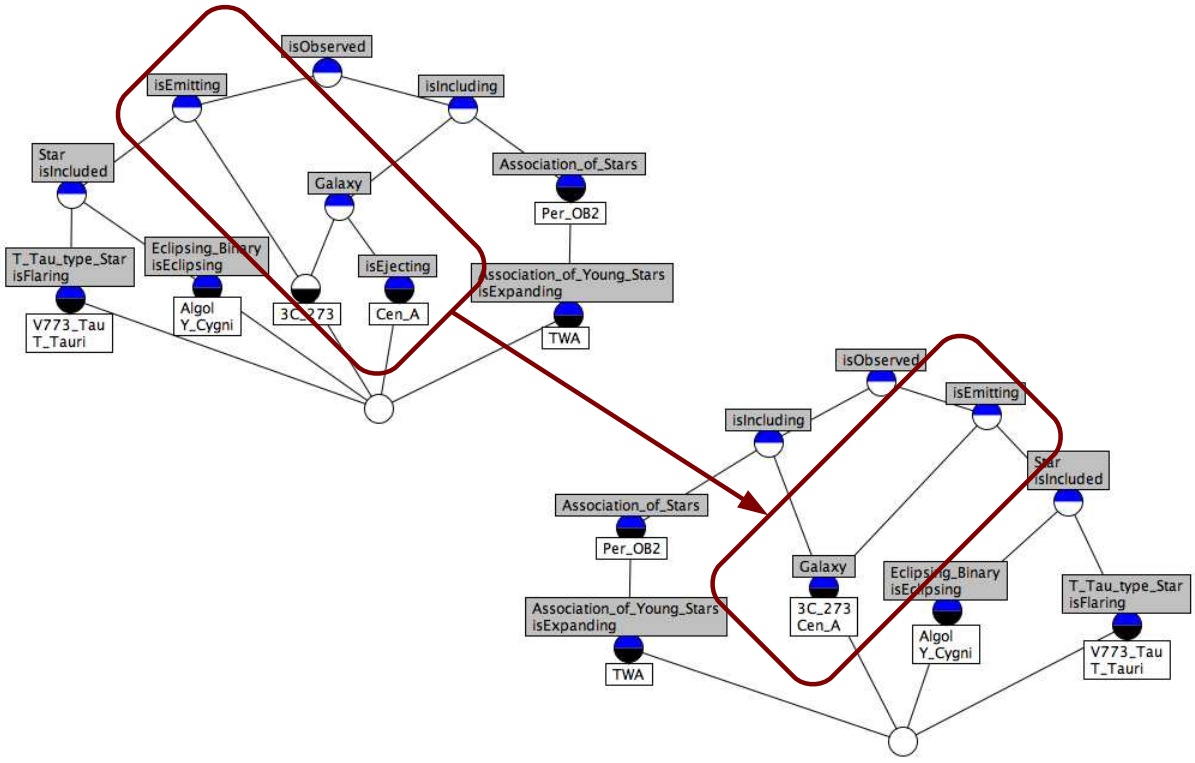


FIG. 6.7 – Transformation du treillis des concepts correspondant à la fusion des deux attributs `isEmitting` et `isEjecting` dans le contexte formel

Trace de la fusion des attributs dans le processus. Pour fusionner deux attributs $\texttt{attribut}_1$ et $\texttt{attribut}_2$ et ne garder que l'attribut $\texttt{attribut}_2$, on doit ajouter une substitution par expression régulière dans le fichier « AttributsBinaires ». Cette expression est de la forme : $\texttt{\$ligne=~s /attribut}_1\texttt{/attribut}_2\texttt{/ ;}$

Cette expression permet de supprimer l'attribut concerné dans ce contexte et de garder la trace de cette suppression. Ainsi, même si les ressources utilisées changent, les deux attributs seront toujours fusionnés par notre processus. Nous prenons l'exemple du tableau 6.9 qui fusionne les deux attributs `isEjecting` et `isEmitting` pour ne garder que l'attribut `isEmitting`. Cette opération nécessite l'ajout de l'expression régulière suivante :

```
$ligne=~s /isEjecting/isEmitting/ ;
```

Diviser un attribut en deux. Plusieurs attributs peuvent être regroupés en un seul (généralement parce que les experts les définissent comme étant des synonymes), mais parfois les experts commettent des erreurs en regroupant des attributs qui n'ont pas réellement la même signification. Par exemple, les deux attributs `isReddening` et `isRedshifting` ont été regroupés dans le même attribut `isReddening` (Voir le tableau 6.10). Voici des phrases qui ont servi à extraire des dépendances incluant ces deux attributs :

- *« This conclusion is supported by the correlations between the forbidden line luminosities, the near infrared excess, and the reddening of T_ Tauri stars from Cabrit et al. (1990ApJ...354..687C). »* Dans ce cas, l'analyseur extrait la dépendance syntaxique `prep_of(reddening-21, T_Tauri-23)` qui est transformée en `(isReddening,T_Tauri)`,
- *« In addition we gathered spectroscopy for the Mkn_421 with unknown redshift and for companion galaxies. »* Alors, l'analyseur extrait la dépendance syntaxique `prep_for(redshift-11, Mkn_421-8)` qui est transformée en `(isReddening, Mkn_421)`.

Ainsi, un concept contenant les deux objets `T_Tauri` et `Mkn_421` est créé dans le treillis puisque les deux objets possèdent au moins un attribut en commun. Les experts, en examinant ce concept, se rendent compte qu'il n'est pas pertinent et qu'il faudrait mieux le diviser en deux et que de plus l'attribut `isReddening` ne convient pas à l'objet `Mkn_421` (l'objet `Mkn_421` ne rougit pas, il est observé par rayonnement infrarouge). Pour diviser cet attribut en deux, il faut créer deux attributs différents et se référer au corpus de textes pour affecter le bon attribut aux bons objets. Le contexte formel résultant est présenté dans le tableau 6.10.

Trace de la division d'un attribut en deux dans le processus. Pour diviser un attribut $\texttt{attribut}_1$ en deux attributs $\texttt{attribut}_1$ et $\texttt{attribut}_2$, l'expert doit supprimer l'expression régulière dans le fichier « AttributsBinaires ». Cette expression est de la forme :

$\texttt{\$ligne=~s /attribut}_1\texttt{/attribut}_2\texttt{/ ;}$

Dans le cas de l'exemple du tableau 6.10, pour diviser l'attribut `isReddening` en deux attributs `isReddening` et `isRedshifting`, il faut supprimer dans le fichier « AttributsBinaires » l'expression régulière suivante :

```
$ligne=~s /isRedshifting/isReddening/ ;
```

Supprimer un attribut à un objet. Un attribut peut être mal attribué à un objet à cause par exemple d'une mauvaise dépendance syntaxique extraite par l'analyseur ; l'expert peut donc le lui supprimer. Dans le contexte formel, cela revient à changer en « faux » la cellule (objet, attribut) correspondante.

TAB. 6.10 – Division de l'attribut `isReddening` en deux attributs distincts `isReddening` et `isRedshifting` dans le contexte formel

	Attributs binaires									Classes SIMBAD						
	isObserved	isIncluding	isReddening	isEmitting	isEclipsing	isExpanding	isIncluded	isFlaring	isRedshifting	Galaxy	Asso._of_Young_Stars	Asso._of_Stars	T_Tau_type_Star	Eclipsing_Binary	Star	BL_Lac
Cen_A	×	×		×						×						
3C_273	×	×		×						×						
TWA	×	×				×					×	×				
Per_OB2	×	×										×				
T_Tauri	×		×	×			×	×					×		×	
Y_Cygni	×			×	×		×							×	×	
V773_Tau	×		×	×			×	×					×		×	
Algol	×			×	×		×							×	×	
Mkn_421	×						×		×	×						×

Trace de la suppression d'un attribut à un objet dans le processus. Pour supprimer un attribut à un objet, il suffit d'ajouter une expression régulière qui supprime la paire dans (objet$_j$, attribut$_i$) le fichier « AttributsBinaires ». Cette expression est de la forme :

`$ligne=~s /(objet`$_j$`, attribut`$_i$`)// ;`

Par exemple, si l'expert veut enlever la corrélation entre l'objet `Cen_A` et l'attribut `isEmitting`, il ajoute dans le fichier « AttributsBinaires » l'expression :

```
$ligne=~s /(Cen_A,isEmitting)// ;
```

Supprimer un attribut pour tous les objets. Lorsque les experts interprètent le schéma d'ontologie résultant, ils peuvent s'apercevoir qu'un attribut n'est pas très significatif. La colonne avec cet attribut est alors supprimée dans le contexte formel.

Par exemple, l'attribut `isObserved` est possédé par tous les objets célestes. Les astronomes pourraient décider de la supprimer. Cette suppression n'aura pas d'impact sur la structure du treillis. Le tableau 6.11 présente cet exemple, le treillis correspondant à ce tableau est présenté dans la figure 6.9.

Trace de la Fusion des attributs dans le processus. Pour supprimer un attribut `attribut`$_1$ du contexte formel, l'expert doit ajouter une expression régulière dans le fichier « AttributsBinaires ». Cette expression est de la forme : `$ligne=~s /attribut`$_1$`//g ;`

Si nous prenons l'exemple du tableau 6.11, pour supprimer l'attribut `isObserved`, on doit ajouter dans le fichier « AttributsBinaires » l'expression régulière `$ligne=~s /isObserved// ;`

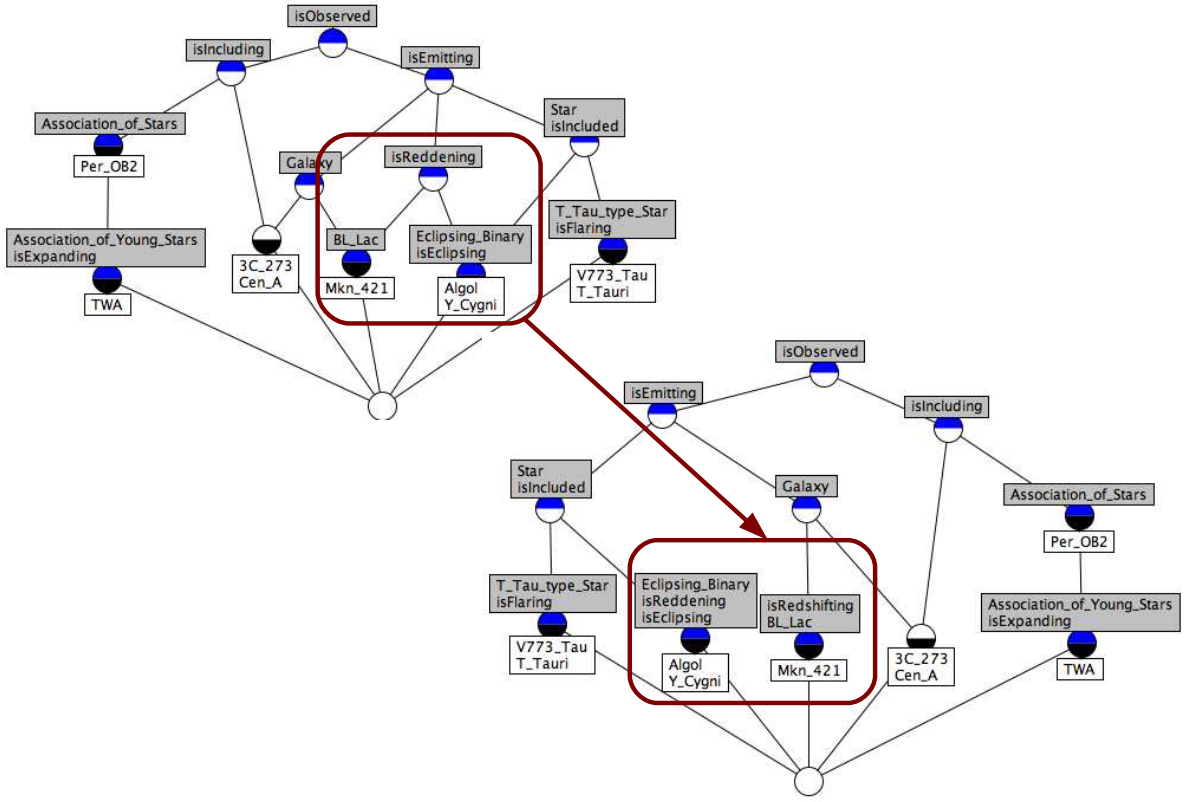


FIG. 6.8 – Treillis correspondant à la division l'attribut `isReddening` en deux attributs distincts `isReddening` et `isReddening` dans le contexte formel

Ajouter un attribut à un ensemble d'objets. Si l'expert considère qu'un attribut intéressant n'a pas été utilisé pour décrire un ensemble d'objet, il peut décider de le rajouter pour tous les objets concernés.

Par exemple, les experts pourraient considérer l'attribut `isOscillating` comme étant intéressant pour définir les objets de la classe `Eclipsing_Binary`. Le tableau 6.12 présente cet exemple ; le treillis correspondant à ce tableau est présenté dans la figure 6.10.

Trace de l'ajout d'un attribut dans le processus. Pour ajouter un attribut $\texttt{attribut}_1$, l'expert doit supprimer l'expression régulière qui regroupe cet attribut avec un autre dans le fichier « AttributsBinaires ». Cette expression est de la forme : $\texttt{\$ligne=~s /attribut}_1\texttt{/attribut}_2\texttt{/g ;}$

Dans le cas de l'exemple de le tableau 6.12, pour ajouter l'attribut `isOscillating`, on doit supprimer dans le fichier « AttributsBinaires » l'expression régulière `$ligne=~s /isOscillating/isMoving/g ;`

6.3.3 Les opérations sur *DO3*.

Les opérations sur les attributs relationnels sont similaires à celles appliquées sur les attributs binaires.

TAB. 6.11 – Suppression de l'attribut `isObserved` à tous les objets du contexte formel

	Attributs binaires							Classes SIMBAD						
	isIncluding	isReddening	isEmitting	isEclipsing	isExpanding	isIncluded	isFlaring	Galaxy	Asso._of_Young_Stars	Asso._of_Stars	T_Tau_type_Star	Eclipsing_Binary	Star	BL Lac
Cen_A	×		×					×						
3C_273	×		×					×						
TWA	×				×				×	×				
Per_OB2	×									×				
T_Tauri		×	×			×	×				×		×	
Y_Cygni			×	×		×						×	×	
V773_Tau		×	×			×	×				×		×	
Algol			×	×		×						×	×	
Mkn_421		×	×			×		×						×

TAB. 6.12 – Ajout de l'attribut `isOscillating` dans le contexte formel

	Attributs binaires								Classes SIMBAD						
	isOscillating	isIncluding	isReddening	isEmitting	isEclipsing	isExpanding	isIncluded	isFlaring	Galaxy	Asso._of_Young_Stars	Asso._of_Stars	T_Tau_type_Star	Eclipsing_Binary	Star	BL Lac
Cen_A		×		×					×						
3C_273		×		×					×						
TWA		×				×				×	×				
Per_OB2		×									×				
T_Tauri			×	×			×	×				×		×	
Y_Cygni	×			×	×		×						×	×	
V773_Tau			×	×			×	×				×		×	
Algol	×			×	×		×						×	×	
Mkn_421			×	×			×		×						×

6.4 Découverte d'unités de connaissances

Le processus PACTOLE a extrait des unités de connaissances dans les deux domaines d'application, l'astronomie et la microbiologie.

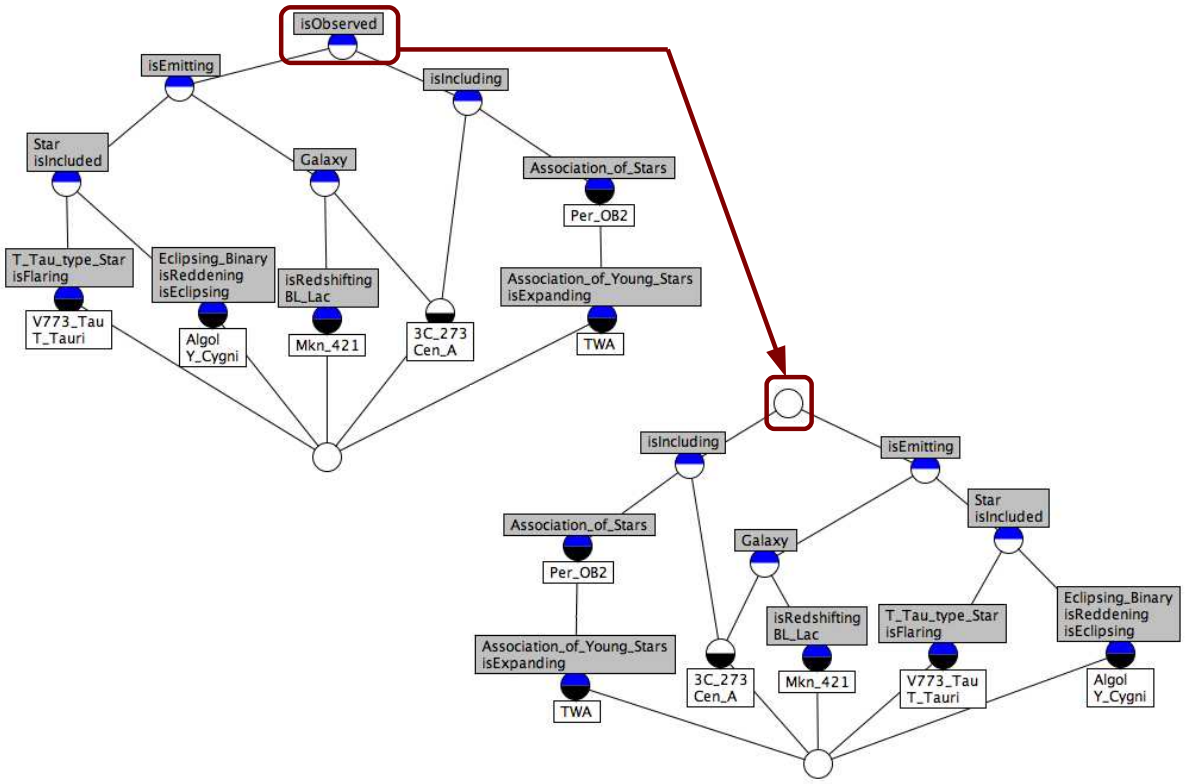


FIG. 6.9 – Treillis correspondant à la suppression de l'attribut `isObserved` à tous les objets du contexte formel

6.4.1 Découverte d'unités de connaissances dans le domaine de l'astronomie

Dans le domaine de l'astronomie, PACTOLE a permis d'enrichir la hiérarchie de la base SIMBAD. Ces unités de connaissances peuvent être divisées en trois types. Le premier type de connaissances est l'identification de nouveaux objets célestes (voir la sous-section 4.4.1 page 67) qui a servi aux astronomes pour enrichir les entrées de la base SIMBAD. Le deuxième type de connaissance est la mise en évidence de nouvelles corrélations entre les objets célestes et leurs attributs binaires (voir la sous-section 4.6.2 page 76). Enfin, le troisième type de connaissances recouvre la proposition de nouvelles classes dans la hiérarchie source de la base SIMBAD (voir 5.1.3 page 86). Ces deux derniers types de connaissances permettent aux astronomes de définir de nouvelles classes et ainsi d'enrichir la hiérarchie source de la base SIMBAD.

Mise en évidence de classes déjà connues. Certaines corrélations entre attributs et objets sont déjà connues dans le domaine de l'astronomie, mais l'extraction semi-automatique de ces corrélations est très importante pour les experts du domaine. Par exemple le concept (`{Algol, SAO_186497, HS_Her, TW_Cnc, V649_Cas, MM_Herculis, Y_Cygni}`, `{Star, Eclipsing_Star, isObserved, isEmitting, isEclipsing}`) définit la classe `{Eclipsing_Star}`

Proposition de nouvelles classes. Le concept (`{Orion, TWA}`, `{Association_of_stars, isExpanding, isObserved}`) a mené à la création d'une nouvelle classe « `Association_of_Young_Stars` ».

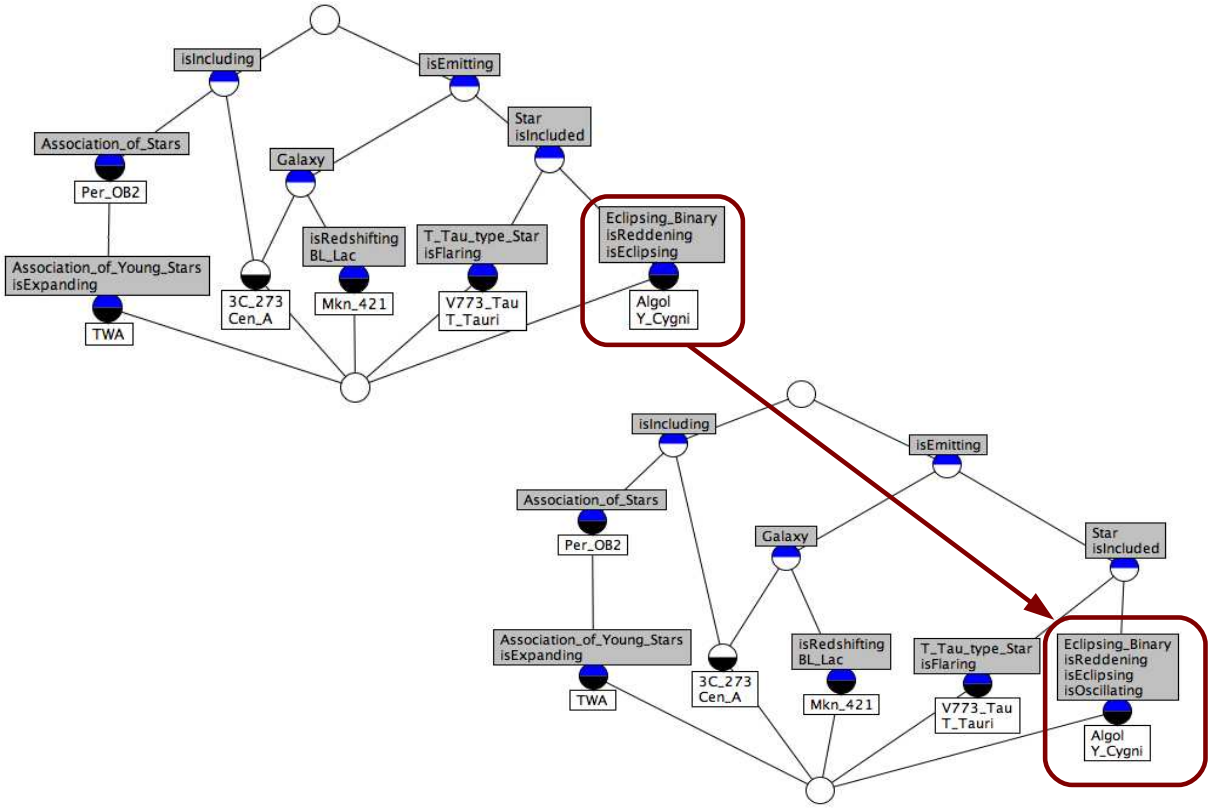


FIG. 6.10 – Treillis correspondant à l'ajout de attribut `isOscillating` dans le contexte formel

Changement de classe. Les corrélations entre les objets célestes et certains attributs permettent de changer la classe d'un objet. Par exemple, à partir du concept ({`AB_Dor, Algol, OJ_287, RS_CVn, T_Tauri, V773_Tau`}, {`isFlaring, isObserved`}). Le fait que l'objet `OJ_287` qui est affecté à la classe `BL_Lac` possède l'attribut `isFlaring` fait que cet objet n'est plus un `BL_Lac`, mais un `Noyau_de_galaxie`.

6.4.2 Découverte d'unités de connaissances dans le domaine de la microbiologie

Dans le domaine de la microbiologie comme pour celui de l'astronomie, les experts ont été intéressés par la définition des classes déjà existantes et la découverte de nouvelles classes, mais aussi par la mise en évidence des relations d'appartenance ou de résistance entre les gènes, les bactéries et les antibiotiques (voir la sous-section 5.2.5 page 94). Nous présentons dans cette partie quelques exemples des concepts qui ont été interprétés par les experts. Pour chaque exemple, les experts ont essayé de donner une interprétation et justifier pourquoi ils considéraient un concept comme étant intéressant.

Mise en évidence de classes déjà connues. Certaines interactions entre les classes de bactéries et les classes d'antibiotiques sont déjà connues. L'analyse relationnelle de concepts peut nous permettre de retrouver cette connaissance des experts de façon semi-automatique. Par exemple, le concept du treillis des bactéries `B17` ({`Brachyspira_Hyodysenteriae,`

Citrobacter_Freundii, Enterobacter_Aerogenes, Escherichia_Coli, Klebsiella_Oxytoca, Klebsiella_Pneumoniae, Neisseria_Gonorrhoeae, Pseudomonas_Aeruginosa, Rickettsia_Prowazekii, Salmonelle_Typhimurium, Serratia_Marcescens} {hasNegativeGram, isResisting:A7}) résiste au concept des antibiotiques A7 ({Isoniazid, Macrolide, Nalidixic, Pefloxacin, Rifampicin, Sparfloxacin, Sulfathiazole, Tetracycline, Triclosan, Vancomycin} {FRB2}). Ce phénomène de résistance peut s'expliquer par le fait que les bactéries à gram négatif (qui ne possèdent pas de parois) résistent aux antibiotiques *hydrophobes* (qui n'aiment pas l'eau).

Proposition de nouveaux attributs. Les microbiologistes ont trouvé certains concepts très hétérogènes. Par exemple, le concept ({Citrobacter_Freundii, Enterobacter_Aerogenes, Enterobacteriaceae, Escherichia_Coli, Pseudomonas_Aeruginosa, Salmonelle_Typhimurium, Serratia_Marcescens} {isResisting:A4, isResisting:A7,isSticks, hasNegativeGram, isHeterotrophic, isMobile}) est composé de deux familles distinctes. La première famille est composée des bactéries {Citrobacter_Freundii, Enterobacter_Aerogenes, Enterobacteriaceae, Escherichia_Coli, Salmonelle_Typhimurium, Serratia_Marcescens} qui sont toutes des Enterobacteriaceae. Et la seconde famille (la famille Pseudomonas) est composée de la bactérie {Pseudomonas_Aeruginosa}. Les microbiologistes ont proposé d'ajouter l'attribut Activity_Oxydase pour les distinguer. Le test d'Activity_Oxydase est un test standard qui détecte une enzyme possédée par certaines familles de bactéries. Dans notre cas, ce test est négatif pour la première famille et positif pour la seconde.

Chapitre 7

Conclusion et perspectives

7.1 Conclusion générale

Les travaux effectués dans le cadre de cette thèse ont porté sur la construction d'ontologies de domaine (ici l'astronomie et la microbiologie) à partir de ressources textuelles hétérogènes. Plus précisément, il s'agit de mettre en œuvre l'Analyse Formelle de Concepts (AFC) et son extension l'Analyse Relationnelle de Concepts (ARC). Ces deux processus permettent de construire des treillis de concepts à partir de tableaux binaires, d'objets et d'attributs pour l'AFC ; d'objets, d'attributs et de relations entre objets pour l'ARC. Ensuite, les treillis obtenus deviennent naturellement des supports pour la représentation des connaissances relatives aux domaines et aux ressources étudiés. Ces travaux ont abouti à la mise en place d'une méthodologie de construction d'ontologies originale nommée «Property And Class Characterization from Text to OntoLogy Enrichment» (PACTOLE). Cette approche est caractérisée par deux apports majeurs par rapport à l'état de l'art.

Le premier apport est relatif à la façon dont sont décrits les objets du domaine à partir desquels sera construite l'ontologie. En fonction des ressources les plus couramment disponibles (corpus de textes, bases de données ou thésaurus), trois types principaux de descripteurs d'objets sont considérés. Le premier type de descripteurs est constitué d'un ensemble prédéfini de classes affectées aux objets manuellement par les experts du domaine. Le deuxième type de descripteurs d'objets rassemble les attributs binaires décrivant des caractéristiques propres aux objets. Enfin, le troisième type de descripteurs d'objets se compose des liens inter-objets (ou attributs relationnels), c'est-à-dire des relations existantes entre les objets. Des méthodes d'extraction et d'analyse à partir des ressources du domaine ont été proposées pour chaque type de descripteurs. Ces méthodes utilisent des outils de Traitement Automatique de la Langue Naturelle (TALN) et d'Extraction d'Information (EI).

Le second apport de ce travail consiste à utiliser l'AFC, puis une de ses extensions, l'ARC, pour construire des hiérarchies de concepts qui serviront de schéma d'ontologie (schéma conceptuel). Une autre façon de considérer cette contribution consisterait à dire que le niveau terminologique de l'ontologie (TBOX) est dérivé à partir du niveau assertionnel (ABOX) dans une démarche ascendante typique d'un processus d'apprentissage inductif. L'AFC regroupe des objets partageant les mêmes attributs binaires dans des concepts d'un treillis. L'ARC est une extension de l'AFC qui permet de regrouper des objets partageant les mêmes attributs binaires, mais aussi les mêmes attributs relationnels. Les treillis finaux obtenus contiennent des concepts où cohabitent à la fois des attributs binaires et relationnels. De cette façon, des définitions étendues sont proposées aux experts du domaine pour être associées aux classes prédéfinies dans ce domaine ainsi que

de nouvelles classes inexistantes dans la hiérarchie initiale. Ces nouvelles classes peuvent être considérées pertinentes et ajoutées par les experts comme nouvelles «unités de connaissances». L'ensemble de ces éléments est ensuite représenté dans le cadre d'un langage de représentation des connaissances comme le langage $\mathcal{FLE}$ de la famille des logiques de descriptions, puis est implémenté en OWL (Web Ontology Language). Différentes questions auxquelles notre système répond automatiquement en expliquant les mécanismes de raisonnement ont également été proposées.

Des expériences pratiques ont été menées dans deux domaines d'application que sont l'astronomie et la microbiologie. Pour chaque domaine, nous avons proposé une évaluation des experts du domaine ainsi que des méthodes d'interaction avec eux. Une évaluation de l'apport de la méthodologie a aussi été présentée avec les connaissances extraites qui se sont avérées pertinentes d'après le jugement des experts.

7.2 Perspectives

Les perspectives de ce travail sont à la fois nombreuses et prometteuses. Tout d'abord l'application de notre méthodologie PACTOLE à d'autres domaines d'application peut permettre de définir de nouveaux descripteurs d'objets qui pourraient nécessiter l'utilisation d'autres extensions de l'AFC (présentés dans la sous-section 3.3.7). Dès lors, nous avons pu remarquer dans nos deux expérimentations sur l'astronomie et la microbiologie que les types de descripteurs utilisés pour construire l'ontologie du domaine dépendent non seulement des experts du domaine, mais aussi des ressources disponibles. Il est évident que nous pouvons dès maintenant nous intéresser à d'autres domaines scientifiques tels que la biochimie, l'archéologie, la géologie... pour affiner notre méthodologie et identifier de nouvelles connaissances sous-entendues dans les domaines concernés. En fait, la mise en œuvre d'une investigation équivalente à celle réalisée en astronomie et en microbiologie à plusieurs autres domaines scientifiques pourrait nous permettre d'affiner notre méthodologie. En effet, il existe de nombreuses autres ressources de descripteurs dont PACTOLE peut s'enrichir telles que : des mesures, des intervalles,... Or, pour prendre en compte ces nouveaux descripteurs, des extensions de l'AFC existent déjà.

Une autre perspective qui s'offre à nous concerne l'utilisation de l'ontologie résultant d'une exploitation de PACTOLE pour l'aide à la classification et la recherche d'information dans les textes. Nous pouvons alors nous intéresser à la manipulation des textes par leur contenu. Cette contribution peut permettre la classification des textes par non seulement, rapport aux termes qu'ils contiennent, mais aussi par rapport aux relations que ces termes entretiennent. De même, elle peut enrichir une recherche d'information par l'ajout aux requêtes des informations sur certaines relations entre concepts. Par exemple, dans le domaine de la microbiologie, un expert peut rechercher la bactérie `b` et l'antibiotique `a`, mais il peut aussi rechercher la bactérie `b` qui possède une relation de résistance avec l'antibiotique `a`. Cette perspective peut nous permettre de définir une boucle : des textes à l'ontologie et de l'ontologie aux textes.

Bibliographie

[Agrawal et Srikant, 1995] R. Agrawal et R. Srikant. Mining generalized association rules. In *21th International Conference on Very Large Data Bases (VLDB'95)*, pages 407–419, San Francisco, CA, USA, 1995. Kaufmann, M.

[Ananiadou et Mc Naught, 2005] S. Ananiadou et J. Mc Naught. *Text Mining for Biology And Biomedicine*. Artech House Publishers, 2005.

[Antoniou et van Harmelen, 2004] G. Antoniou et F. van Harmelen. *A Semantic Web Primer*. MIT Press, 2004.

[Arpírez *et al.*, 2003] J.C. Arpírez, O. Corcho, M. Fernández-López, et A. Gómez-Pérez. WebODE in a nutshell. *AI Mag.*, 24(3) :37–47, 2003.

[Aussenac-Gilles *et al.*, 2000] N. Aussenac-Gilles, B. Biébow, et S. Szulman. Revisiting ontology design : A method based on corpus analysis. In Dieng R. et O. Corby, editors, *12th International Conference in Knowlegde Engineering and knowledge Management (EKAW'00)*, volume 1937, pages 172–188, 2000.

[Baader *et al.*, 2003] F. Baader, D. Calvanese, D. de McGuinness, D. Nardi, et P. Patel-Schneider, editors. *The Description Logic Handbook. Theory, Implementation and Applications*. Cambridge University Press, 2003.

[Badra *et al.*, 2008] F. Badra, M. d'Aquin, J. Lieber, et T. Meilender. Edhibou : a customizable interface for decision support in a semantic portal. In C. Bizer et A. Joshi, editors, *Poster and Demonstration Session at the 7th International Semantic Web Conference (ISWC2008), Karlsruhe, Germany, October 28, 2008*. CEUR-WS.org, 2008.

[Barbut et Monjardet, 1970] M. Barbut et B. Monjardet. *Ordre et classification – Algèbre et combinatoire (2 tomes)*. Hachette, 1970.

[Barriere, 2002] C. Barriere. Investigating the causal relation in informative texts. *Terminology*, 7 :135–154, 2002.

[Bechhofer *et al.*, 2003] S. Bechhofer, R. Volz, et P. Lord. Cooking the Semantic Web with the OWL API. In *2nd International Semantic Web Conference (ISWC'03)*, pages 659–675, Sanibel Island (FL US), 2003. Springer.

[Belohlávek et Sklenar, 2005] R. Belohlávek et V. Sklenar. Formal Concept Analysis constrained by attribute-dependency formulas. In B. Ganter et R. Godin, editors, *International Conference on Formal Concept Analysis (ICFCA'05)*, pages 176–191. Springer, 2005.

[Bendaoud *et al.*, 2007a] R. Bendaoud, M. Rouane Hacene, Y. Toussaint, B. Delecroix, et A. Napoli. Construction d'une ontologie à partir d'un corpus de textes avec l'acf. In *18E Journées francophones d'Ingénierie des Connaissances (IC'07)*, Grenoble, France, 2007.

[Bendaoud *et al.*, 2007b] R. Bendaoud, M. Rouane Hacene, Y. Toussaint, B. Delecroix, et A. Napoli. Text-based ontology construction using relational concept analysis. In *International Workshop on Ontology Dynamics - (IWOD'07)*, pages 55–68, Innsbruck, Autriche, 2007.

[Bendaoud *et al.*, 2008a] R. Bendaoud, A. Napoli, et Y. Toussaint. Formal concept analysis : A unified framework for building and refining ontologies. In *16th International Conference on Knowledge Engineering and Knowledge Management Knowledge Patterns (EKAW'08)*, volume 5268, pages 156–171, Acitrezza, Catania, Italy, 2008. Springer.

[Bendaoud *et al.*, 2008b] R. Bendaoud, A. Napoli, et Y. Toussaint. Pactole : A methodology and a system for semi-automatically enriching an ontology from a collection of texts. In *16th International Conference on Conceptual Structures (ICCS'08) - Conceptual Structures : Knowledge Visualization and Reasoning*, volume 5113, pages 203–216, Toulouse, France, 2008. Springer.

[Bendaoud *et al.*, 2008c] R. Bendaoud, A. Napoli, et Y. Toussaint. A proposal for an interactive ontology design process based on formal concept analysis. In *5th International Conference on Formal Ontology in Information Systems (FOIS'08)*, pages 311–323, Saarbrücken, Germany, 2008.

[Bendaoud *et al.*, 2009] R. Bendaoud, Y. Toussaint, et A. Napoli. L'analyse formelle de concepts au service de la construction et l'enrichissement d'une ontologie. In *Revue des Nouvelles Technologies de l'Information. Deuxième numéro spécial sur la : Fouille de données complexes*, page 32 pages (à paraître). Cépaduès, 2009.

[Bernaras *et al.*, 1996] A. Bernaras, I. Laresgoiti, et J. Corera. Building and reusing ontologies for electrical network applications. In *12th European Conference on Artificial Intelligence (ECAI'96)*, pages 298–302, Budapest, Hungary, 1996.

[Berners-Lee, Mai 2001] T. Berners-Lee. The semantic Web. In *Scientific American Magazine*, Mai 2001.

[Birkhoff, 1967] Garrett Birkhoff. *Lattice Theory*, volume 25 of *ASM Colloquium Publications*. AMS, Providence, RI, 3rd edition, 1967.

[Bordat, 1986] J.P. Bordat. Calcul pratique du treillis de galois d'une correspondance. *Mathématiques et Sciences humaines*, 96 :31–47, 1986.

[Boudjlida, 1999] N. Boudjlida. *Bases de données relationnelles et systèmes d'informations : Langages, systèmes et méthodes.* Dunod, Paris, 1999.

[Bourigault, 1994] C. Bourigault. *LEXTER, Un logiciel d'Extraction de TERminologie. Application à l'acquisition de connaissances à partir de textes.* Thèse d'informatique, Ecole des Hautes Etudes en Sciences Sociales, Paris, 1994.

[Brachman et Anand, 1996] R.J. Brachman et T. Anand. The process of knowledge discovery in databases. *Advances in knowledge discovery and data mining*, pages 37–57, 1996.

[Breiman *et al.*, 1984] L. Breiman, J. Friedman, C.J. Stone, et R.A. Olshen, editors. *Classification and Regression Trees.* Wadsworth and Brooks, 1984.

[Brewster *et al.*, 2003] C. Brewster, F. Ciravegna, et Y. Wilks. Background and foreground knowledge in dynamic ontology construction. In *Proceedings of the Semantic Web Workshop, Toronto, August 2003*, 2003.

[Brickley et Guha, 2000] D. Brickley et R. Guha. Resource description framework (RDF) schema specification 1.0. Technical report, World Wide Web Consortium, 2000.

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

[Brito et Polaillon, 2005] P. Brito et G. Polaillon. Structuring probablilistic data by Galois lattices. *Mathématiques et Sciences humaines / Mathematics and Social Sciences*, 1(169) :77–104, 2005.

[Carpineto et Romano, 2000] C. Carpineto et G. Romano. Order-theoretical ranking. *Journal of the American Society for Information Sciences (JASIS'00)*, 51 :587–601, 2000.

[Carpineto et Romano, 2004] C. Carpineto et G. Romano. *Concept Data Analysis : Theory and Applications*. John Wiley & Sons, 2004.

[Charniak, 1995] E. Charniak. Natural language learning. *CSURV : Computing Surveys*, 27, 1995.

[Chein, 1969] M. Chein. Algorithme de recherche des sous-matrices premières d'une matrice. *Bull. Math. R.S. Roumanie*, 1969.

[Church, 1988] K.W. Church. A stochastic parts program and noun phrase parser for unrestricted text. In *Second Conference on Applied Natural Language Processing (ANLP'88)*, pages 136–143, 1988.

[Cimiano *et al.*, 2005] P. Cimiano, A. Hotho, et S. Staab. Learning concept hierarchies from text corpora using formal concept analysis. In *Journal of Artificial Intelligence Research (JAIR)*, volume Volume 24, pages 305–339, 2005.

[Cimiano, 2006] P. Cimiano. *Ontology Learning and Population from Text : Algorithms, Evaluation and Applications*. Springer, 2006.

[Codd, 1970] E.F. Codd. A relational model of data for large shared data banks. *Commun. ACM*, 13(6) :377–387, 1970.

[Condamines, 2005] A. Condamines. *Sémantique et Corpus*. Hermès, 2005.

[Cornuéjols A., 2003] Miclet L. Cornuéjols A. *Apprentissage artificiel - Concepts et algorithmes*. Eyrolles, Paris, 2003.

[Crane *et al.*, 2005] G. Crane, K. Bontcheva, J.A. Rydberg-Cox, et C.E. Wulfman. Emerging language technologies and the rediscovery of the past : a research agenda. *International Journal on Digital Libraries*, 5(4) :309–316, 2005.

[Crouch, 1988] C. Crouch. A cluster-based approach to thesaurus construction. In *11th International SIGIR Conference on Reseach and Development in Information Retrieval*, 1988.

[Curran et Moens, 2002] J.R. Curran et M. Moens. Improvements in automatic thesaurus extraction. In *Workshop of the ACL Special Interest Group on the Lexicon (SIGLEX)*, pages 59–66, Philadelphia, USA, 2002.

[Daelemans *et al.*, 1999] W. Daelemans, S. Buchholz, et J. Veenstra. Memory-based shallow parsing. In *CoNLL'99*, pages 53–60, Bergen, Norway, 1999.

[Dao *et al.*, 2004] M. Dao, M. Huchard, M. Hacene Rouane, C. Roume, et P. Valtchev. Improving generalization level in UML models : Iterative cross generalization in practice. In *Proceedings of the 12th International Conference on Conceptual Structures (ICCS'04)*, volume 3127, pages 346–360, Huntsville, AL, 2004. Springer.

[d'Aquin *et al.*, 2005] M. d'Aquin, J. Lieber, et A. Napoli. Decentralized case-based reasoning for the Semantic Web. In *International Semantic Web Conference (ISWC'05)*, pages 142–155, 2005.

[Davies *et al.*, 2005] J. Davies, A. Duke, N. Kings, D. Mladenic, K. Bontcheva, M. Grcar, R. Benjamins, J. Contreras, M.B. Civico, et T. Glover. Next Generation Knowledge Access. *Data and Knowledge Engineering*, 9(5) :64–84, 2005.

[de Marneffe *et al.*, 2006] M.C. de Marneffe, B. MacCartney, et C.D. Manning. Generating typed dependency parses from phrase structure parses. In *5th International conference on Language Resources and Evaluation (LREC'06)*, GENOA, ITALY, 2006.

[Diday *et al.*, 1982] E. Diday, J. Lemaire, J. Pouget, et F. Testu. *Éléments d'analyse de données.* Dunod, 1982.

[Dubois *et al.*, 1994] J. Dubois, L. Guespin, M. Giacomo, C Marcellesi, J.B. Marcellesi, et J.P. Mével. *Dictionnaire de linguistique et des sciences du langage.* Collection Trésors du Français, Larousse, 1994.

[Dunham, 2002] M.H. Dunham. *Data Mining : Introductory and Advanced Topics.* Prentice Hall, 2002.

[Faatz et Steinmetz, 2004] A. Faatz et R. Steinmetz. Ontology enrichment evaluation. In E. Motta, N. Shadbolt, A. Stutt, et N. Gibbins, editors, *14th International Conference on Engineering Knowledge in the Age of the Semantic Web (EKAW'04)*, volume 3257/2004, pages 497–498, Whittlebury Hall, UK, 2004. Springer.

[Faure et Nedellec, 1999] D. Faure et C. Nedellec. Knowledge acquisition of predicate argument structures from technical texts using machine learning : The system asium. In *11th International Conference in Knowledge Acquisition, Modeling and Management (EKAW'99)*, pages 329–334, Dagstuhl Castle, Germany, 1999. Springer.

[Fayyad *et al.*, 1996] U.M. Fayyad, G. Piatetsky-Shapiro, et P. Smyth. From data mining to knowledge discovery in databases. In *From Data Mining to Knowledge Discovery*, page chapitre1, 1996.

[Fayyad, 1996] U.M. Fayyad. Data mining and knowledge discovery in databases : Applications in astronomy and planetary science. In *13th National Conference on Artificial Intelligence and 8th Innovative Applications of Artificial Intelligence Conference (AAAI/IAAI'96)*, volume 2, pages 1590–1592, 1996.

[Feigenbaum, 1961] E.A. Feigenbaum. The simulation of verbal learning behavior. In *IRE-AIEE-ACM '61 (Western) : Papers presented at the May 9-11, 1961, western joint IRE-AIEE-ACM computer conference*, pages 121–132, New York, NY, USA, 1961. ACM.

[Feldman et Sanger, 2007] R. Feldman et J. Sanger. *The Text mining handbook.* Cambridge, 2007.

[Fensel *et al.*, 2002] D. Fensel, W. Wahlster, et H. Lieberman, editors. *Spinning the Semantic Web : Bringing the World Wide Web to Its Full Potential.* MIT Press, Cambridge, MA, USA, 2002.

[Ferré, 2002] S. Ferré. *Systèmes d'information logiques : un paradigme logico-contextuel pour interroger, naviguer et apprendre.* Thèse d'université, Université de Rennes 1, Octobre 2002.

[Finin *et al.*, 1994] T. Finin, R. Fritzson, D. McKay, et R. McEntire. Kqml as an agent communication language. In Adam N., Bhargava B., et Yesha Y., editors, *3rd International Conference on Information and Knowledge Management (CIKM'94).* ACM Press, 1994.

[Fisher, 1987] D.H. Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2(2) :139–172, 1987.

[Ganter et Kuznetsov, 2001] B. Ganter et S.O. Kuznetsov. Pattern structures and their projections. In H.S. Delugach et G. Stumme, editors, *Conceptual Structures : Broadening the Base, 9th International Conference on Conceptual Structures (ICCS'01). Stanford, CA, USA, July 30-August 3, 2001*, pages 129–142. Springer, 2001.

[Ganter et Wille, 1999] B. Ganter et R. Wille. *Formal Concept Analysis, Mathematical Foundations.* Springer, 1999.

[Ganter, 1984] B. Ganter. Two basic algorithms in concept analysis. Technical report, Preprint 831, Technische Hochschule Darmstadt, 1984.

[Garcia, 1998] D. Garcia. *Analyse automatique des textes pour l'organisation causale des actions. Réalisation du système informatique COATIS.* Thèse d'informatique, Université de Paris-Sorbonne, 1998.

[Gennari *et al.*, 1989] J.H. Gennari, P. Langley, et D. Fisher. Models of incremental concept formation. *Artificial Intelligence*, 40(1-3) :11–61, 1989.

[Godin *et al.*, 1991] R. Godin, R. Missaoui, et H. Alaoui. Learning algorithms using a Galois lattice structure. In *3rd International Conference on Tools for Artificial Intelligence*, pages 22–29, 1991.

[Godin, 2000] R. Godin. *Systèmes de gestion de bases de données*, volume I : Fichiers et bases de données relationnelles (SQL). Loze-Dion, 2000.

[Gómez-pérez *et al.*, 2004] A. Gómez-pérez, M. Fernández-Lòpez, et O. Corcho. *Ontological Engineering.* Springer, 2004.

[Goujon, 1999] B. Goujon. Extraction d'informations techniques pour la veille par l'exploitation de notions indépendantes d'un domaine. *Terminologies nouvelles*, 19 :33–42, 1999.

[Grefenstette, 1994] G. Grefenstette. *Explorations in Automatic Thesaurus Discovery.* Kluwer Academic Publishers, Norwell, MA, USA, 1994.

[Gruber, 1993] T.R. Gruber. Towards principles for the design of ontologies used for knowledge sharing. In *Formal Ontology in Conceptual Analysis and Knowledge Representation*, pages 97–112. Kluwer Academic, Netherlands, 1993.

[Grüninger et Fox, 1995] M. Grüninger et M. Fox. Methodology for the design and evaluation of ontologies. In *Workshop on Basic Ontological Issues in Knowledge Sharing, (IJCAI'95), April 13, 1995*, 1995.

[Guyon et Elisseeff, 2003] I. Guyon et A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3 :1157–1182, 2003.

[Guénoche, 1990] A. Guénoche. Construction du treillis de Galois d'une relation binaire. *Revue Math. Inf. Sci. Hum, 109*, pages 41–53, 1990.

[Haarslev et Möller, 2001] V. Haarslev et R. Möller. Description of the racer system and its applications. In *International Workshop on Description Logics (DL'01), 1.-3. August*, pages 131–141, Stanford, USA, 2001.

[Haav, 2004] H.M. Haav. A semi-automatic method to ontology design by using FCA. In *3th International Conference on Concept Lattices and their Application (CLA'04)*. CEUR-WS.org, 2004.

[Habert et Nazarenko, 1996] B. Habert et A. Nazarenko. La syntaxe comme marche-pied de l'acquisition des connaissances : Bilan critique d'une expérience. In *Actes des septièmes Journées Acquisition des Connaissances (JAC'96)*, pages 137–148, Sète, 1996.

[Hahn et Schulz, 2004] U. Hahn et S. Schulz. Building a very large ontology from medical thesauri. In S. Staab et R. Studer, editors, *Handbook on ontologies*, pages 133–150. Springer, 2004.

[Han et Kamber, 2001] J. Han et M. Kamber. *Data Mining : Concepts and Techniques.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001.

[Harris, 1968] Z. Harris. *Mathematical Structure of Language.* Wiley, 1968.

[Hearst, 1992] M.A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *14th International Conference on Computational Linguistics (COLING'92)*, 1992.

[Horrocks *et al.*, 2004] I. Horrocks, P. Patel-Schneider, et F. van Harmelen. From SHIQ and RDF to OWL : The making of a web ontology language. *Journal of Web Semantics*, 1(1) :7–26, 2004.

[Horrocks et Patel-Schneider, 1998] I. Horrocks et P.F. Patel-Schneider. Optimising propositional modal satisfiability for description logic subsumption. In *In Proceedings of AISC-98*, pages 234–246. Springer, 1998.

[Horrocks, 1998] I.R. Horrocks. Using an expressive description logic : Fact or fiction. In *Principles of Knowledge Representation and Reasoning : 6th International Conference (KR'98)*, pages 636–647. Morgan Kaufmann, 1998.

[ISO, 1989] ISO. Database language sql with integrity enhancement. Iso-9075-1989(e), International Standard Organization, 1989.

[ISO, April 1990] ISO. Database language sql2 and sql3. Iso working draft, International Standard Organization, April 1990.

[Jacquemin, 1997] C. Jacquemin. *Variation terminologique : Reconnaissance et acquisition automatiques de termes et de leurs variantes en corpus.* Mémoire d'habilitation à diriger des recherches en informatique, Université de Nantes, 1997.

[Jaoua et Elloumi, 2002] A. Jaoua et S. Elloumi. Galois connection, formal concepts and Galois lattice in real relations : application in a real classifier. *Journal of Systems and Software*, 60(2) :149–163, 2002.

[Jouis, 1993] C. Jouis. *Contributions à la conceptualisation et à la modélisation des connaissances à partir d'une analyse linguistique de textes. Réalisation d'un prototype : le système SEEK.* Thèse d'informatique, Ecole des Hautes Etudes en Sciences Sociales, Paris, 1993.

[Jurafsky et Martin, 2000] D. Jurafsky et J.H. Martin. *Speech and language processing.* Prentice Hall, 2000.

[Kayser, 1997] D. Kayser. *La réprésentation des connaissances.* Hermes, 1997.

[Klein et Manning, 2002] D. Klein et C.D. Manning. Fast exact inference with a factored model for natural language parsing. In *In Advances in Neural Information Processing Systems*, pages 3–10. MIT Press, 2002.

[Kohavi et John, 1997] R. Kohavi et G.H. John. Wrappers for feature subset selection. *Artif. Intell.*, 97(1–2) :273–324, 1997.

[Kuznetsov et Obiedkov, 2002] S.O. Kuznetsov et S.A. Obiedkov. Comparing performance of algorithms for generating concept lattices. *Journal of Experimental & Theoretical Artificial Intelligence*, 14(2-3) :189–216, 2002.

[Lager, 1998] T. Lager. Logic for part-of-speech tagging and shallow parsing. In *(NODALIDA'98)*, Copenhagen, Danmark, 1998.

[Lame, 2002] G. Lame. *Construction d'ontologie à partir de texte, une ontologie du droit dédiée à la recherche d'information sur le Web.* Thèse d'informatique, Ecole des Mines de Paris, 2002.

[Langley, 1996] P. Langley. *Elements of machine learning.* Kaufmann, M., 1996.

[Lassila, 1998] O. Lassila. Web metadata : A matter of semantics. In *IEEE Internet Computing*, volume 2(4), 1998.

[Lebart et Salem, 1994] L. Lebart et A. Salem. *Statistique textuelle.* Dunod, 1994.

[Lenat et Guha, 1990] D.B. Lenat et R.V. Guha. *Building a large Knowledge-based System : Representation and inference in the Cyc Project.* Addison-Wesley Longman, Boston, Massachusetts,USA, 1990.

[Levy et Manning, 2004] R. Levy et C.D. Manning. Deep dependencies from context-free statistical parsers : Correcting the surface dependency approximation. In *Meeting of the Association for Computational Linguistics (ACL'04)*, pages 327–334, 2004.

[Maedche et Staab, 2000] A. Maedche et S. Staab. Discovering conceptual relation from text. In *14th European Conference on Artifical Intelligence (ECAI'00)*, pages 321–325, Berlin, Germany, 2000.

[Maedche et Staab, 2004] E. Maedche et S. Staab. Ontology learning. In *Handbook on Ontologies*, pages 173–189. Springer, 2004.

[McCray et Nelson, 1995] A.T. McCray et S.J. Nelson. The representation of meaning in the UMLS. *Methods of Information in Medecine*, 34(1/2) :193–201, 1995.

[Messai *et al.*, 2008] N. Messai, M.D. Devignes, A. Napoli, et M. Smaïl-Tabbone. Many-valued concept lattices for conceptual clustering and information retrieval. In M. Ghallab, C.D. Spyropoulos, N. Fakotakis, et N. Avouris, editors, *18th biennial European Conference on Artificial Intelligence, (ECAI'2008), 21-25 July,*, volume 178, pages 127–131, Patras, Greece, 2008. IOS Press.

[Morin et Martienne, 2000] E. Morin et E. Martienne. Using a symbolic machine learning tool to refine lexico-syntactic patterns. In R.L. de Mántaras et E. Plaza, editors, *11th European Conference on Machine Learning (ECML'00)*, pages 292–299. Springer, 2000.

[Napoli, 1997] A. Napoli. Une introduction aux logiques de descriptions. Technical report, Rapport de recherche INRIA n°3314, 1997.

[Nirenburg et Raskin, 2004] S. Nirenburg et V. Raskin. *Ontological Semantics (Language, Speech, and Communication)*. The MIT Press, 2004.

[Norris, 1978] E.M. Norris. An algorithm for computing the maximal rectangles in a binary relation. *Revue Roumaine de Mathématiques Pures et Appliquées*, 23(2) :243–250, 1978.

[Noy *et al.*, 2001] N.F. Noy, M. Sintek, S. Decker, M. Crubézy, R.W. Fergerson, et M.A. Musen. Creating semantic web contents with. In *Protégé-2000. IEEE Intelligent Systems (2001*, volume 16(2), pages 60–71, 2001.

[Noy et McGuinness, 2001] N.F. Noy et D.L. McGuinness. Ontology development 101 : A guide to creating your first ontology. Technical report, Stanford Knowledge Systems Laboratory, Stanford University, USA, 2001.

[Paolucci *et al.*, 2002] M. Paolucci, T. Kawamura, T.R. Payne, et K. Sycara. Semantic matching of Web services capabilities. In I. Horrocks et J. Hendler, editors, *International Semantic Web Conference (ISWC'02)*, volume 2342. Springer, 2002.

[Polaillon, 1998] G. Polaillon. *Organisation et interprétation par les treillis de Galois de données de type multivalué, intervalle ou histogramme.* Thèse de doctorat en informatique, Université Paris IX-Dauphine, Décembre 1998.

[Punyakanok et Roth, 2000] V. Punyakanok et D. Roth. The use of classifiers in sequential inference. In T.K. Leen, T.G. Dietterich, et V. Tresp, editors, *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS'00)*, pages 995–1001, Denver, CO, USA, 2000. MIT Press.

[Quinlan, 1986] J.R. Quinlan. Induction of Decision Trees. *Machine Learning*, 1(1) :81–106, 1986.

[Quinlan, 1988] J.R. Quinlan. Decision trees and multi-valued attributes. In J.E. Hayes, D. Michie, et J. Richards, editors, *Machine Intelligence (Volume 11 : Logic and the acquisition of knowledge)*, pages 305–318. Clarendon Press, Oxford, England, 1988.

[Rouane-Hacene *et al.*, 2007] A. Rouane-Hacene, M. Huchard, A. Napoli, et P. Valtchev. Proposal for combining formal concept analysis and description logics for mining relational data. In *Int. Conference on Formal Concept Analysis, ICFCA 2007, Clermont-Ferrand, France*, pages 51–65. Springer, 2007.

[Rouane, Septembre 2006] M. Rouane. *Etude de l'analyse formelle dans les données relationnelles. Application à la restructuration des modèles structuraux UML.* Thèse d'informatique, Université de Montréal, Faculté des études superieurs, Septembre, 2006.

[Rousselot *et al.*, 1996] F. Rousselot, P. Frath, et R. Oueslati. Extracting concepts and relations from corpora. In *Proceedings of ECAI Workshop on Corpus-Orientated Semantic analysis*, Budapest, 1996.

[Russell et Norvig, 2003] J. Russell et P. Norvig. *Artificial Intelligence : A Modern Approach.* Pearson Education, 2003.

[Saeys *et al.*, 2007] Y. Saeys, I. Inza, et P. Larranaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19) :2507–2517, 2007.

[Salton et Buckley, 1988] G. Salton et C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5) :513–523, 1988.

[Schreiber *et al.*, 1999] G. Schreiber, H. Akkermans, A. Anjewierden, R. Dehoog, N. Shadbolt, W. Vandevelde, et B. Wielinga. *Knowledge Engineering and Management : The CommonKADS Methodology.* MIT Press, Cambridge, Massachusetts,USA, 1999.

[Simon, 2000] A. Simon. *Outils classificatoires par objets pour l'extraction de connaissances dans les bases de données.* Thèse d'informatique, Université Henri Poincaré, 2000.

[Sinclair, 1996] J. Sinclair. Preliminary recommendations on corpus typology. Technical report, Expert Advisory Group on Language Engineering Standards document EAG-TCWG-CTYP/P, 1996.

[Sirin *et al.*, 2003] E. Sirin, J. Hendler, et B. Parsia. Semi-automatic composition of Web services using semantic descriptions. In *Web Services : Modeling, Architecture and Infrastructure workshop in ICEIS 2003*, pages 17–24, 2003.

[Sowa, 2000] J.F. Sowa. *Knowledge Representation : Logical, Philosophical, and Computational Foundations.* Brooks/Cole, 2000.

[Staab *et al.*, 2001] S. Staab, R. Studer, H.P. Schnurr, et Y. Sure. Knowledge processes and ontologies. *IEEE Intelligent Systems*, 16(1) :26–34, 2001.

[Staab et Maedche, 2001] S. Staab et A. Maedche. Knowledge portals : Ontologies at work. *AI Magazine*, 22(2) :63–75, 2001.

[Studer *et al.*, 1998] R. Studer, V.R. Benjamins, et D. Fensel. Knowledge engineering : Principles and methods. *Data Knowledge Engineering*, 25(1-2) :161–197, 1998.

[Stumme *et al.*, 1998] G. Stumme, R. Wille, et U. Wille. Conceptual knowledge discovery in databases using formal concept analysis methods. In *Second European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD'98)*, pages 450–458, London, UK, 1998. Springer-Verlag.

[Stumme *et al.*, 2002] G. Stumme, R. Taouil, Y. Bastide, N. Pasquier, et L. Lakhal. Computing iceberg concept lattices with t. *Data and Knowledge Engineering*, 42(2) :189–222, 2002.

[Stumme et Maedche, 2001] G. Stumme et A. Maedche. FCA-merge : Bottom-up merging of ontologies. In *International Joint Conference on Artificial Intelligence (IJCAI'01)*, pages 225–234, 2001.

[Sure *et al.*, 2002] Y. Sure, M. Erdmann, J. Angele, S. Staab, R. Studer, et D. Wenke. Ontoedit : Collaborative ontology development for the semantic web. In *International Semantic Web Conference*, pages 221–235. Springer, 2002.

[Taboada *et al.*, 2000] M. Taboada, J. Des, M. Arguello, J. Mira, et D. Martínez. A medical ontology for integrating case-based reasoning, rule-based reasoning, and patient databases. In *EUROCAST '99 : Proceedings on Computer Aided Systems Theory*, pages 521–527, London, UK, 2000. Springer-Verlag.

[Toussaint, 2004] Y. Toussaint. Extraction de connaissances à partir de textes structuré. In *Document Numérique*, volume 8, pages 1–24, 2004.

[Toutanova et Manning, 2000] K. Toutanova et C.D. Manning. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *The Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000)*, pages 63–70, 2000.

[Tzoukermann *et al.*, 1997] E. Tzoukermann, J. Klavans, et C. Jacquemin. Effective use of natural language processing techniques for automatic conflation of multi-word terms : the role of derivational morphology, part of speech tagging, and shallow parsing. In *20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'97)*, pages 148–155, 1997.

[Uschold et King, 1996] M. Uschold et M. King. Ontologies : Principles, methods and applications. *Knowledge Engineering Reviews*, 1996.

[Valente *et al.*, 1999] A. Valente, T. Russ, R. MacGregor, et W. Swartout. Building and (re)using an ontology of air campaign planning. *IEEE Intelligent Systems*, 14(1) :27–36, 1999.

[Valtchev *et al.*, 2003] P. Valtchev, D. Grosser, C. Roume, et M. Rouane Hacene. Galicia : an open platform for lattices. In *In Using Conceptual Structures : Contributions to the 11th Intl. Conference on Conceptual Structures (ICCS'03*, pages 241–254. Springer, 2003.

[Valtchev et Missaoui, 2001] P. Valtchev et R. Missaoui. Building concept (Galois) lattices from parts : generalizing the incremental methods. In *Proceedings of the ICCS ?01*, pages 290–303. Springer, 2001.

[Wille, 2002] R. Wille. Why can concept lattices support knowledge discovery in databases ? *Journal of Theoretical Artificial Intelligence*, 14(2–3) :81–92, 2002.

[WOWG, Février 2004a] WOWG. Owl web ontology language guide. W3c recommendation, W3C Web Ontology Working Group (WOWG), Février 2004.

[WOWG, Février 2004b] WOWG. Owl web ontology language overview. W3c recommendation, W3C Web Ontology Working Group (WOWG), Février 2004.

[Wriggers *et al.*, 2007] P. Wriggers, M. Siplivaya, I. Joukova, et R. Slivin. Intelligent support of engineering analysis using ontology and case-based reasoning. *Eng. Appl. Artif. Intell.*, 20(5) :709–720, 2007.

[Yu et Liu, 2004] L. Yu et H. Liu. Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research*, 5 :1205–1224, 2004.

Annexe A

Annexe-Sparql

A.1 Description d'un concept en OWL dans le domaine de la microbiologie

```
<owl:Class rdf:about="#B5">
      <rdfs:subClassOf rdf:resource="#B1"/>
      <rdfs:subClassOf rdf:resource="#B3"/>
      <rdfs:subClassOf rdf:resource="#GammaProteobacteria"/>
      <rdfs:subClassOf>
         <owl:Restriction>
            <owl:onProperty rdf:resource="#isResisting"/>
            <owl:someValuesFrom rdf:resource="#A2"/>
         </owl:Restriction>
      </rdfs:subClassOf>
      <rdfs:subClassOf>
         <owl:Restriction>
            <owl:onProperty rdf:resource="#isAnaerobic"/>
            <owl:someValuesFrom rdf:resource="&owl;Thing"/>
         </owl:Restriction>
      </rdfs:subClassOf>
   </owl:Class>
```

FIG. A.1 – Description du concept `B5` en OWL dans le domaine de la microbiologie

A.2 Réponses à la requête d'instanciation en astrologie

```
<?xml version="1.0"?>
<sparql
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:xs="http://www.w3.org/2001/XMLSchema#"
   xmlns="http://www.w3.org/2005/sparql-results#" >
 <head>
   <variable name="a"/>
   <variable name="b"/>
   <variable name="c"/>
 </head>
 <results><result>
    <binding name="a">
     <uri>http://www.semanticweb.org/ontologies/2009/1/OntoAstro4.owl#C0</uri>
    </binding>
    <binding name="b">
     <uri>http://www.semanticweb.org/ontologies/2009/1/OntoAstro4.owl#C6</uri>
    </binding>
    <binding name="c">
     <uri>http://www.semanticweb.org/ontologies/2009/1/OntoAstro4.owl#C6</uri>
    </binding>
  </result>
  <result>
    <binding name="a">
     <uri>http://www.semanticweb.org/ontologies/2009/1/OntoAstro4.owl#C0</uri>
    </binding>
    <binding name="b">
     <uri>http://www.semanticweb.org/ontologies/2009/1/OntoAstro4.owl#C6</uri>
    </binding>
    <binding name="c">
     <uri>http://www.w3.org/2002/07/owl#Nothing</uri>
    </binding>
  </result>
 </results></sparql>
```

FIG. A.2 – Réponses à la requête SPARQL d'instanciation de l'objet `Angel` en astrologie

A.3 Réponses à la requête d'instanciation en microbiologie

```
<?xml version="1.0"?>
<sparql
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:xs="http://www.w3.org/2001/XMLSchema#"
   xmlns="http://www.w3.org/2005/sparql-results#" >
 <head>
   <variable name="a"/>
   <variable name="b"/>
   <variable name="c"/>
   <variable name="d"/>
 </head>
 <results><result>
    <binding name="a">

<uri>http://www.semanticweb.org/ontologies/2008/11/MicroBio1.owl#B6</uri>
    </binding>
    <binding name="b">

<uri>http://www.semanticweb.org/ontologies/2008/11/MicroBio1.owl#B2</uri>
    </binding>
    <binding name="c">

<uri>http://www.semanticweb.org/ontologies/2008/11/MicroBio1.owl#B3</uri>
    </binding>
    <binding name="d">

<uri>http://www.semanticweb.org/ontologies/2008/11/MicroBio1.owl#B6</uri>
    </binding>
   </result>
 </results></sparql>
```

FIG. A.3 – Réponses à la requête SPARQL d'instanciation de l'objet `Staphylococcus_Aureus` dans le domaine de la microbiologie

A.4 Réponses à la requête de détection de domaine d'une relation en microbiologie

```
<?xml version="1.0"?>
<sparql
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:xs="http://www.w3.org/2001/XMLSchema#"
   xmlns="http://www.w3.org/2005/sparql-results#" >
  <head>
    <variable name="a"/>
    <variable name="b"/>
    <variable name="c"/>
    <variable name="d"/>
    <variable name="e"/>
    <variable name="y"/>
  </head>
  <results><result>
      <binding name="a">
        <uri>http://www.semanticweb.org/ontologies/2008/11/MicroBio1.owl#A2</uri>
      </binding>
      <binding name="b">
        <uri>http://www.semanticweb.org/ontologies/2008/11/MicroBio1.owl#A2</uri>
      </binding>
      <binding name="c">
        <uri>http://www.semanticweb.org/ontologies/2008/11/MicroBio1.owl#A2</uri>
      </binding>
      <binding name="d">
        <uri>http://www.semanticweb.org/ontologies/2008/11/MicroBio1.owl#A2</uri>
      </binding>
      <binding name="e">
        <uri>http://www.semanticweb.org/ontologies/2008/11/MicroBio1.owl#B5</uri>
      </binding>
      <binding name="y">
        <uri>http://www.semanticweb.org/ontologies/2008/11/MicroBio1.owl#Klebsiella_Oxytoca</uri>
      </binding>
    </result>
```

FIG. A.4 – Première partie des réponses à la requête SPARQL de détection de domaine d'une relation dont le co-domaine est le concept de l'objet `Norfloxacin` dans le domaine de la microbiologie

```
<result>
    <binding name="a">
      <uri>http://www.semanticweb.org/ontologies/2008/11/MicroBio1.owl#A2</uri>
    </binding>
    <binding name="b">
      <uri>http://www.semanticweb.org/ontologies/2008/11/MicroBio1.owl#A2</uri>
    </binding>
    <binding name="c">
      <uri>http://www.semanticweb.org/ontologies/2008/11/MicroBio1.owl#A2</uri>
    </binding>
    <binding name="d">
      <uri>http://www.semanticweb.org/ontologies/2008/11/MicroBio1.owl#A2</uri>
    </binding>
    <binding name="e">
      <uri>http://www.semanticweb.org/ontologies/2008/11/MicroBio1.owl#B5</uri>
    </binding>
    <binding name="y">

<uri>http://www.semanticweb.org/ontologies/2008/11/MicroBio1.owl#Klebsiella_Pneumoniae
</uri>
    </binding>
  </result>
 </results></sparql>
```

FIG. A.5 – Seconde partie des réponses à la requête SPARQL de détection de domaine d'une relation dont le co-domaine est le concept de l'objet `Norfloxacin` dans le domaine de la microbiologie

Annexe B

Annexe-Treillis

B.1 Présentation du treillis de concepts global de l'astronomie

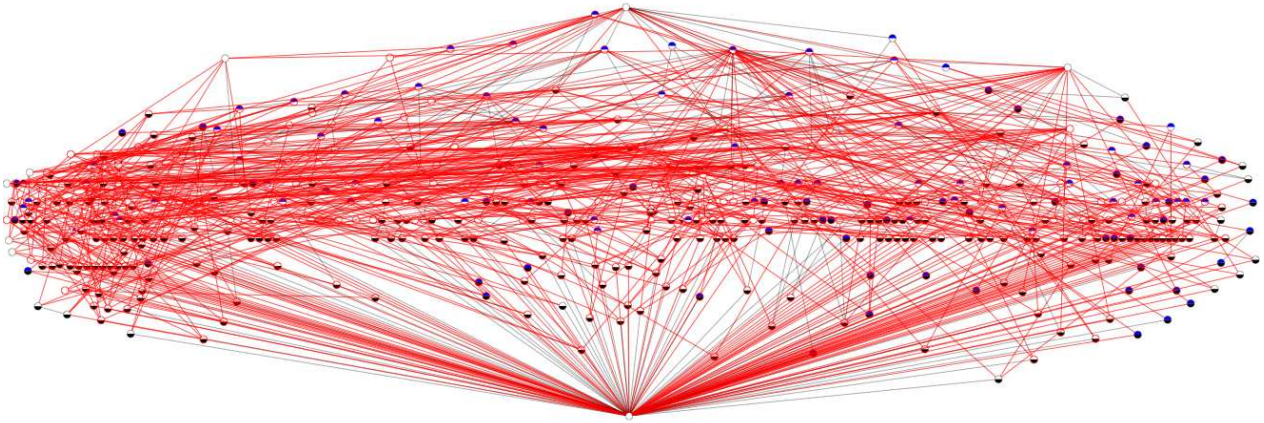


FIG. B.1 – Treillis complet du contexte des objets célestes $\mathbb{K}_O = (G_O, M_O, I_O)$ avec le logiciel ConExp

B.2 Présentation du treillis de concepts complet des bactéries

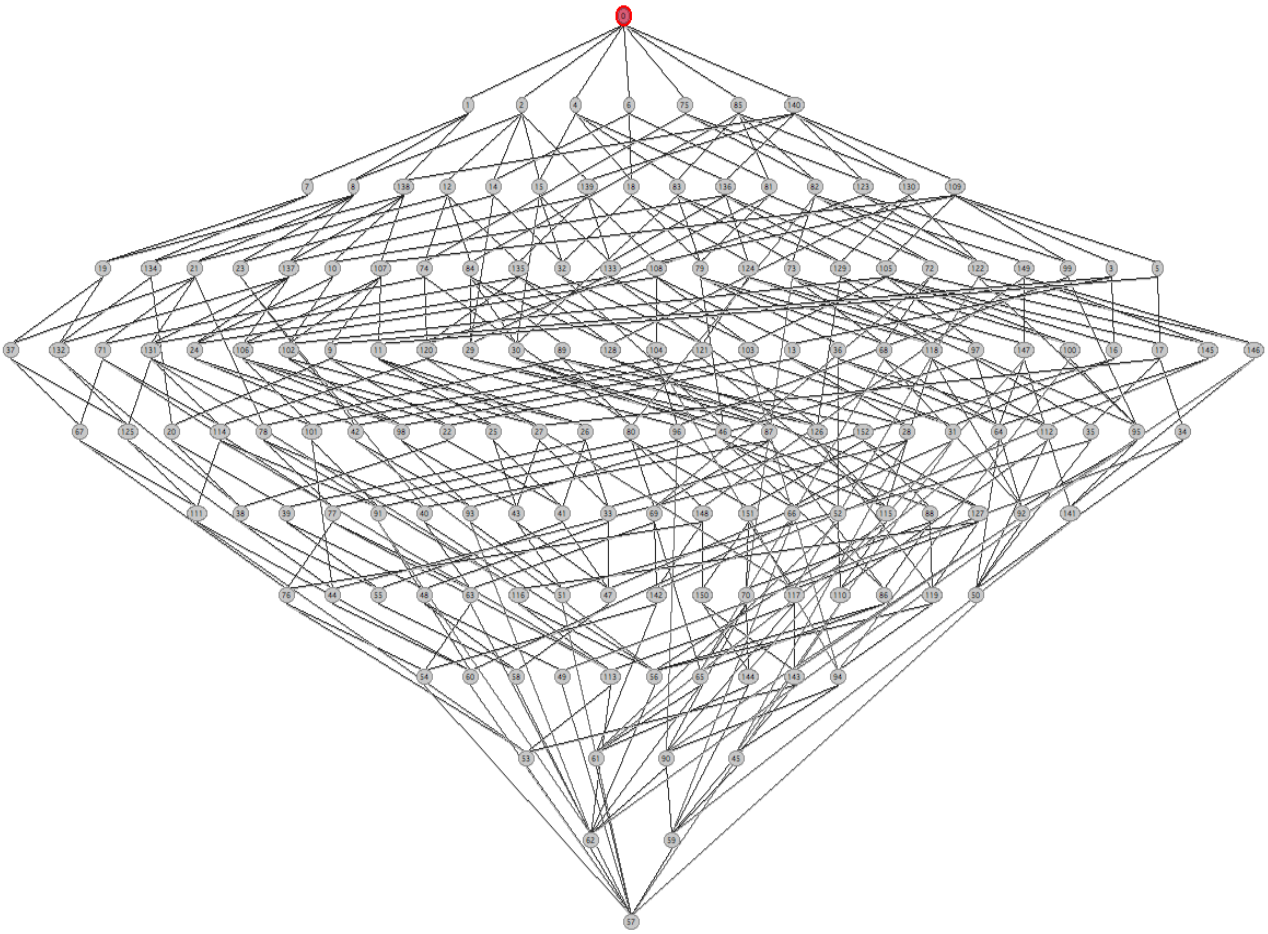


FIG. B.2 – Treillis complet du contexte des bactéries $\mathbb{K}_B = (G_B, M_B, I_B)$ avec le logiciel Galicia

B.3 Présentation du treillis de concepts complet des antibiotiques

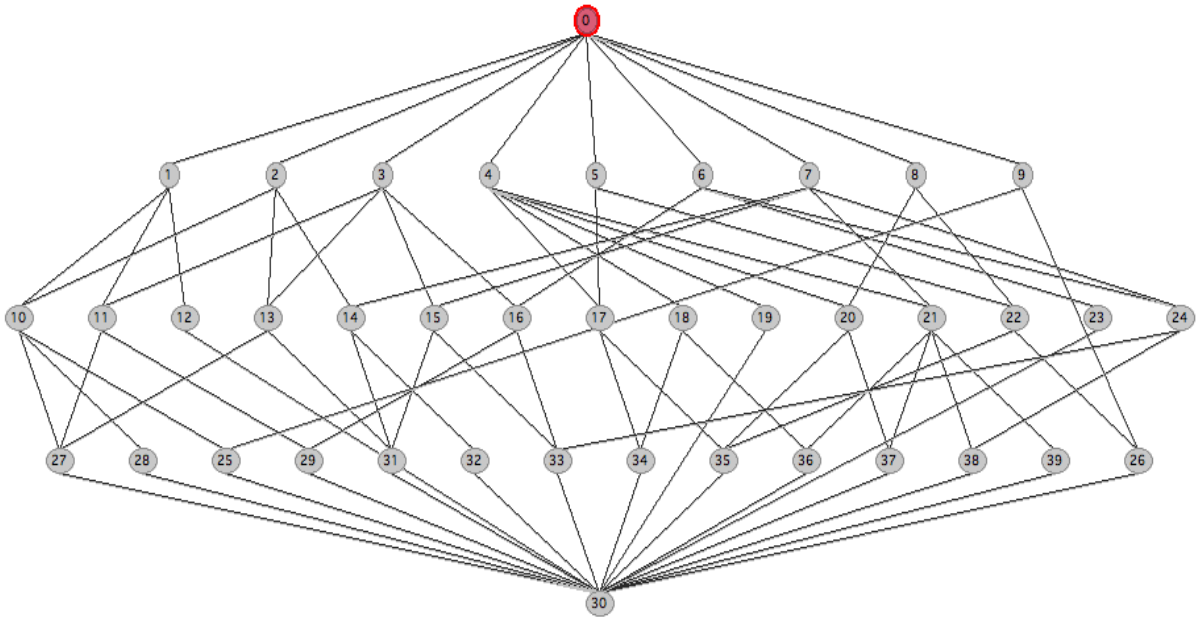


FIG. B.3 – Treillis complet du contexte des antibiotiques $\mathbb{K}_A = (G_A, M_A, I_A)$ avec le logiciel Galicia

B.4 Présentation du treillis de concepts complet des gènes

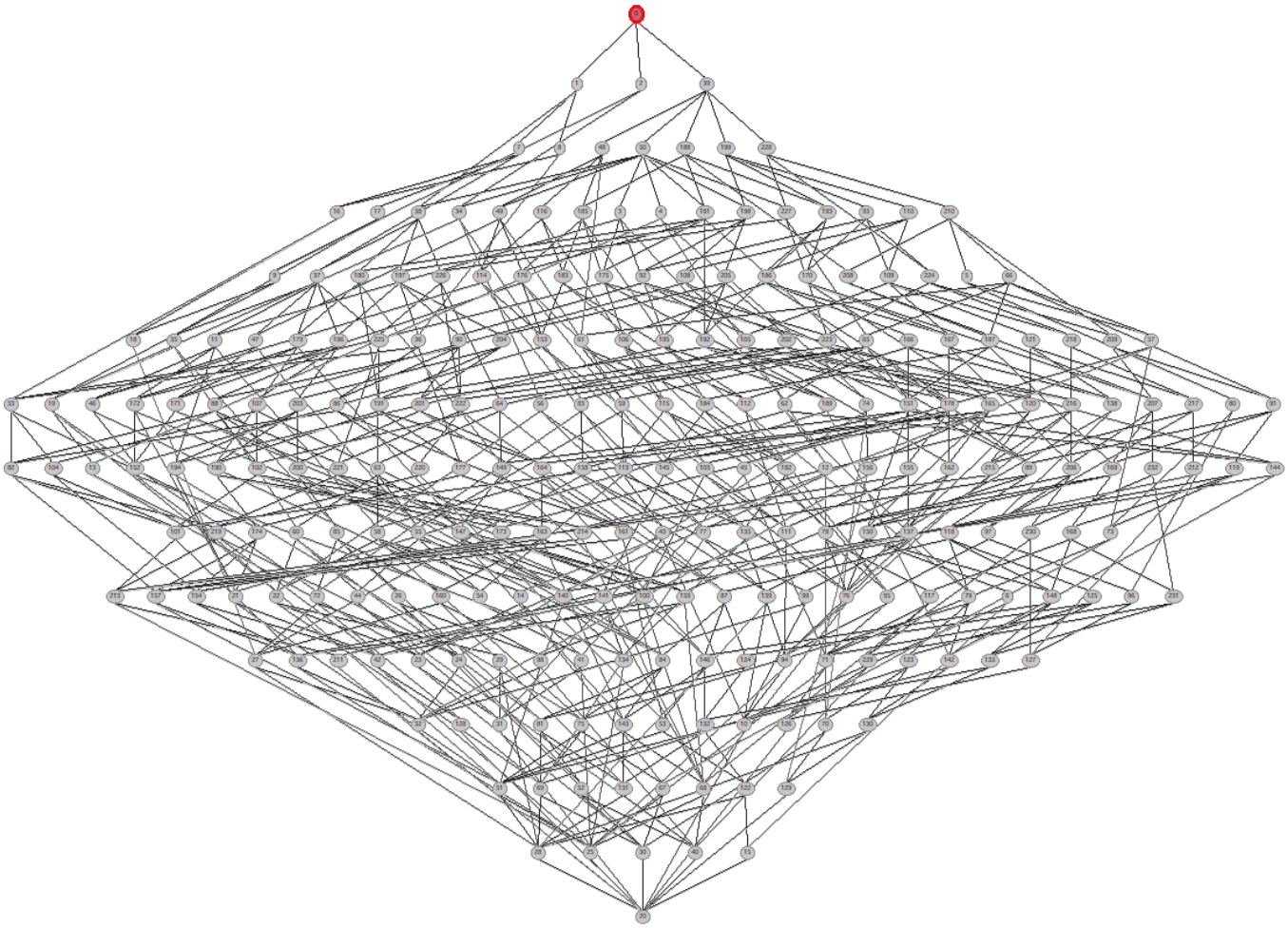


FIG. B.4 – Treillis complet du contexte des gènes $\mathbb{K}_G = (G_G, M_G, I_G)$ avec le logiciel Galicia

Résumé

Les ontologies sont diversement employées notamment dans les domaines du Web sémantique, de l'ingénierie des connaissances,... En effet, elles permettent de partager, de diffuser et d'actualiser les connaissances d'un domaine. Afin de construire ces ontologies, notre méthodologie utilise tout d'abord des méthodes de Traitement Automatique de la Langue Naturelle (TALN) et d'Extraction d'Information (EI) pour extraire des données préparées à partir de chaque ressource du domaine (corpus de textes, bases de données, thesaurus). Puis, ces données sont fouillées avec les méthodes de fouilles : l'Analyse Formelle de concepts (AFC) et l'Analyse Relationnelle de Concepts (ARC). L'AFC regroupe des objets partageant les mêmes attributs dans des concepts d'un treillis. L'ARC, une extension de l'AFC, permet de regrouper des objets partageant les mêmes attributs, mais aussi les mêmes attributs relationnels. L'apposition de contextes (une propriété de l'AFC) permet d'associer ces attributs et relations à un ensemble de classes prédéfinies et hiérarchisées par les experts du domaine. De cette façon, des définitions étendues sont proposées aux experts du domaine pour ces classes prédéfinies ainsi que de nouvelles classes inexistantes dans la hiérarchie initiale. Ces nouvelles classes peuvent être considérées pertinentes et ajoutées par les experts en tant que nouvelles « unités de connaissances ». Les treillis résultant des méthodes de fouille constituent ce que nous appelons schéma d'ontologie. Ce schéma d'ontologie est ensuite représenté par le langage $\mathcal{FLE}$ de la famille des logiques de descriptions afin d'avoir une ontologie. Cette ontologie, implémentée en OWL, a permis à notre système de répondre automatiquement à différentes questions proposées par les experts du domaine. Des expériences pratiques ont été menées dans deux domaines d'application : l'astronomie et la microbiologie.

Mots-clés: Contruction d'ontologies, TALN, AFC, ARC.

Abstract

Ontologies are used in different fields like the semantic Web or the knowledge engineering. Ontologies allow to share, to diffuse and to update knowledge domain. This thesis proposes a methodology to build ontologies using methods of Natural Language Processing (NLP) and Information Extraction (IE) for extracting prepared data from each kind of available resources in the domain (text corpora, databases, thesaurus). Then, these prepared data are mining two mining methods : Formal Concept Analysis (FCA) and Relational Concept Analysis (RCA). The FCA regroups a set of objects sharing the same set of attributes in the same concept. The RCA, an extension of the FCA regroups a set of objects sharing the same attributes and the same relations (relational attributes) in the same concept. The apposition of contexts, a property of the FCA, affects a set of attributes and relational attributes to classes pre-defined and hierarchised by the domain experts. These affectations allow us to present classes and their definitions to the experts of domain as well as new nonexistent classes in the initial hierarchy. These new classes can be considered appropriate and added by experts as new «knowledge units». The Lattices resulting from the data mining methods are considered as « ontology schema ». This ontology schema is represented in the $\mathcal{FLE}$ description logics language to obtain ontology. This ontology, is implemented in the OWL, allows us to request it. This methodology was tested in different domains : Microbiology and Astronomy.

Keywords: Building ontology, NLP, FCA, RCA.