UNIVERSITE DE SAAD DAHLAB DE BLIDA

Faculté des sciences

Département d'informatique

MEMOIRE DE MAGISTER

Spécialité : Ingénierie des systèmes et de la connaissance

APPRENTISSAGE DE CONCEPTS POUR LA CONSTRUCTION D'ONTOLOGIES BASE SUR LE CONCEPTUAL CLUSTERING

Par

FAREH Messaouda

Devant le jury composé de

A.GUESSOUM	Professeur, U de Blida	Président
H.ABED	Maître de conférence, U de Blida	Examinateur
L.ADMANE	Maître de conférence, INI Alger	Examinateur
W.HIDOUCI	Chargé de cours, INI Alger	Examinateur
S.OUKID	Maitre de conférence, U de BLIDA	Rapporteur

Blida, juin 2006

RESUME

Ces dernières années, le besoin d'ontologie apparaît pleinement, notamment en ingénierie des connaissances. Dans le cadre particulier du Web, les Ontologies vont jouer un rôle prépondérant dans le Web Sémantique, fondé sur l'annotation sémantique des ressources du Web. La vision du Web sémantique est celle d'un Web dans lequel les ressources sont accessibles non seulement aux humains, mais également aux processus automatisés comme celui d'un agent automatisé errant le Web, accomplissant des tâches utiles. Les ontologies permettent de représenter la connaissance partagée d'un domaine dans une formalisation utilisable pour différentes applications. De ce fait, les ordinateurs "comprendront" la signification de données sémantiques dans une page Web en suivant des liens vers des ontologies spécifiées.

Dans le cadre de construction d'ontologies par apprentissage, nous nous intéressons particulièrement à l'apprentissage de concepts. Nous présentons dans ce mémoire une méthode pour apprendre de concepts d'une ontologie décrite par un schéma représenté dans un langage de description appelé RDF, et un cas d'apprentissage présenté par un graphe RDF, ce graphe est définit en utilisant le schéma RDF initial. Nous construisons une nouvelle hiérarchie entre de nouveaux concepts qui sont plus spécifiques au domaine, basant sur le conceptual clustering et d'une manière incrémentale.

Mots clés **:** ontologie, apprentissage, processus de construction, RDF, RDFS, conceptual clustering**.**

ملخص

في السنوات الاخيرة احتياج الانتلوجية(l'ontologie) زاد كثيرا،خاصة في ميدان هندسة المعرفة. فى الاطار الخاص الواب الانتلوجية تلعب دور مهم في الواب المعنوي(le web sémantique) الذى يتركز على الملاحظة المعنوية لمصادر الواب. نظرة الواب المعنوي هي نظرة الواب حيث تكون المصادر سهلة المنال ليس من طرف الانسان فقط و لكن أيضا من طرف معالج اوتوماتيكي مثل عامل معلوماتي يتجول في الواب، ينجز مهمات نافعة. الانتلوجية تسمح بتمثيل المعرفة الموزعة لمجال ، في شكل مستعمل لمختلف التطبيقات. و لهذا الآلة تفهم معنى المعلومات في صفحة واب وهذا تبعا لروابط الانتلوجيةالخاصة.

في اطار بناء الانتلوجية بالتعلم، نهتم بشكل خاص بتعلم المفاهيم .نقدم في هذه المذكرة طريقة لتعلم المفاهيم لانتلوجية موصفة في مخطط ممثل بلغة وصف تسمى ب RDF وحالة تعلم ممثلة برسم بياني بRDF ،هذا الرسم البيانى يعرف باستعمال المخطط الابتدائي ل RDF. نتحصل على تسلسل جديد بين مفاهيم جديدة والتي تكون خاصة بالمجال المعين، مستعملين تجمع المفاهيم(conceptual clustering) بطريقة متزايدة.

الكلمات الرئيسية : الانتلوجية ، طريقة تعلم، مراحل بناء الانتلوجية،RDF، RDFS ، تجمع المفاهيم .

ABSTRACT

Those last years, the need of ontology appears fully, notably in Knowledge engineering. In the particular field of the Web, Ontologies go to play a part preponderant in the Semantic Web, founded on the semantic ressources annotating of the Web. The vision of the semantic Web is that of a Web in who ressources are no accessible only to human, but equally to the automatized process as that of an automatized agent rambling the Web, accomplishing useful tasks. Ontologies permit to present the divided knowledge of a domain in a usable formalization for different applications. Thereby, computers "will comprise" the meaning of semantic datas in a Web page in proportionately to ties to specifyed ontologies.

In the field of building ontologies by learning, we interest particularly to the learning concepts. We present in this memorial a method to learning concepts of an ontology described by an schema written in a language of description called RDF, and a learning example presented by a graph RDF, this graph is defines using the schema RDF initial. We build a new hierarchy enters new concepts who are more specific to the domain, basing on the conceptual clustering and of a manner incremental.

.

Keywords**:** ontology, learning technique, process of construction, RDF, RDFS, conceptual clustering**.**

REMERCIEMENTS

Je remercie avant tout le bon dieu qui m'a aidé à réaliser ce modeste travail.

Je tiens à remercier ma promotrice Mme. OUKID d'avoir aidé à réaliser ce mémoire et l'améliorer par ses remarques et conseils, pour sa patience, sa confiance, et sa compréhensibilité.

Je remercie les personnes qui m'ont fait l'honneur de participer au jury de ce mémoire : Mme ABED, Mr ADMANE et Mr HIDOUCI qui ont accepté la charge d'examinateur et dont les remarques enrichissantes ont contribué à améliorer ce mémoire, et Mr GUESSOUM qui, dans sa charge de Président de jury de ce mémoire, a su tout mettre en oeuvre pour que ma soutenance se déroule dans les meilleures conditions.

Je tiens à exprimer mes profondes gratitudes à Mr.BELKAID Ismail pour son aide et ses conseils, durant mon projet.

Je remercie particulièrement Mme.BOUMAHDI Djaouida pour son aide et ses encouragements le long du projet.

J'adresse mes sincères remerciements aux amis qui ont partagé avec moi les moments de doute et les moments d'espoir : Djaouida, Maamar, Fouaz et Hamza.

Je remercie tous les enseignants de la faculté des sciences de BLIDA et surtout mes enseignants du département informatique.

Du fond du coeur je remercie mes parents. Aucun mot n'est assez fort pour leur exprimer la reconnaissance sincère que je leur porte pour la richesse de leurs enseignements.

Je tiens à exprimer mes vifs remerciements à mon fiancé Aziz qui m'a beaucoup aidé, par ces conseils, ses encouragements et ses préoccupations, surtout dans les moments difficiles.

Je remercie, de tout coeur, tous ceux qui ont contribué de prés ou de loin à la réalisation de ce travail.

TABLE DES MATIERES

LISTE DES ILLUSTRATIONS, BRAPHIQUES ET TABLEAUX

INTRODUCTION

Web sémantique

L'avènement du World Wide Web (WWW) a mis, à la portée de chacun, une source considérable d'information. Le web contient plusieurs milliards de documents qui sont consultés par plusieurs centaines de millions d'utilisateurs à travers le monde.

Les pages Web représentant une énorme masse de connaissances difficilement exploitables. Cette masse augmente sans cesse ainsi que le nombre d'utilisateurs qui veulent pouvoir trouver facilement les informations qu'ils y recherchent. En raison de ce volume, il est difficile de trouver de l'information pertinente dans cette masse de données. La croissance du web (en volume, diversité des formats, langages) rendra de plus en plus difficile son utilisation.

Le Web sémantique est une solution proposée à ce problème. C'est un prolongement du Web courant où l'information a un sens bien défini ce qui permet de la traiter aussi bien par des ordinateurs que des humains. En donnant un sens aux informations cela permet aux agents (au sens informatique) de mieux comprendre le contenu et ainsi de mieux assister les humains.

La vision du web sémantique s'appuie principalement sur l'utilisation des ontologies représentées dans un langage de description permettant de rassembler un ensemble de définitions de concepts, et d'exprimer des annotations sur les ressources web, les ontologies permettent au agents informatique de comprendre les diverses annotations et de communiquer entre eux.

Ontologies

Les ontologies occupent une place centrale dans la réalisation du Web sémantique parce qu'elles donnent un sens à l'information. Le terme ontologie est emprunté du domaine de la philosophie où il signifie l'étude sur l'existence de l'être en tant qu'être.

La définition la plus souvent rencontrée est celle de Gruber : “une ontologie est une spécification explicite d'une conceptualisation ”. Une conceptualisation étant une vision simplifiée du monde que l’on veut représenter.

La notion d’ontologie est devenue un élément clé dans toute une gamme d’applications faisant appel à des connaissances. Une ontologie est définie comme la conceptualisation des objets reconnus comme existant dans un domaine, de leurs propriétés et des relations les reliant. Leur structure permet de représenter les connaissances d’un domaine sous un format informatique en vue de les rendre utilisables pour différentes applications

Les ontologies sont des représentations structurées de connaissances d’un domaine sous forme de réseau conceptuel. Elles sont aujourd’hui considérées comme un support indispensable à la communication entre des agents, les ontologies sont utilisées dans les domaines de l'intelligence artificielle et de la représentation des connaissances. Son champ d'application s'élargit considérablement.

Les ontologies se veulent des représentations de connaissances bâties au niveau conceptuel. Pour construire une ontologie d'un domaine donné, il faut décider des termes qui le constituent, ainsi que des relations qui les unissent. Ces données sont ensuite représentées par des langages pour permettre une compréhension commune et partageable d’un domaine qui peut être communiquée à des humains et à des ordinateurs.

Les langages des ontologies du Web sémantique

La mise en place d’ontologie sur le Web nécessite des standards qui assurent une interopérabilité sémantique.

- XML

Le XML (eXtensible Markup Langage) est un standard pour structurer des données. En 1998 l’organisme de normalisation du Web le W3C fit une recommandation sur XML 1.0. XML est un langage qui permet de présenter de l’information à l’aide de balises. XML schéma (XMLS) est un standard qui définit des types de données (entier, réel, décimal, etc..) qui peuvent être utilisés dans des documents XML. La flexibilité de XML permet d’inventer des balises pour décrire de l’information.

Une déclaration de type de document : Document Type Declaration (DTD) permet de spécifier comment utiliser les balises d’un document XML pour former un document bien formé. La DTD spécifie seulement la syntaxe du document et non sa sémantique. XML est un

outil pour permettre d'encoder n'importe quelle structure de donnée mais est incapable d'en fournir son usage et son sens. Les applications qui utilisent XML pour échanger des données doivent donc s'entendre sur le vocabulaire. D'autres langages sont nécessaires pour atteindre ce but.

- RDF(S)

Le Web sémantique est une approche prometteuse où la sémantique du contenu des documents est rendue explicite. RDF (Resource Description Framework) est à la base du Web sémantique.

Le W3C a adopté le langage RDF comme un des formalismes standard de représentation de connaissances sur le Web. Utilisant la syntaxe XML qui constitue déjà un standard, le RDF permet de décrire des ressources Web en termes de ressources, propriétés et valeurs. Une ressource peut être une page Web (identifiée par son URI, United Ressource Identifier) ou une partie de page. Les propriétés couvrent les notions d'attributs, relations ou aspects et servent à décrire une caractéristique d'une ressource en précisant sa valeur. Les valeurs peuvent être des ressources ou des littéraux. Le langage RDF dispose d'une sémantique formelle analogue à celle des Graphes Conceptuels.

Pour décrire n'importe quel type de connaissances à l'aide de ce formalisme, on doit d'abord décrire en RDF le modèle sémantique à utiliser. Par exemple, pour décrire des connaissances en terme de concepts et de relations hiérarchisés, l'introduction des types « concepts » et « relations » et des propriétés de subsomption et d'instanciation est nécessaire. Un schéma de base incluant les primitives sémantiques généralement utilisées a ainsi été ajouté au RDF et constitue ce qu'on appelle le RDF Schema (RDFS).

Les spécifications de RDFS viennent enrichir les primitives de RDF en assignant une sémantique supplémentaire aux ressources. RDFS ajoute les notions de classe pour les ressources, il permet de définir une ressource comme étant une classe. Une classe est un concept générique représentant un type ou une catégorie similaire à la notion de classe dans les langages orientés objets. Il permet aussi de définir les relations entre ces concepts ainsi que les hiérarchie des concepts et celle des relations.

Dans notre travail on s'intéresse à la construction de la hiérarchie des concepts, à partir d'une ontologie initiale présentée par un RDFS, et un graphe RDF défini en se basant sur ce RDFS, ce qui représente un cas d'apprentissage.

Apprentissage des ontologies

La définition de méthodes et d'outils d'aide à la construction d'ontologies constitue un enjeu important pour les applications du Web Sémantique.

L'utilisation d'ontologies entraîne un besoin de moyens pour pouvoir gérer ces ontologies : comment les créer (automatiquement ou manuellement), comment les présenter à l'utilisateur, comment les faire évoluer, les mettre à jours et les modifier.

Notre travail se focalise sur l'apprentissage des ontologies et plus particulièrement sur l'apprentissage des concepts pour la construction des ontologies, il permet d'enrichir des ontologies initiales pour obtenir des nouveaux concepts plus spécifiques aux domaines.

Nous nous intéressons par la construction d'ontologies par apprentissage, à partir d'une ontologie initiale décrite par un schéma présenté par le langage RDF,il s'agit du RDFS , un RDFS permet de représenter les connaissances ontologiques utilisées dans des déclarations RDF, c'est un ensemble de déclarations de classes et de propriétés.

Apprentissage de concepts

L'utilisation d'ontologies conduit nécessairement à des techniques pour permettre la construction, la gestion et l'enrichissement de ces ontologies. L'élaboration de concepts définis dans une ontologie représente une difficulté pour l'ontologiste. Donc comment identifier les concepts ?

L'objectif de l'apprentissage des concepts est d'améliorer les anciens concepts et de découvrir les nouveaux, pour obtenir des concepts efficaces, plus spécifique, et adaptés au domaine.

Notre travail a pour objectif de définir une méthode d'aide à la construction d'une hiérarchie. C'est une construction automatique de classes. Nous allons porter notre attention essentiellement sur les méthodes d'apprentissage dites de regroupement conceptuel qui permettent de construire une hiérarchie, il s'agit du conceptual clustering.

La méthode proposée est fondée sur le regroupement conceptuel des objets, la construction de la hiérarchie est incrémentale. Les méthodes incrémentales représentent une approche efficace vers le regroupement conceptuel.

Pour décrire notre travail, nous avons organisé ce mémoire comme suit :

Premièrement nous proposons un état de l'art de l'ingénierie ontologique et d'apprentissage tout en mettant en lumière certaines des principales étapes de processus de construction et les formalismes de représentation. L'état de l'art est composé de deux chapitres :

Le premier commence par présenter les ontologies, ainsi que différentes activités du processus de construction d'une ontologie. Les différents éléments dont elles sont constituées sont ensuite décrits. A la fin de ce chapitre, une typologie des ontologies structurée selon les domaines de connaissances modélisés et selon le degré d'engagement sémantique qu'elles autorisent sera exposée.

Le reste de cette partie décrit l'apprentissage structurel des concepts ainsi que les méthodes de conceptual clustering. En premier lieu, une présentation de la problématique de classification et une description des différents types de clustering seront exposées, en décrivant quelques notions fondamentales pour l'organisation d'une hiérarchie des concepts tels que : la taxonomie, la spécialisation et la subsomption.

Nous passons par la suite de ce mémoire est à la construction des ontologies du web sémantique, après avoir exposée dans la première partie les principaux langages formels de représentations des ontologies, comme les logiques de description et les graphes conceptuels, nous traitons dans cette partie les langages destinés pour la représentation des ontologies en vue de leur utilisation pour le web sémantique comme le RDF. Celui-ci est présenté dans le premier chapitre de cette partie.

Dans le deuxième chapitre nous abordons les ontologie RDFS et l'annotation des ressources web par le RDF, nous proposons premièrement un passage depuis les graphes conceptuel vers le RDF puis nous passons à la construction des ontologie RDFS et à la présentation des graphes RDF en utilisant ces schéma.

A la fin de cette partie nous décrivons dans un troisième chapitre la méthode proposée pour l'apprentissage de concepts et la conceptualisation automatique d'une ontologie. nous présentons dans un premier lieu une description détaillées des différentes étapes de construction d'une hiérarchie des concepts en se basant sur le conceptual clustering , et dans un deuxième lieu nous exposons notre outil DLOT (Definition and Learning Ontology Tool) qui permet de définir des ontologies RDFS, et d'éditer des graphes RDF en utilisant ces ontologies puis de faire l'apprentissage pour enrichir l'ontologie initiale , il s'agit d'une nouvelle hiérarchie entre de nouveaux concepts. Le dernier chapitre de cette partie consiste en

une présentation des résultats et des tests sur deux applications en utilisant notre outil DLOT et nous montrons pour chaque test les différents concepts et hiérarchies obtenus d'une manière détaillée.

Nous achevons notre mémoire par une conclusion générale, et dégageant les perspectives.

CHAPITRE 01

LES ONTOLOGIES : DEFINITIONS ET PROCESSUS DE CONSTRUCTION

1.1.Introduction

Les ontologies deviennent des ressources essentielles pour la construction de systèmes à base de connaissance. Cependant, leur construction et modélisation constituent un problème délicat du fait qu'elles manipulent la sémantique.

Afin de définir les ontologies, nous avons organisé ce chapitre comme suit:

La première section consiste en une présentation générale des ontologies comprenant des définitions sur les ontologies, leur origine, et leur utilité au sein d'un système à base de connaissance. La seconde section est consacrée à la description des différentes étapes d'un processus de construction d'une ontologie.

1.2.Origine et définitions des ontologies

L'ontologie est un terme philosophique qui signifie la *Science ou théorie de l'être*. Bien que ce soient les Grecs qui aient inventé cette science, ils ne l'avaient pas appelé ontologie, le terme étant beaucoup plus récent (XVIIe siècle:17) que la discipline qu'il désigne [15]. La discipline a évolué en se rapprochant des sciences cognitives et de l'IA, il y a seulement une vingtaine d'années.

Devant la problématique de la sémantique des informations, les systèmes à base de connaissance (SBC) qui leur ont succédé sont censés permettre le stockage et la consultation de connaissances, le raisonnement automatique sur les connaissances stockées, la modification des connaissances stockées, et, avec le développement des réseaux, le partage de connaissances entre systèmes informatiques. De manière générale, il ne s'agit plus de faire manipuler en aveugle des connaissances à la machine, mais de permettre un dialogue, une coopération entre le système et l'utilisateur humain. Le système et l'utilisateur doivent avoir accès non seulement aux termes,

mais également à la sémantique que l'être humain associe aux différents termes. Plus précisément, les représentations symboliques utilisées dans les machines doivent avoir du sens aussi bien pour la machine que pour les utilisateurs. Pour cela, la représentation des connaissances sous forme de règles logiques, utilisée dans les Systèmes Experts, ne suffit plus [1]. Pour modéliser la richesse sémantique des connaissances, de nouveaux formalismes sont introduits, qui représentent les connaissances. Les logiques de descriptions et les graphes conceptuels sont des exemples de tels formalismes. Ces langages permettent de représenter les concepts sous-jacents à un domaine de connaissance, les relations qui les lient, et la sémantique de ces relations, indépendamment de l'usage que l'on souhaite faire de ces connaissances.

Nous présentons dans cette section quelques définitions sur les ontologies :

1) La définition de GRUBER : La première définition de la notion d'ontologie fut proposée par GRUBER en 1993 [2], il considère que toute représentation de connaissances est basée sur une conceptualisation. Une conceptualisation est l'ensemble des entités existantes dans le domaine et les relations existantes entre ces entités.

GRUBER définit une ontologie comme la *« spécification d'une conceptualisation »*.

«Ontology is a specification of a conceptualization…. That is, ontology is a description of the concepts and relationships that can exist for an agent or a community of agents» [2].

2) L'ontologie, étymologiquement «ontos» (être) et «logos» (science, langage), s'intéresse à la «science de l'être en tant qu'être, indépendamment de ses manifestations particulières». Elle relève donc de la métaphysique [17]. Le terme d'«ontologie» a une signification plus large, il désigne l'ensemble des connaissances relatives à un domaine : objets, concepts, relations et propriétés.

3) La définition de J. Sowa:

Le sujet d'ontologies est l'étude des catégories des choses qui existent ou peuvent être existé dans certains domaines, le résultat de cette étude est une ontologie. Elle est représentée par des concepts et les relations entre ces concepts.

« The subject of *ontology* is the study of the *categories* of things that exist or may exist in some domain. The product of such a study, called *ontology*, the types in the ontology represent the *predicates*, *word senses*, or *concept and relation types* [7].

1.2.Composition d'une ontologie

Les connaissances traduites par une ontologie sont à véhiculer à l'aide des éléments suivants : *Concepts*, *Relations, Fonctions, Axiomes,* et *Instances*.

❖Les concepts, aussi appelés termes ou classes de l'ontologie, correspondent aux abstractions *pertinentes* d'un segment de la réalité (le domaine du problème), retenues en fonction des objectifs qu'on se donne et de l'application envisagée pour l'ontologie. Un *concept* est une abstraction réunissant un certain nombre d'entités du "monde réel" (Aristote) qui sont ses *instances* [20].

❖ Les relations traduisent les associations existant entre les concepts présents dans le segment analysé de la réalité. Ces relations incluent les associations suivantes:

- Sous-classe-de (généralisation – spécialisation) ;
- Partie-de (agrégation ou composition) ;
- Associée-à ;
- Instance de, etc.

Ces relations nous permettent d'apercevoir la structuration et l'interrelation des concepts, les uns par rapport aux autres.

❖ Les fonctions constituent des cas particuliers de relations, dans laquelle un élément de la relation, le nième (extrant) est défini en fonction des n-1 éléments précédents (intrants).

❖ Les axiomes constituent des assertions, acceptées comme vraies, à propos des abstractions du domaine traduites par l'ontologie [15].

❖ Les instances constituent la définition extensionnelle de l'ontologie, une extension est un ensemble d'objets véhiculent les connaissances du domaine.

1.3.L'utilité des ontologies

Pour bien aborder les ontologies, il est nécessaire de préciser leur utilité au sein des SBC et dans un cadre plus large.

❖ Les connaissances du domaine d'un SBC

L'ontologie devait servir de représentation des connaissances du domaine pour un SBC. Plus précisément, l'ontologie sert de squelette à la représentation des connaissances du domaine dans

la mesure où elle décrit les objets, leurs propriétés et la façon dont ils peuvent se combiner pour constituer des connaissances du domaine complètes.

❖ La communication

Les ontologies peuvent intervenir dans la communication entre humains. Dans ce cas, elles servent par exemple, à créer au sein d'un groupe ou d'une entreprise un vocabulaire standardisé. Pour de tels besoins, on est plutôt dans le cadre d'une ontologie informelle.

Dans le cas de la communication entre êtres humains et ordinateurs, l'ontologie est formelle et sert en général une tâche précise dans le SBC ou le système d'information.

❖ L'interopérabilité

L'interopérabilité est une spécialisation de la communication, dans ce cas vue entre deux ordinateurs. L'ontologie répertorie alors les concepts que des applications peuvent s'échanger même si elles sont distantes et développées sur des bases différentes. Cette interopérabilité est l'interopérabilité sémantique qui s'appuie d'abord sur une interopérabilité syntaxique.

❖ L'indexation et la recherche d'information

Plus récemment, les travaux autour du Web sémantique ont réactivé la problématique et l'utilisation des ontologies: en sus d'un rôle de médiateur, les ontologies y sont utilisées pour l'indexation, fournissant les index conceptuels décrivant les ressources sur le Web [5].

1.4.Classification des ontologies

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

1) Objet de conceptualisation ;

2) Niveau de détail;

3) Niveau de complétude;

4) Niveau de formalisme de représentation.

Ces dimensions de classification sont illustrées à la figure suivante et le détail de chacune des quatre dimensions est montré par la suite.

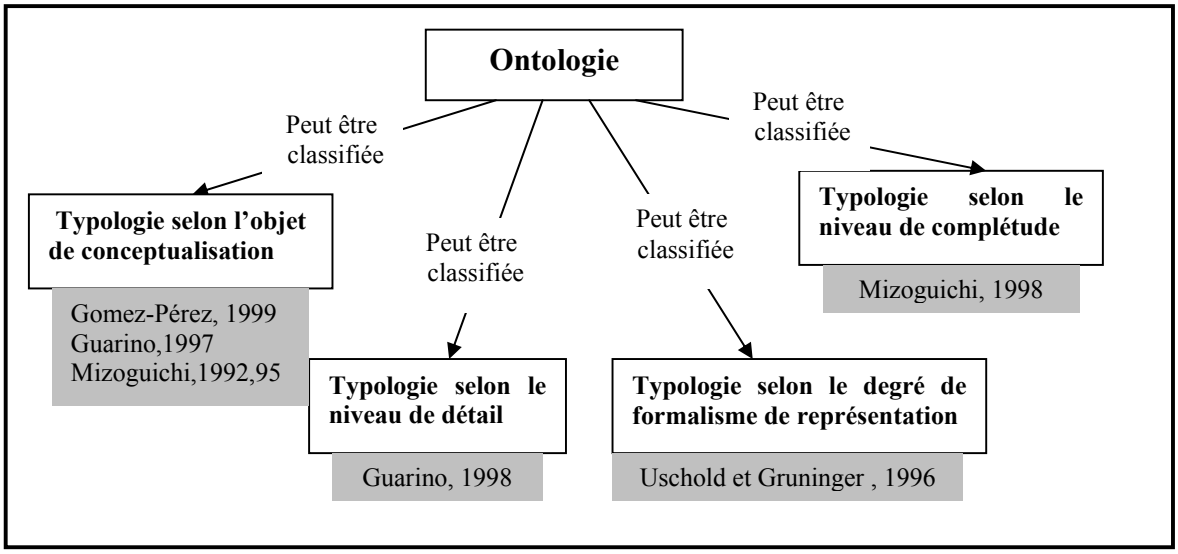


Figure 1. 1 : Typologie d'ontologies selon quatre dimensions de classification

1.5.1.Typologie selon l'objet de conceptualisation

Les ontologies classifiées selon leur objet de conceptualisation de la façon suivante :

1) Représentation des connaissances;

2) Supérieure (Haut niveau);

3) Générique ;

4) Domaine ;

5) Tâche;

6) Application.

• Ontologie de représentation des connaissances : ce type d'ontologies regroupe les concepts (primitives de représentation) impliqués dans la formalisation des connaissances. Un exemple est l'*ontologie de Frame* qui intègre les primitives de représentation des langages à base de *frames* : classes, instances, facettes, propriétés/*slots*, relations, restrictions, valeurs permises, etc.

• Ontologie supérieure ou de Haut niveau : Cette ontologie est une ontologie générale. Son sujet est l'étude des catégories des choses qui existent dans le monde, soit les concepts de haute abstraction tels que: les entités, les événements, les états, les processus, les actions, le temps, l'espace, les relations, les propriétés.

• Ontologie Générique : Cette ontologie aussi appelée, méta-ontologies ou *core ontologies*, véhicule des connaissances génériques moins abstraites que celles véhiculées par l'ontologie de haut niveau, mais assez générales néanmoins pour être réutilisées à travers différents domaines.

Elle peut adresser des connaissances d'un domaine particulier, dans ce cas elle est appelée ontologie générique du domaine, ou encore des connaissances visant à résoudre des problèmes génériques appartenant aux différents domaines dans ce cas c'est une ontologie générique de tâche.

• Ontologie du Domaine : Cette ontologie régit un ensemble de vocabulaires et de concepts qui décrit un domaine d'application ou monde cible. Elle permet de créer des modèles d'objets du monde cible. L'ontologie du domaine est une méta-description d'une représentation des connaissances, c'est-à-dire une sorte de méta-modèle de connaissance dont les concepts et propriétés sont de type déclaratif. La plupart des ontologies existantes sont des ontologies du domaine. De nombreuses ontologies de domaine existent déjà, telles que MENELAS dans le domaine médical et l'ontologie de la géométrie qui est présenté (partiellement) dans la figure suivante :

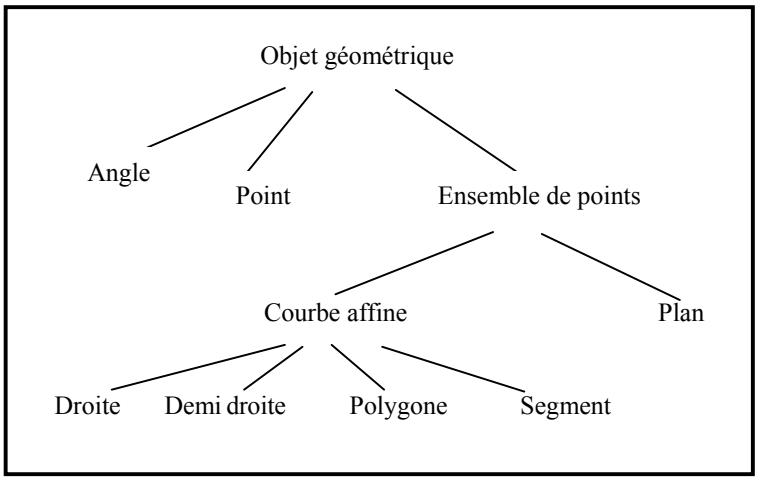


Figure 1. 2 : La hiérarchie partielle des types de concept s du domaine de la géométrie [1].

• Ontologie de Tâches : ce type d'ontologie est utilisé pour conceptualiser des tâches spécifiques dans les systèmes, telles que les tâches de diagnostic, de planification, de conception, de configuration, soit tout ce qui concerne la résolution de problèmes. Elle régit un ensemble de vocabulaires et de concepts qui décrit une structure de résolution des problèmes inhérente aux tâches et indépendante du domaine. L'ontologie de tâche caractérise l'architecture computationnelle d'un système à base de connaissances qui réalise une tâche.

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

Une "ontologie générique du domaine" est indépendante de l'application, dont les concepts et les relations soient réutilisables par différentes applications. Par exemple, de nombreux concepts et relations (Hémisphère, Lobe, Gyrus ou Sulcus, partie-de, etc.) d'une ontologie générique de l'anatomie du cerveau, construite sur le principe de l'indépendance vis a vis de l'application, ces concepts sont de façon évidente partagés par les deux applications présentées ci-après. Chacune d'elles peut, pour ses propres besoins, affiner ces concepts, ou les compléter par des concepts spécifiques à son sous-domaine d'application (e.g. tumeur, aire du langage pour l'application de planing chirurgical).

- Une application d'enseignement de neuroanatomie utilise nécessairement des concepts tels Hemisphere, Lobe, Gyrus ou Sulcus, et des relations taxonomiques, méréologiques (composition) et topologiques et Elle donne une description spécifique pour certain concepts telsque le concept du lobe frontal.
- Une application d'aide au planning chirurgical décrit le geste chirurgical à réaliser comme une succession d'étapes (inciser la peau, découper le volet osseux, etc.). Les concepts anatomiques précédents lui sont indispensables pour définir la localisation, et sont complétés par des concepts pathologiques (tumeur, foyer épileptogène) pour définir des zones cibles, ou par des concepts fonctionnels (aire du langage, aire de l'audition) pour définir des zones à éviter.

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

Cette typologie se base sur la granularité d'une ontologie c'est-à-dire le niveau de détail utilisé lors de la conceptualisation de l'ontologie, deux catégories peuvent être identifiées :

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

2) Granularité large : correspondant à un vocabulaire moins détaillé comme par exemple dans les scénarios d'utilisation spécifiques où les utilisateurs sont déjà préalablement d'accord à propos d'une conceptualisation sous-jacente .

1.5.3.Typologie selon le niveau de complétude

Cette classification est proposée sur trois niveaux suivants :

• Niveau 1 (sémantique) : Tous les concepts (caractérisés par un terme/libellé) doivent respecter les quatre principes différentiels :

1) Communauté avec l'ancêtre : un concept partage une certaine similarité avec son concept père,

2) Différence (spécification) par rapport à l'ancêtre : un concept est différent dans certain propriété à concept père, sinon il n'y aurait pas besoin de définir le concept fils,

3) Communauté avec les concepts frères (situés au même niveau), une propriété est commune aux concepts frères issus du même concept père mais s'exprime différemment pour chaque frère. exemple: les concepts « homme» et « femme » portent la propriété « sexe » héritée de leur concept père « humain », mais cette propriété vaut « masculin » chez « homme » et « féminin » chez « femme »,

4) Différence par rapport aux concepts frères (sinon il n'aurait pas lieu de le définir). Ces principes correspondent à l'engagement sémantique qui assure que chaque concept aura un sens univoque et non contextuel associé. Deux concepts sémantiques sont identiques si l'interprétation à travers les quatre principes différentiels aboutit à un sens équivalent.

• Niveau 2 (ontologique) : Outre les caractéristiques énoncées au niveau précédent, les concepts référentiels (ou formels) se caractérisent par un terme/libellé dont la sémantique est définie par une extension d'objets. L'engagement ontologique spécifie les objets du domaine qui peuvent être associés au concept, conformément à sa signification formelle. Deux concepts formels seront identiques s'ils possèdent la même extension.

• Niveau 3 (opérationnel) : Outre les caractéristiques énoncées au niveau précédent, les concepts du niveau opérationnel ou computationnel sont caractérisés par les opérations qu'il est possible de leur appliquer pour générer des inférences (engagement computationnel).

La figure suivante présente la classification des ontologie suivant le niveau de complétude :

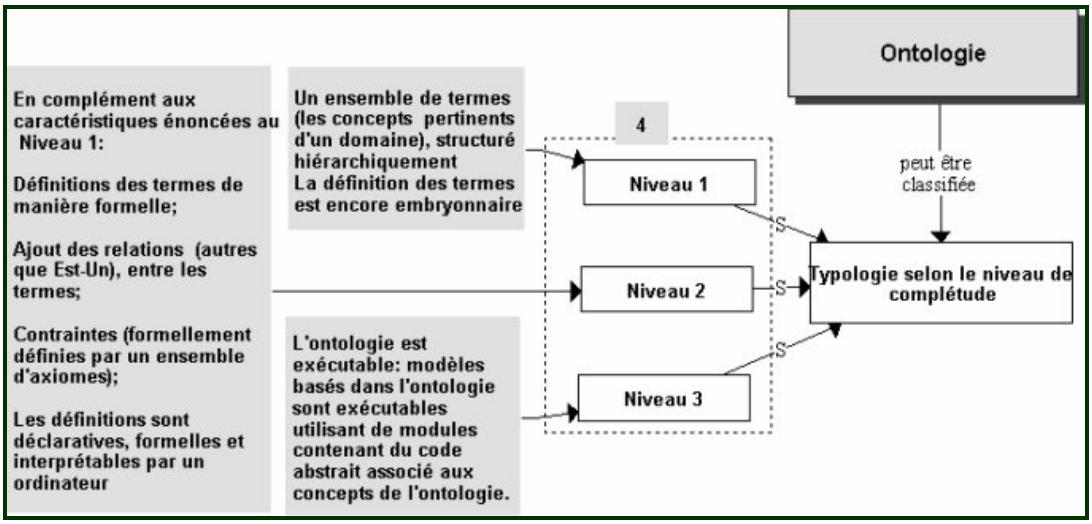


Figure 1. 3 : Classification des ontologies suivant le niveau de complétude [15].

1.5.4.Typologie selon le niveau du formalisme

C'est une classification par rapport au niveau du formalisme de représentation du langage utilisé pour rendre l'ontologie opérationnelle, elle comprend quatre catégories :

1) Informelles : ontologies opérationnelles dans un langage naturel (sémantique ouverte) ;

2) Semi-informelles : utilisation d'un langage naturel structuré et limité ;

3) Semi-formelles : langage artificiel défini formellement;

4) Formelles : utilisation d'un langage artificiel contenant une sémantique formelle, ainsi que des théorèmes et des preuves des propriétés telles la robustesse et l'exhaustivité. Parmi les langages opérationnels nous présentons deux langages : les logique de description et les graphes conceptuels :

A. Les logiques de description

Les Logiques de Description (LD) permettent de représenter les connaissances sous forme de concepts, de rôles et d'individus. Les rôles sont des relations binaires entre concepts et les individus sont les instances des concepts. Une LD est composée de deux parties, un langage terminologique (Tbox) et un langage assertionnel (Abox) :

- Tbox ou connaissances terminologiques : les concepts (les catégories) et les rôles (relation entre éléments de catégories)

- Abox ou connaissances assertionnelles : les individus (qui sont assertés comme étant instances de concepts)

➢ Terminologie de LD : Un langage terminologique permet de définir une terminologie. Cette terminologie est la représentation structurée et déclarative d'un vocabulaire d'un domaine. Une terminologie est définie comme une séquence de déclarations qui sont des introductions : de définitions, de termes et de restrictions sur des définitions déjà déclarées.

<terminologie> ::= { <TermIntroduction> | <restriction> }*

Où le symbole * signifie la répétition 0 ou plusieurs fois du terme. En effet, à tout instant, on peut introduire un nouveau terme ou bien restreindre les caractéristiques d'un terme existant pour le spécialiser et ainsi créer un nouveau terme.

Un terme est introduit une fois et une seule fois et il peut être l'introduction : soit d'un rôle ou d'un concept

<TermIntroduction> ::= <ConceptIintroduction> | <RoleIntroduction>

➢ Concepts : Un concept exprime la notion de classe ou ensemble d'individus. Il correspond à un prédicat unaire en LPO[1].

<ConceptIintroduction> ::=
<AtomicConcept> ≡ <Concept>|
<AtomicConcept > ≤ <Concept> |
<AtomicConcept> ≤ ANYTHING
. <AtomicConcept> ::= <Identifieur >

➢ Rôles de LD : Un rôle exprime une relation entre une paire d'individus : c'est donc un prédicat binaire en LPO.

[1] [1] LPO : Logique de Premier Ordre.

<RoleIintroduction> ::=
<AtomicRole> ≡ <Role> |
<AtomicRole> ≤ <Role> |
<AtomicConcept> ≤ ANYROLE
<AtomicRole> ::= <Identifieur>

- Constructeurs de LD

Pour introduire un concept complexe, c'est-à-dire une combinaison de concepts déjà introduits ou bien prédéfinis, on dispose d'opérateurs de construction ou constructeurs (AND, ALL, ATLEAST, ATMOST) et un opérateur de restriction (DISJOINT).

- AND : La construction (AND <Concept1>, <Concept2>..) exprime le fait que le concept introduit vérifie les définitions <Concept1>, <Concept2>. etc. Par exemple, « une moto est un deux roues à moteur » s'écrit : *Moto ≡ (AND Deux-roues, Engin-à-moteur)*

- ALL : La construction (ALL rôle <Concept>) exprime une restriction de valeur (type): c'est-à-dire le domaine d'un rôle ou encore le type du deuxième élément de la relation binaire qui est un rôle, le type du premier élément étant forcément celui du concept en cours de définition. Par exemple pour dire que « toutes les roues d'un 4x4 sont motrices » 4x4 ≡ (ALL roue Motrice)

- ATLEAST, ATMOST: La construction (ATLEAST n rôle) ou n est un entier strictement positif, exprime une restriction de nombre (cardinalité) : c'est-à-dire le nombre d'argument minimal du rôle (respectivement ATMOST le nombre maximal). Par exemple : pour dire « un mammifère possède quatre jambes exactement on écrira :

Mammifere ≡ (AND (ATLEAST 4 jambe)
(ATMOST 4 jambe))

- Restriction " DISJOINT " : L'introducteur de restriction :

(DISJOINT <AtomicConcept1><AtomicConcept2>) permet d'exprimer le fait que l'intersection des deux ensembles dénotés par les concepts est vide. Par exemple, pour décrire les concepts d'homme et de femme on pourra écrire :

Man ≤ Human, Woman ≤ Human, (DISJOINT Man Woman)

- La négation d'un concept C peut être exprimé par ¬C

B. Les graphes conceptuels

L'une des particularités du graphe conceptuel est de permettre de représenter des connaissances sous forme graphique. Plus précisément, un graphe conceptuel est un graphe biparti étiqueté qui se compose de nœuds « concepts » et de nœuds « relations ». Une telle représentation graphique des connaissances permet à des utilisateurs de comprendre, créer ou modifier directement des objets de ce type, de façon beaucoup plus simple qu'avec une représentation sous forme de formules logiques.

B.1) Définition du modèle des graphes conceptuels : Le modèle de graphe conceptuel se décompose en deux parties suivantes:

- Une partie terminologique dédiée au vocabulaire conceptuel des connaissances à représenter, c'est-à-dire les concepts et les relations. Cette partie correspond à la représentation du modèle conceptuel mais intégré également des connaissances sur la hiérarchisation des concepts et des relations, cette partie est définit par un support.

Un support définit le vocabulaire de base avec lequel on représente des connaissances sur un domaine, il comprend des ensembles ordonnés de types de concepts et de relations, les relations étant munies de signatures.

Définition [22] Un support est un tuplet S= (Tc, T_r, M,σ) vérifiant :

i. L'ensemble des types de concepts, noté par T_c. Un type de concepts encapsule les caractéristiques communes à plusieurs concepts. Les concepts sont des instances de leur type de concepts. On définit une relation de spécialisation sur T_c, notée par $\leq_c$. Deux types de concepts particuliers sont définis : Le type de concepts universel, noté T, qui encapsule tous les autres types, et le type de concepts absurde, noté⊥, qui représente le type de concepts vide d'instance.

ii. L'ensemble des types de relations, noté par T_r. Les types de relations représentent les relations mises en évidence entre les différents concepts. A chaque type de relations est associée une signature basée sur les types de concepts définis dans T_c. Les types de relations sont également hiérarchisés par une relation de spécialisation notée $\leq_r$.

iii. L'ensemble des marqueurs individuels, noté par M. Ces marqueurs permettent d'identifier les concepts, c'est-à-dire les instances d'un type de concepts. Un marqueur particulier, le marqueur

générique, noté *, est ajouté à M. Il sert à désigner un concept dit générique, dont on connaît le type mais pas l'identifiant.

iv. L'ensemble de signatures de types de relations, noté par σ. La signature d'un type de relation t est représentée par un graphe étoile qui fixe l'arité de la relation t.

Les figures suivantes représentent un exemple d'un support du graphe conceptuel. Les types de concepts sont définis par une hiérarchie. L'opération de spécialisation est représentée par une droite liée des types de concepts.

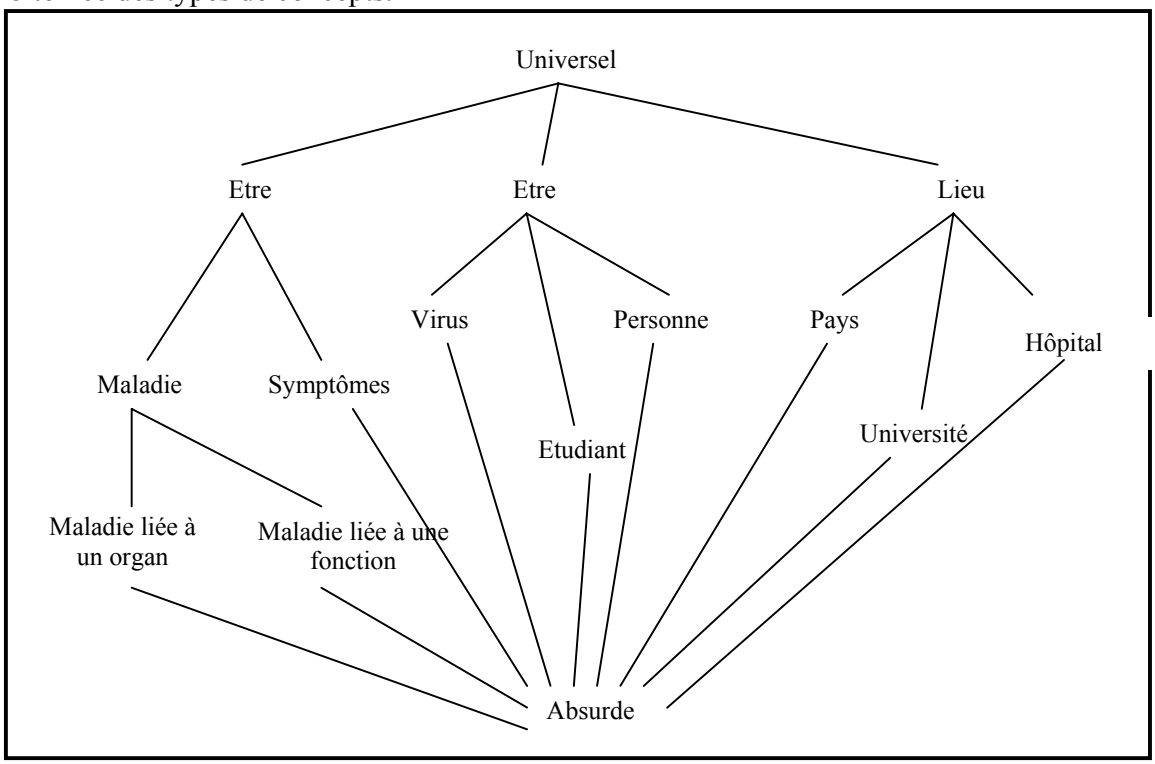


Figure 1. 4:Exemple de hiérarchie de types de concepts [1]

Après avoir construit l'ensemble de types de concepts, on définit des types de relations binaires et leur signature. La figure suivante représente les relations correspondant aux liens entre des types de concepts.

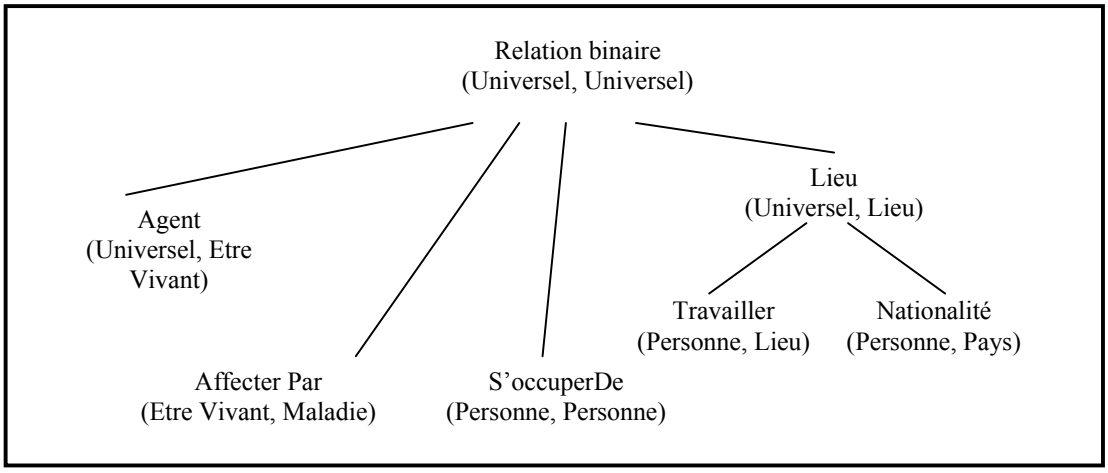


Figure 1. 5 :Exemple de hiérarchie de relations [1]

- Une partie assertionnelle dédiée à la représentation des assertions du domaine de connaissance étudié [1].

Au niveau assertionnel, les connaissances non terminologiques sont représentées en utilisant le vocabulaire décrit dans le support. Ces connaissances sont présentées sous forme de graphes d'un type particulier appelés graphes conceptuels.

Un graphe conceptuel est un multi-graphe fini, biparti, composé de sommets concepts et de sommets relations. Chaque sommet concept possède un type de concepts et un marqueur individuel. Chaque sommet relation possède un type de relations. Les arêtes du graphe ne peuvent lier qu'un sommet concept à un sommet relation. Un exemple du graphe conceptuel simple :

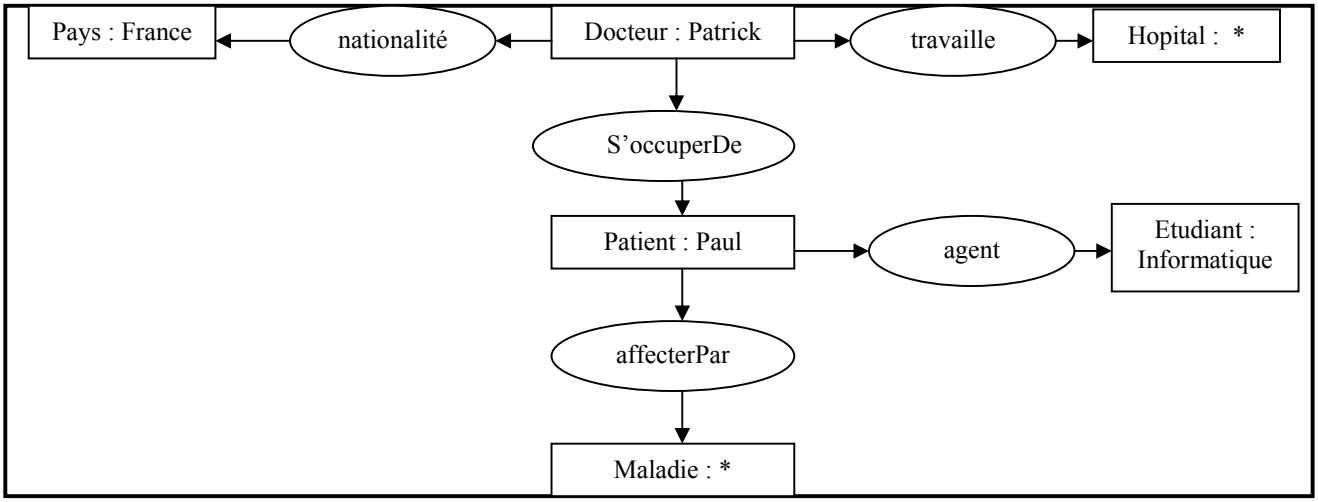


Figure 1. 6 : Un exemple d'un graphe conceptuel [22]

B.2) Les avantages des graphes conceptuels

Le modèle de graphe conceptuel est appliqué largement dans la représentation des connaissances grâce à ses avantages :

- D'abord, la facilité de représentation graphique d'un graphe est appréciée. Il est évident qu'un graphe conceptuel est "plus facile à lire" que la formule logique qui lui est associée. C'est une représentation "objet" faisant bien apparaître les relations entre les divers objets
- Un graphe conceptuel est équivalent à une petite partie de la logique du premier ordre
- D'autre part, les aspects algorithmiques sont très importants car les inférences se calculent par des algorithmes de graphes, et les algorithmes polynomiaux, ou des heuristiques.

1.6.La construction des ontologies

L'ontologie définit le vocabulaire et la sémantique associée qui permettent à deux agents de communiquer sur un domaine de connaissance donné. Une ontologie possède une syntaxe qui est celle du langage utilisé par la machine pour manipuler les connaissances, et une sémantique qui est celle du domaine de connaissance. Étant utilisée par une machine, une ontologie se doit donc être formelle (voir figure 1.7).

Une ontologie sera constituée d'un ensemble de définitions de termes désignant les concepts et les relations de la conceptualisation et d'un ensemble d'axiomes imposant une sémantique dans l'utilisation des termes [1].

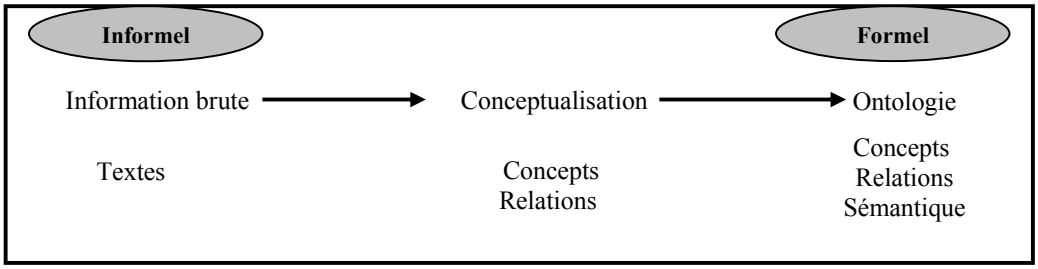


Figure 1. 7 :L'ontologie comme spécification d'une conceptualisation [1]

1.6.1.L'évaluation des besoins

Le but visé par la construction d'une ontologie se décline en :

- L'objectif opérationnel : il est indispensable de bien préciser l'objectif opérationnel de l'ontologie à créer pour le domaine considéré [16].

- Le domaine de connaissance : il doit être délimité aussi précisément que possible, et découpé si besoin est, en termes de connaissances du domaine, connaissances de raisonnement, connaissances de haut niveau (communes à plusieurs domaines) ;
- Les utilisateurs : identifier les futurs utilisateurs de l'ontologie à créer.
- Les sources d'information: déterminer les experts du domaine, les documents technique.
- La portée d'ontologie : déterminer à priori la listes des termes du domaine à représenter.

1.6.2.Le processus de construction

L'analyse des travaux actuels sur les ontologies permet de dégager un certain consensus sur le processus de construction d'une ontologie exploitable dans un système à base de connaissances. La figure 1.7 illustre ce consensus qui repose sur un enchaînement de trois étapes permettant de passer des données brutes à l'ontologie opérationnelle.

Bien que ces étapes soient effectuées en séquence, il est évident que tous les retours arrières sont possibles, en d'autres termes, la réalisation d'une étape peut provoquer la remise en cause des choix adoptés aux étapes précédentes et, par voie de conséquence, conduire à compléter ou modifier les ontologies obtenues en amont du processus [3].

On peut alors décrire le processus représenté dans la figure 1.7 qui est découpé en trois étapes:

1. La conceptualisation consiste, à partir des données brutes, à dégager les concepts et les relations entre ces concepts permettant de décrire de manière informelle les entités cognitives du domaine. Elle est réalisée par un expert du domaine assisté de l'ingénieur de la connaissance et aboutit à un modèle conceptuel [1].

La conceptualisation, permet d'aboutir à un modèle informel, donc sémantiquement ambiguë. Ce modèle, vise à sélectionner les termes qui vont être retenus comme primitives de description du domaine et à associer, à chacun de ces termes, des informations permettant de limiter les ambiguïtés sémantiques dues à leur utilisation [3].

Le travail présenté dans ce mémoire est une conceptualisation automatique d'une ontologie c'est-à-dire, il s'agit de trouver une hiérarchie entre les concepts d'une ontologie. Le fait d'automatiser cette conceptualisation, nous a amené à travailler avec un langage formel permettant le traitement automatique de l'information.

2. L'ontologisation est la formalisation partielle, indépendante des contraintes d'opérationnalisation, du modèle conceptuel dont on précise la sémantique à l'aide d'axiomes et de contraintes sur les concepts et les relations. Elle est réalisée par un ingénieur de la connaissance aidé par l'expert du domaine et aboutit à une ontologie conceptuelle. Celle-ci est semi-formelle, certaines connaissances ne pouvant pas forcément être formalisées à ce niveau [1].

L'ontologisation, consiste à modéliser, les propriétés formelles du domaine considéré. L'objectif est d'obtenir un modèle dans lequel la quasi-totalité des ambiguïtés inhérentes au langage naturel ont été levées. Cependant, dans la majorité des travaux, le modèle obtenu est souvent qualifié de semi-formel pour justifier le fait que certaines connaissances ne peuvent pas être totalement formalisées.

En fait, cette étape produit un résultat en deux parties :

- l'une *formelle* qui dispose d'une sémantique claire,
- l'autre *informelle* qui ne dispose pas d'une sémantique claire, ou tout au moins d'une sémantique fixée *a* priori [3].

3. L'opérationnalisation est une formalisation de l'ontologie conceptuelle à l'aide d'un langage de représentation de connaissances opérationnel, c'est-à-dire qui permet la manipulation automatique des connaissances, par exemple, le modèle des Graphes Conceptuels ou la logique de descriptions. Elle est donc menée par un spécialiste du langage de représentation et par l'ingénieur de la connaissance. On obtient alors une représentation formelle des connaissances du domaine, basée sur l'ontologie [1]. La figure suivante présente les différentes étapes décrites ci-dessus:

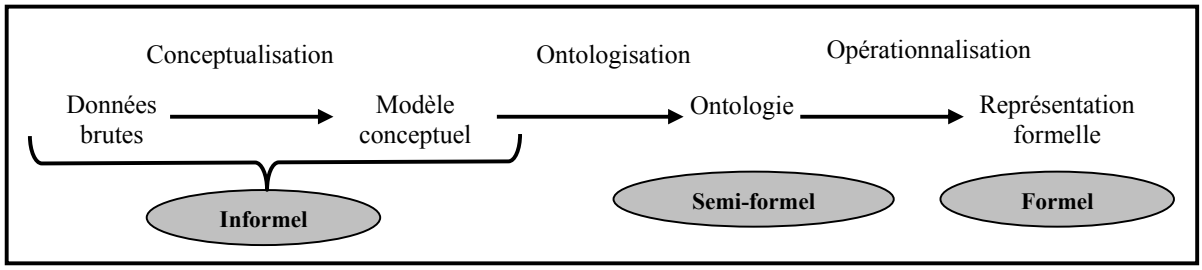


Figure 1. 8 :Processus de représentation de connaissances[1].

1.7.Principes de construction des ontologies

Il existe un ensemble de critères et de principes pour le développement des ontologies. Certain critères sont proposé par Gruber, ils peuvent être résumés comme suit :

- la clarté et l'objectivité des définitions des concepts (Gruber), qui doivent être indépendantes de tout choix d'implémentation [1]. Les ambiguïtés doivent être réduites, des définitions en langage naturel doivent être fournies [5]. L'ontologie doit fournir la signification des termes définis en fournissant des définitions [15].

- Complétude (Gruber) : Une définition exprimée par des conditions nécessaires et suffisantes est préférée à une définition partielle [15].

- L'extensibilité (Gruber) : une ontologie doit pouvoir être étendue sans modification [1]. L'ontologie doit être construite de telle manière que l'on puisse l'étendre facilement, sans remettre en cause ce qui a déjà été fait [5].

- Diversification des hiérarchies (Arpirez) : Ce principe est adopté pour augmenter la puissance fournie par les mécanismes d'héritage multiple. Si suffisamment de connaissances sont représentées dans l'ontologie et que suffisamment de différentes classifications de critères sont utilisées, il est plus facile d'ajouter de nouveaux concepts et de les faire hériter de propriétés de différents points de vue [15].

- Distance sémantique minimale (Arpirez) : Il s'agit de la distance minimale entre les concepts enfants de mêmes parents. Les concepts similaires sont groupés et représentés comme des sous-classes d'une classe, et devraient être définis en utilisant les mêmes primitives, considérant que les concepts qui sont moins similaires sont représentés plus loin dans la hiérarchie [15].

1.8.Conclusion

Nous avons vu le long de ce chapitre qu'une ontologie est une conceptualisation d'un domaine à laquelle sont associés un ou plusieurs vocabulaires de termes. Les concepts se structurent en un système et participent à la signification des termes.

L'ontologie est définie pour un objectif donné et exprime un point de vue partagé par une communauté. Elle est vue comme un mécanisme d'interopérabilité du fait qu'elle permet la

communication entre systèmes informatiques, indépendamment de l'architecture de chaque système, et de leurs domaines d'applications.

Une ontologie s'exprime dans un langage et repose sur une théorie (sémantique) garante des propriétés de l'ontologie en termes de consensus, cohérence, réutilisation et partage.

Dans ce travail, nous nous intéressons par l'étape de conceptualisation d'une ontologie, ceci est fait d'une manière automatique, ce qui justifie le choix d'un langage formel, et tant que notre contexte est le web, notre choix est porté sur un langage qui permet la description des ressources du web, c'est le modèle RDF (ressource description framework). Et comme le Web contient de vastes bases de données avec plusieurs buts et de multiples sources non homogènes, l'être humain ne peut pas agir à cause de la grande masse d'information, nous devons dans ce cas utiliser des techniques d'apprentissage pour la construction des ontologies pour permettre leur traitement automatique, en construisant les hiérarchies des concepts et des relations. L'apprentissage des ontologies sera présenté dans le chapitre suivant.

CHAPITRE 02

APPRENTISSAGE STRUCTUREL DES CONCEPTS

2.1.Introduction

En ingénierie des connaissances, le besoin des ontologies apparaît pleinement, ainsi que la construction des ontologies en utilisant des techniques d'apprentissages, ce travail aborde précisément le problème de l'apprentissage de concepts pour la construction d'ontologies.

Dans ce présant chapitre, nous commençons par définir les concepts structurés qui reflètent la structure naturelle, notamment dans les domaines complexes, la définition d'un concept sera présentée par extension et intension, passer ensuite aux notions fondamentales pour l'organisation d'une hiérarchie des concepts y compris : la taxonomie, la spécialisation et la subsomption. La deuxième section de ce chapitre, s'intéresse à l'apprentissage des concepts, de telles méthodes ont été développées au sein de l'apprentissage automatique en Intelligence Artificielle. Nous allons porter notre attention essentiellement sur les méthodes d'apprentissage dites de regroupement conceptuel permettant de construire une hiérarchie entre un ensemble de concepts.

2.2.Concepts structurées

Dans l'apprentissage des concepts non supervisé, quelques récents travaux sont limités par des domaines non structurés, dans leurs instances sont décrites par fixer des ensembles des pairs attribut valeur et par conséquent, plusieurs domaines peuvent être décrit par des simples langages [59]. Mais les instances dans les domaines complexes ont des structures naturelles, les objets ont des composants et des relations entre ces composants, donc pour certains domaines la représentation attribut valeur est inadéquate.

Un objet structuré compris des parties et des relations entre ces parties, un concept structuré est définit par des objets structurés dans leurs extensions [60].

2.2.1.Définition d'un concept : extension et intension

Les connaissances portent sur des objets auxquels il est fait référence à travers des concepts. Un concept peut représenter un objet matériel, une notion, une idée [63]. Un concept peut être divisé en trois parties : un terme (ou plusieurs), une notion et un ensemble d'objets. La notion également appelée intension du concept, contient la sémantique du concept, exprimée en termes de propriétés et d'attributs.

L'ensemble d'objets, également appelé extension du concept, regroupe les objets manipulés à travers le concept, ces objets sont appelés instances du concept. Par exemple, le terme « table » renvoie à la fois à la notion de table comme objet de type « meuble » possédant un plateau et des pieds, et à l'ensemble des objets correspondant à cette description. Un concept est ainsi doté d'une sémantique référentielle (celle imposée par son extension) et d'une sémantique différentielle (celle imposée par son intension).

Il est à noter qu'un concept peut très bien avoir une extension vide. Il s'agit alors d'un concept générique, correspondant généralement à une notion abstraite [63].

Deux concepts peuvent partager la même extension sans pour autant avoir la même intension. Ceci correspond à des points de vue différents sur un même objet. Par exemple, l'ordinateur peut être considéré comme outil de travail, ou comme moyen de divertissement.

Bien que le langage naturel contient de nombreux termes désignant plusieurs concepts sémantiques différents (par exemple « table » pour un meuble et « table » pour un tableau de valeurs numériques), de telles ambiguïtés ne sont pas gérables en machine, où un concept est généralement identifié à l'aide de ses termes. Mais la restriction à un domaine de connaissance permet généralement d'éviter les homonymies de concepts. Il apparaît par contre souhaitable de gérer les synonymies et de permettre la désignation d'un concept par plusieurs termes, pour assurer une plus grande souplesse d'utilisation de l'ontologie. De manière générale, l'articulation entre les aspects référentiels et différentiels des concepts est délicate. Certains auteurs estiment que l'intension d'un concept permet à elle seule d'en préciser le sens. D'autres jugent indispensable de spécifier un concept à travers les deux paradigmes sémantiques [63].

2.2.2.La taxonomie

Organiser automatiquement des connaissances est devenue une préoccupation centrale dans les domaines, comme la recherche d'information. En Intelligence Artificielle, ce problème fait l'objet de recherches dans le domaine du regroupement conceptuel. La problématique du regroupement conceptuel pour l'organisation de connaissances est définit principalement par le fait que l'on ne cherche pas à construire un sous ensemble de regroupements particuliers mais tous les regroupements d'objets ayant des similitudes. De fait, la construction d'une classification n'est pas guidée par la capacité prédictive de celle ci mais par le type de langage de généralisation utilisée pour décrire les classes.

Intuitivement, deux objets se ressemblent si leurs descriptions se ressemblent. Plus formellement la proximité entre objets n'est pas évaluée par une distance numérique mais représentée par la description maximalement spécifique qui généralisent les descriptions de ces objets [58].

Le choix du langage de représentation des objets est un compromis difficile. En effet, plus le langage est expressif et permet de représenter des types d'attributs variés des connaissances sur le domaine ou encore la structure des objets, mais aussi la complexité de la construction de la classification augmente.

2.2.2.1.Définition d'une taxonomie

Une taxonomie est définie par un ensemble de classes C, muni d'une relation de généralité $\leq$ qui est un ordre partiel et d'une fonction I d'interprétation des classes vers un domaine X quelconque [58].

2.2.2.2.Objectifs d'une taxonomie

La construction de taxonomie est une opération courante en sciences : en médecine pour découvrir des classes de malades ayant les mêmes combinaisons de symptômes, en botanique pour mettre en évidence des sous espaces d'une même variété etc.

Une taxonomie scientifique peut être motivée par deux objectifs principaux .D'une part, le but originel de tout processus de regroupement est la structuration d'un ensemble de données, par là, une amélioration de la visibilité (possibilité de visualisation) est réalisée. En outre, une taxonomie offre une description synthétique des données et donc facilite l'accès vers celles-ci.

La connaissance structurée peut appuyer la découverte de concepts sur un domaine : les classes présentes dans la collection d'individus peuvent traduire un phénomène ou une loi jusqu'aux lors inconnus. D'autre part, la taxonomie peut servir à des fins de prédiction : prédire une caractéristique particulière d'un nouvel individu selon sa classe dans la taxonomie.

L'objectif vis à par une taxonomie influe sur le processus de sa construction. Celui ci consiste à identifier les classes et à les caractériser. Intuitivement, on regroupe les individus qui exhibent des similitudes selon un critère. C'est ce critère qui reflète le but de la construction, il a donc souvent un caractère subjectif. La difficulté essentielle d'une automatisation du processus de construction est l'expression du critère de regroupement approprié.

L'utilité d'une taxonomie peut se manifester sur plusieurs plans. Par l'ensemble des classes, elle peut contribuer à confirmer ou infirmer des hypothèses sur le domaine considéré, ainsi que suggérer de nouvelles hypothèses. Dans un plan pratique, la taxonomie offre une représentation synthétique et structurée de l'ensemble des données.

2.2.3.La subsomption

La subsomption : un concept C1 subsume un concept C2 si toute propriété sémantique de C1 est aussi une propriété sémantique de C2. Si C1 est plus spécifique que C2. L'extension d'un concept subsumé est forcément plus réduite que celle du concept qui le subsume. Son intension est par contre plus riche. La subsomption sert à la hiérarchisation de l'ensemble des concepts de l'ontologie. Par exemple, *Humain* subsume *Homme* [63].

2.2.4.La spécialisation

Les classes d'une base de connaissances sont organisées hiérarchiquement selon une relation de généralité, la spécialisation.

La spécialisation est une relation d'ordre partiel. Elle a une sémantique qui repose sur l'inclusion ensembliste : l'extension d'une classe est incluse dans celle de chacune de ses super classes.

La spécialisation structure l'ensemble des classes en une hiérarchie. Lorsque la hiérarchie résultante est un arbre (au plus une super classe) on parle de mono_spécialisation, dans le cas contraire on est en présence d'une multi_spécialisation [58].

L’héritage est un mécanisme permettant de profiter de l’organisation hiérarchique des classes pour optimiser le stockage et les modifications des connaissances exprimées. Ainsi, une classe hérite de ses super classes l’ensemble des attributs que celles ci définissent avec la totalité des facettes. La classe peut reprendre tel quel un attribut, le masquer ou le raffiner (redéfinir). Du coup, les descriptions des attributs hérités d’une super classe n’ont pas le même statut. En cas de mono spécialisation, l’ordre entre les super classes d’une classe est total et permet une combinaison entre descriptions héritées sans conflits. En revanche, si l’on autorise une classe à avoir plusieurs super classes incomparables, on parle d’héritage multiple, la situation est beaucoup plus complexe car des conflits peuvent apparaître entre définitions héritées de super classes indépendantes [58].

2.3.Description et représentation des concepts

L’utilisation d’ontologies rend nécessaire des techniques pour gérer ces ontologies, pour les créer et pour les mettre à jour. Dans le cas de l’utilisation de formalismes permettant de définir des concepts, l’élaboration de concepts définis est une difficulté pour l’ontologiste.

La question suivante est donc importante : Comment identifier les concepts définis à intégrer dans l’ontologie ? À partir d’ontologie contenant des concepts primitifs, nous exploitons les annotations utilisant ces ontologies pour repérer de nouveaux concepts définis. Il s’agit de former à partir de la base d’objets, des concepts, et d’organiser ces concepts dans une structure hiérarchique suivant la relation de généralisation définie sur les intensions et les extensions.

2.3.1.Apprentissage des concepts

L'apprentissage est une composante majeure de tout système d'Intelligence Artificielle, lui permettant d'accomplir des tâches incomplètement spécifiées et/ou d'améliorer ses performances. L'apprentissage automatique, permet d'acquérir de telles connaissances à partir d'exemples. Les systèmes d'apprentissage ont montré leur capacité à acquérir des connaissances structurées tout en prenant en compte les connaissances du domaine et des besoins quand elles sont disponibles.

Parmi les techniques d’apprentissage des concepts, nous nous intéressons par celles du regroupement conceptuel, ces techniques essayent de former des classes rassemblant les objets similaires. Afin de créer une hiérarchie contenant toutes les classes possibles, où les classes sont un couple intension et extension. Mais comme notre représentation des objets est par les graphes, la notion de similarité n’a pas la même définition que dans les objets classiques où la similarité

est mesurée par une distance, est par conséquence la complexité des opérations de manipulation de graphes.

2.3.1.1.Découverte des concepts par apprentissage

L'objectif de l'apprentissage des concepts est de découvrir de nouveaux concepts et d'améliorer les anciens pour obtenir des concepts efficaces et adaptés au domaine.

Les actions réalisées sont principalement :

- La création d'un concept, par exemple, pour créer des concepts correspondant au domaine, avec des règles.

- La définition des propriétés, une fois les concepts créés, il faut remplir les différentes propriétés qu'ils peuvent contenir,

- L'ajout, la vérification ou la suppression d'un lien entre les concepts (relation). Du fait des erreurs possibles dues aux règles appliquées qui peuvent être erronées ou à l'environnement incertain …etc., donc il peut être utile de vérifier des liens.

2.3.1.2. Classification automatique de classes et regroupement conceptuel

La classification est un problème de regroupement conceptuel qui cherche à former des concepts à partie de l'observation d'un domaine [64]. Le regroupement conceptuel a pour objectif de construire un ensemble de concepts classifiés hiérarchiquement regroupant des objets selon leurs similarités.

La classification constitue le coeur des systèmes à objets, dont elle structure la connaissance et supporte une part de raisonnement et du calcul. Elle reflète l'organisation des concepts du domaine et elle prend des formes diverses telles que les arborescences, des graphes sans circuits ou des treilles.

La classification est principalement le résultat :

- De la découverte de classes, par deux procédés
    - Le premier est la division en classes d'un ensemble d'entités, qui produira des « classes décrites en extension »,

- Le deuxième est la description d'un ensemble d'entités par des propriétés caractéristiques, menant à des « classes décrites en intension »,

➢ De l'organisation des classes les unes par rapport aux autres, ce que l'on peut appeler « classification des classes »,

Du placements d'une entité relativement aux classes existantes, ce que l'on peut appeler « classification d'instances » [64].

2.3.1.2.1.La classification non supervisé :Clustering

Le problème de la classification en général est de construire une procédure permettant d'associer une classe à un objet. En classification non supervisé (ou Clustering), les classes possibles et leur nombre ne sont pas connues à l'avance, et les exemples disponibles sont non étiquetés. Le but est donc de découvrir des relations intéressantes qui peuvent exister implicitement entre les données , et qui permettront de regrouper dans un même groupe (ou cluster) les objets considérés comme similaires , pour constituer les classes [55].

Le Clustering peut être défini comme un processus d'organiser des objets en groupes dont les membres sont semblables d'une certaine manière. Les algorithmes de clustering groupe les instances pour les rassembler, en basant sur la similarité ou la mesure de distance entre les pairs des instances pour les mettre dans une hiérarchie "à partir de zéro" [24] [25] [23].

1) Les différentes techniques de Clustering

En général il y a deux modèles principaux de clustering : [23] [56]

a) Clustering Partitionel

Dans la phase d'initialisation, l'ensemble de tous les termes est un cluster. Dans la phase d'amélioration des petits clusters sont itérativement générées en dédoublant le plus grand clusters ou la séparation du moins homogène cluster dans plusieurs subclusters.

L'algorithme de partitionnement est basé sur le principe "generate and test". On crée plusieurs partitions, puis on évalue la qualité du partitionnement à partir de certains critères. On utilise donc beaucoup le calcul de distance entre deux éléments avec cette méthode [50].

b) Clustering hiérarchique

Les algorithmes de clustering hiérarchique créent une décomposition hiérarchique des objets. Ils existent deux types: agglomerative (bottom-up) ou divisive (top-down):

- Méthodes hiérarchiques ascendantes (bottom-up) ou agglomératives : Les méthodes hiérarchiques ascendantes sont les plus connues des méthodes de classification automatique. Elles aboutissent à un empilement de partitions dont le sommet réuni une partition et la base autant de partition que d'objets à classer. Cet empilement peut se représenter par un arbre composé de branches ayant une bifurcation binaire du sommet vers la base, la raison pour laquelle on les nomme hiérarchique [48]. L'obtention d'une partition se fait par coupure de l'arbre à un seuil donné.

Dans le clustering agglomératif, initialement chaque individu est placé dans une classe propre, On calcule la matrice de ressemblance M entre chaque couple de classes. Tant qu'il reste au moins deux classes dans M on sélectionne dans M les deux classes les plus similaires I et J On remplace I et J par une classe G plus générale [52].On met à jour M en calculant la ressemblance entre G et les classes restantes à l'aide de la mesure de distance entre groupes.

Donc, dans la phase d'initialisation, chaque terme est définie pour constituer un cluster, par la suite les clusters sont itérativement produits en fusionnant les plus semblables / moins différent jusqu'à un certain critère d'arrêt est atteint [23].

- Méthodes hiérarchiques descendantes (top-down) ou divisives : Une classification hiérarchique descendante réalise un dendrogramme non pas par agrégation, mais par division depuis tous les éléments formant une classe jusqu'à la partition formée de tous les éléments simples [48].

Dans la phase d'initialisation, l'ensemble de tous les termes est un classifieur. Dans la phase d'amélioration des petits classifieurs sont itérativement générées en dédoublant le plus grand classifieur ou la séparation du moins homogène classifieur dans plusieurs subcassifieurs.

Les techniques de Clustering: agglomératives et devisives, sont utilisées pour produire des descriptions hiérarchiques des termes.

Les clusters hiérarchiques essaient de segmenter l'espace des exemples successivement. On prend tout l'espace comme un seul ensemble, puis on le découpe en deux, ainsi de suite. On obtient une structure hiérarchique comparable a celle d'un arbre n-aire [50].

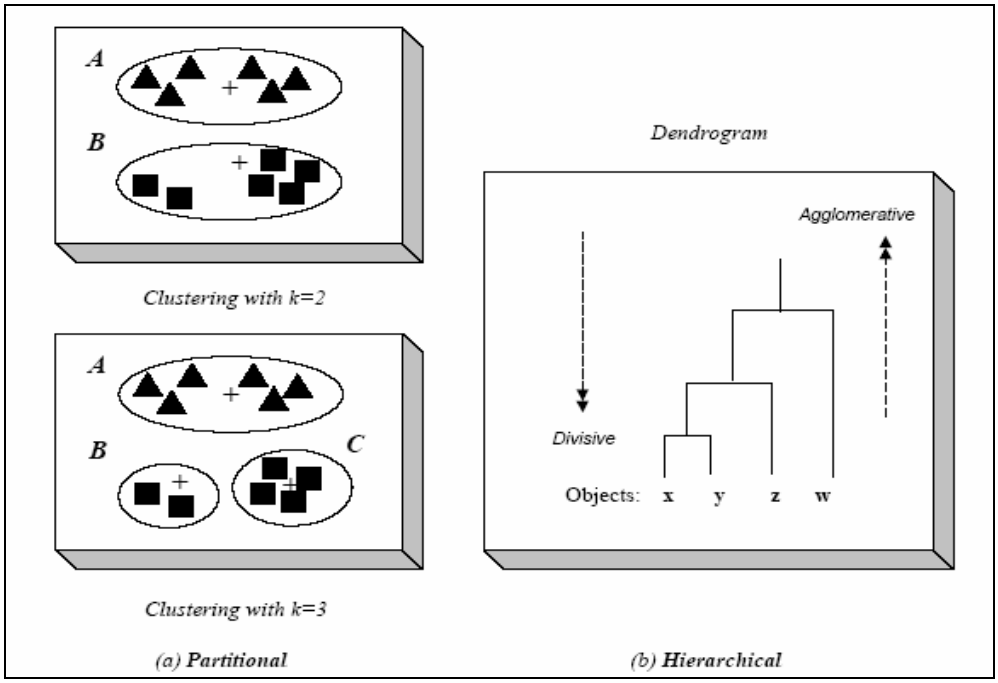


Figure 2. 1:(a) Partitional clustering for k = 2; 3, (b) Hierarchical clustering dendrogram

2) Apprentissage Incrémental et non incrémental

L'apprentissage inductive peut être performé en deux alternatives: incrémental et non incrémental :

Le système d'apprentissage incrémental traite chaque instance une par une et vise ou améliore ses modèles internes avec chaque nouvelle instance à chaque étape. L'apprentissage incrémental est la manière que les humains apprennent [57].

L'insuffisance inévitable dans cette approche, elle est sensible pour la présentation par ordre des exemples d'apprentissage [57], d'autre part les systèmes d'apprentissage non incrémental construit les descriptions des concepts après voir toute les instances d'apprentissage pour maximiser les performances.

Une méthode incrémentale va être exécuter de façon continus, et va intégrer les données au fur et a mesures de leurs arrivée dans l'algorithme. A l'inverse, une méthode non incrémentale va

considérer un ensemble de données fournies en entrée, et sera exécutée sur cet ensemble de données. Si, par la suite, une nouvelle donnée devait être fournie en entrée de l'algorithmes, celui- ci devrait être relancé à nouveau [55].

- Apprentissage incrémental : pour utiliser l'apprentissage incrémental, le concepteur doit décomposer le problème en tâches de plus en plus complexes. Bien entendu, les situations procurant une récompense positive constituent les points de départ d'une telle décomposition.

Toutefois, le processus de décomposition pourrait être automatisé et exécuté par l'agent apprenant. Celui-ci pourrait explorer aléatoirement son environnement et lorsqu'il trouverait des situations intéressantes, il pourrait travailler dessus et apprendre à les résoudre efficacement [53]. Les connaissances apprises pourraient ensuite être transmises aux autres agents.

Les méthodes incrémentales représentent une approche efficace vers le regroupement conceptuel. Du coup elles sont bien adaptées au traitement de grands ensembles de données. Leur fonctionnement autonome fait d'elles un bon outil d'extraction de connaissances dans les bases de données [58].

2.3.1.3.Algorithmes basé sur le Conceptuel Clustering et les distances globales

Parmi les algorithmes basé sur le conceptual clustering, nous présentons l'algorithme CobWeb, cet algorithme est utilisée pour la construction de la hiérarchie entre les concepts, en utilisant une mesure d'utilité qui est représenté par une distance.

2.3.1.3.1.Présentation générale du système CobWeb

CobWeb construit des hiérarchies de classes en utilisant une "mesure d'utilité". Étant donné la classification courante et un nouvel exemple [52], CobWeb mesure quelle classe devrait accueillir le nouvel exemple, ou si une nouvelle classe devrait être créée, en calculant l'utilité de chacune des actions. La mesure de distance est fournie avec l'algorithme (c'est la mesure d'utilité), et il n'y a pas besoin de spécifier combien de classes on désire obtenir. CobWeb ne travaille que sur des valeurs discrètes et des ensembles complets de données (pas de valeurs manquantes).

Afin d'obtenir une classification correcte, il faut respecter un certain compromis entre deux critères : la similarité intra-classe, qui atteste du fait que les objets d'une même classe sont

proches, et la dissimilarité inter-classe, qui assure que deux concepts distincts représentent bien deux choses différentes.

2.3.1.3.2.Critiques

- Avec la représentation attribut /valeur on touche pas la totalité de la réalité, donc cette représentation ne suffit pas pour présenter des données réels et complexe, ce qui induit la nécessité d'une autre représentation qui doit être plus expressive, il s'agit de la représentation graphique, comme le modèle RDF, ce dernier se base sur la création des graphe RDF ainsi leur schémas (RDFS) où les déclarations de RDF sont extraites.

2.4.Conclusion

Dans les domaines complexes les données ont une nature structurelle, un concept structuré peut être représenté par un graphe, où il est composé d'un ensemble des objets et des relations entre ces objets. Un concept structuré est définit par extension et intension. Pour organiser ces concepts en une taxonomie, il faut avoir des relations de subsumption ou de spécialisation entre les concepts. Le clustering est une technique pour construire une classification d'une manière automatique. Nous avons présenté trois types de clustering que nous avons distingué, nous nous intéressons par le conceptuel Clustering incrémental qui sert à construire une hiérarchie des concepts à partir des exemples d'une manière incrémentale.

La nécessité de représenter les concepts d'une ontologie représentant les ressources du web, nous a amené à choisir le modèle de description des ressource du web RDF : c'est un moyen pour exprimer des affirmations sur des ressources de façon que des machines peuvent les interpréter, ces affirmations décrivent la sémantique de cette ressource. Notre travail est de développer une méthode de classification de concepts pour permettre une conceptualisation automatique d'une ontologie dans un environnement du web sémantique. Ce que nous présentons dans la partie suivante.

CHAPITRE 03

LE LANGAGE DU WEB SEMANTIQUE : RDF

3.1.Introduction

Exprimer de la connaissance sur le web est l'ambition du web sémantique, c'est grâce à l'utilisation des ontologies. Mais si les ontologies doivent s'échanger librement sur le web, il est nécessaire de les représenter par un langage formel. Pour cela on dispose de langages comme RDF. C'est un vrai langage d'ontologie qui est plus expressif, c'est l'objet de ce chapitre.

La vision du Web sémantique est celle d'un Web dans lequel les ressources sont accessibles non seulement aux humains, mais également aux processus automatisés. L'automatisation des tâches nécessite d'élever le statut du Web de lisible par machine à quelque chose que nous pouvons appeler « compréhensible » par machine.

Dans cette vision du Web sémantique, les ontologies sont de plus en plus devenus une matière de recherches importante. Elles sont utilisées dans de nombreux domaines et elles jouent un rôle crucial dans le développement du Web sémantique.

Pour réaliser cette vision, il sera nécessaire d'annoter des ressources Web avec les méta données (c.-à-d., les données décrivant leur contenu/fonctionnalité). En particulier, nous pouvons souhaiter annoter des ressources Web avec les méta-données sémantiques qui fournissent une certaine indication de la teneur d'une ressource. Pour que l'information soit accessible non seulement aux humains mais également aux agents logiciels.

Afin de faire ceci, nous avons besoin de langages qui soutiennent la représentation des méta-données sémantiques. Les propositions de langages de méta-données sont issues de recommandations du World Wide Web Consortium (W3C)[2], c'est le cas particulier de Resource Description Framework (RDF), de RDF Schéma (RDF(S)) et de XML/XML Schéma [31].

[2] Le W3C (World Wide Web Consortium) a été créé en 1994 par Tim Berners-Lee et le MIT (Massachussets Institute of Technology), pour gérer les technologies et les évolutions du web. Ce consortium international, dirigé par

Ce chapitre a pour objectif d'introduire les concepts du langage RDF. Ce langage peut être utilisé pour la définition sémantique du Web ou de toutes autres informations électroniques. Les présentations ci-dessous n'ont pas la prétention d'être suffisante pour pouvoir créer des documents RDF, mais doivent permettre de comprendre les concepts de base. Nous montrerons aussi comment RDF Schema permet de définir de nouveaux concepts et de les utiliser dans le cadre de la définition de modèles RDF.

3.2.Vers un Web sémantique

Le Web actuel est un immense espace d'informations, dans lequel l'individu aura certainement besoin des machines pour l'aider à retrouver les informations qu'il recherche. Or la structure actuelle du Web ne facilite pas cette tache, étant donné qu'elle (la structure) avantage plus la compréhension humaine au détriment des machines [42].

3.2.1.Les lacunes du web actuel

Le Web comporte de vastes bases de données avec plusieurs buts et de multiples sources non homogènes. Ceci prouve qu'il y a nécessité d'améliorer l'accessibilité à cette importante masse d'informations et disposer d'outils plus sophistiqués pour une meilleure recherche et organisation au sein du Web [42].

Les évolutions du Web sont induites par l'augmentation de la couverture géographique de l'Internet. Des populations de plus en plus variées accèdent au réseau. La multiplicité des langues devient primordiale. Tous les systèmes d'écriture doivent être utilisables, séparément ou ensemble, dans de véritables documents multilingues. En plus de l'enrichissement sémantique, il devient nécessaire d'adopter des représentations universelles des contenus textuels [41].

Il est clair que le Web actuel doit changer d'orientation d'un Web qui présente ou affiche les informations disponibles, à un Web "intelligent" ou compréhensible. Il s'agit du Web sémantique [43].

T. Berners-Lee, repose sur trois grands organismes de recherche, qui sont les hôtes du W3C : le MIT à Cambridge, la Keio University au Japon, l'INRIA en France. Outre ces trois organismes, le W3C est composé de nombreux membres : 1)- grandes entreprises d'informatique : Adobe Systems, Apple, Bull, IBM, Cisco Systems…etc , 2)- opérateurs de télécommunications : AT&T, France Télécom…etc 3) - autres laboratoires et organismes de recherche : CNRS, le CERN 4) - institutions militaires : l'OTAN est membre du W3C.

En effet, le World Wide Web a été conçu à l'origine pour la compréhension humaine, et bien que tout ce qui y réside soit *lisible par une machine*, ces données ne sont pas *compréhensibles par une machine*. D'ou, dans le Web des débuts, seul un humain peut réellement utiliser l'information disponible sur le Web. Les machines ne sont pas capables de l'interpréter et ne fournissent donc que des outils de localisation, de transfert, de mise en forme et de présentation. C'est le genre d'inconvénients que le Web sémantique se propose de corriger.

Dans le Web sémantique, l'information a une signification explicite, ce qui permet aux machines de la traiter réellement. Cette nouvelle vision du Web s'appuie sur deux concepts :

- une représentation plus riche, plus structurée, plus rigoureuse de l'information elle-même, condition première pour que des programmes puissent agir sur le contenu,
- des métadonnées, c'est-à-dire une description externe rigoureusement formalisée de l'information principale. Des calculs sur les métadonnées permettent d'inférer des propriétés sur les données qu'elles décrivent [41]. C'est à cause du volume d'information que le Web contient, il est impossible de gérer cela manuellement [36]. La solution est d'utiliser les métadonnées pour décrire les données contenues sur le Web. Les métadonnées sont des "données à propos des données" (par exemple, un catalogue de bibliothèque est une compilation de métadonnées, puisqu'il décrit les publications) ou spécifiquement dans le contexte de cette spécification "des données décrivant les ressources Webs".

3.2.2.Notions du Web sémantique

L'objectif du web sémantique est de tendre vers un web dont la sémantique des données serait à la fois compréhensible par des utilisateurs humains et appréhendables par des entités informatiques (agents, moteurs de recherche, serveurs d'informations).

3.2.2.1.Définition du Web sémantique

Le Web Sémantique *"SW"* est une extension ou amélioration du Web actuel, dans lequel la signification des données est prise en compte, afin de permettre aux utilisateurs et aux machines de travailler en collaboration en utilisant des règles d'inférence[43].

Tout cela aura pour conséquence de réduire les données non pertinentes et accroître l'utilité du Web.

Le Web sémantique est une prolongation du Web courant dont laquelle l'information possède une signification bien définie qui permet aux ordinateurs et aux personnes de mieux travailler en coopération [28]. En donnant un sens aux informations cela permet aux agents de mieux comprendre le contenu et ainsi de mieux assister les humains.

Le web sémantique est un projet lancé par le W3C pour pallier les insuffisances actuelles du web, entre autre :

- imprécision du web et de la recherche d'information
- hétérogénéité des formats, des informations
- absence de description et d'indexation des ressources [29]

L'objectif du Web Sémantique est de rendre le contenu du Web compréhensible à des machines. Le procédé consiste à exploiter :

- des ontologies. Une ontologie est un vocabulaire constitué de concepts, de relations, voire d'axiomes, liés à un certain domaine.
- un langage commun pour exprimer les ontologies et décrire des annotations utilisant les termes de ces ontologies,
- des moteurs de raisonnement permettant d'inférer sur les annotations d'après les axiomes déclarés dans les ontologies [19],

3.2.2.2.Métadonnées et annotation sur le web

Un des grands principes du Web sémantique est qu'il est nécessaire d'associer aux ressources du Web des informations exploitables par des agents logiciels afin de favoriser l'exploitation de ces ressources.

Associer une information exploitable à une ressource signifie deux choses essentielles.

La première est que cette information doit d'une manière ou d'une autre être structurée – utilisable – et descriptive – de la ressource, de son utilisation – afin de faciliter et d'en améliorer l'accès dans le cas d'une ressource directement visualisée par un utilisateur (par exemple en permettant une recherche d'information plus efficace et plus ciblée), mais aussi l'exploitation dans le cas d'une ressource exploitée dans le cadre d'un service à l'utilisateur (l'utilisateur n'est alors pas forcément conscient de l'utilisation de la ressource).

La seconde est que la ressource en question doit exister et pouvoir être exploitée sur le Web indépendamment des informations qui lui sont associées dans le cadre du Web sémantique : celles-ci sont utiles, mais non nécessaires pour accéder et utiliser la ressource, la page Web ou le service.

Sur le Web, les informations sont fortement distribuées, volumineuses, hétérogènes et souvent peu structurées. Il est donc nécessaire d'avoir des méthodes et outils pour comprendre, manipuler et partager les documents. L'annotation sémantique à partir d'ontologies semble être la meilleure approche pour partager et exploiter l'information sur le Web. Les outils d'annotation visent à améliorer communication et interopérabilité sur le Web. Ces annotations permettent d'associer des notes de lecture aux documents, de partager l'information, d'effectuer des tâches rédactionnelles en groupe, etc.

1) Annotation

➢ L'annotation est une des formes les plus communes de méta-données dans le contexte du Web. L'annotation sémantique peut être faite sous la forme de commentaires, de notes, d'explications, de questions, de références, d'exemples, de conseil, de correction ou de n'importe quel autre type de remarque externe qui peut être attaché à un document Web ou à une partie choisie de ce document. Puisque ces éléments sont externes, il est possible d'annoter n'importe quel document Web indépendamment, sans devoir éditer ce document.

Du point de vue technique, des annotations sont habituellement vues comme des métadonnées, car elles fournissent des informations additionnelles sur un morceau existant de données [28].

2) Caractérisation des annotations

Une annotation est une information graphique ou textuelle attachée à un document et le plus souvent placée dans le document.

Les annotations peuvent prendre plusieurs formes comme des icônes, des symboles de liens, des mises en forme (souligné, italique), des images, etc.

Les annotations comportent plusieurs dimensions qui peuvent être liées à la structuration, concernant les fonctions ou le rôle des annotations.

3.2.3.Le langage du Web XML

XML reste un langage très important pour structurer le contenu de Web. Après une étude de Gartner, "l'XML dépassera en 2003 tous les autres formats de publication de données, et cela pour tous les contenus multi-supports" [44].

XML permet de décrire la structure de tout type de document. C'est est un langage de description et d'échange de documents structurés. Comme SGML[3] dont il est issu. La principale caractéristique du XML étant la possibilité d'utiliser des balises "personnalisées" [46], il décrit la structure d'un document à l'aide d'un système de balises marquant le début et la fin des éléments qui le compose. Un document est bien formé s'il obéit aux règles syntaxiques du langage XML, et il sera correctement traité par un programme adéquat, comme un parseur XML. Un document valide est forcément un document bien formé mais il obéit en plus à une structure type définie dans une DTD (*Document Type Definition*) [34].

- Les avantages de XML

  - XML apparaît comme un format d'échange de données universel.
  - XML garantit à ses utilisateurs l'indépendance de leurs documents de toute technologie propriétaire.
  - XML unifie le monde du traitement de document et celui du Web.
  - Souplesse et puissance.

Malgré tout ce que XML peut apporter, il fournit une surface syntaxique pour les documents structurés mais ne fournit aucune contrainte sémantique sur le sens de ces documents, c'est à dire même si XML a un grand pouvoir de description des documents, il n'en décrit que la structure, pas le contenu ou le sens. XML apporte beaucoup pour réaliser la vision du Web sémantique, mais il doit être complété par un autre moyen, qui s'intéresse au sens des données disponibles. On entre là dans le domaine des méta-données et de RDF, un langage d'usage général pour la description des informations du Web [41].

[3]**SGML**: **S**tandard **G**eneralized **M**arkup **L**anguage

3.2.4.Le langage du Web Sémantique RDF

Le Web sémantique est la prochaine évolution importante du Web. De par la taille que le World Wide Web a atteinte, l'accessibilité aux informations disponibles sur celui-ci exige des moyens de raisonnement. Les ontologies jouent un rôle clé dans la livraison du Web Sémantique en facilitant le partage d'information entre les communautés de humains et des agents logiciels. Pour cela, différents langages de représentation des connaissances adaptés au Web ont été créés. Les premiers, comme SHOE[4] [38] et Ontobroker [39] proposaient une extension de la syntaxe HTML pour annoter les pages Web avec des "métadonnées" sémantiques permettant une recherche d'information guidée par le contenu sémantique des pages Web. Cependant, leur manque de sémantique ne permet pas aux personnes de créer de nouveaux vocabulaires par exemple en XML, les DTDs et les schémas XML peuvent être utilisés pour échanger les données entre les parties qui ont convenu les définitions avant l'échange mais le même terme peut être utilisé avec différents sens dans différents contextes et des termes différents peuvent être utilisés pour des articles ayant le même sens. RDF et RDF Schema ont essayé de résoudre ce problème en permettant d'associer des sémantiques simples aux identificateurs. Avec RDF Schema, une personne peut définir des classes qui ayant plusieurs sous-classes et super classes, et peut aussi définir des propriétés pouvant avoir des sous propriétés, de domaines ou d'intervalle. Dans ce sens RDF Schema est un langage d'ontologie pour le Web primitif.

Le langage Resource Description Framework (RDF) est le standard émergeant proposé par le W3C pour la représentation et l'échange de métadonnées sur le Web, il est basé sur un modèle de triplets (ressource, propriété, valeur) et possède une syntaxe XML. Le langage RDF Schema (RDFS) est le standard accompagnant RDF qui permet de représenter les connaissances ontologiques relatives aux annotations [31].

RDF est muni d'une syntaxe, et d'une sémantique. Aucun mécanisme d'inférence n'est cependant proposé dans la recommandation [35].

3.2.4.1. Contexte et Web Sémantique

Le RDF part du constat qu'actuellement, toutes les informations disponibles sur le Web sont lisibles par des machines, mais ne sont pas pour autant interprétables par celles-ci. L'exemple le

[4] **SHOE** : **S**imple **H**TML **O**ntology **E**xtensions

plus marquant est le moteur de recherche que l'on utilise quotidiennement : à partir d'un mot-clé, il récupère un ensemble considérable d'informations qui n'ont en commun que le ou les mots-clés soumis au moteur de recherche, indépendamment de tout contexte d'utilisation, et donc indépendamment de toute interprétation.

La vocation initiale de RDF est l'utilisation de métadonnées pour décrire les ressources du Web, le terme "ressource" représentant toute information pouvant être référencée par un URI (Uniform Resource Identifier), comme par exemple un site Web complet, une page Web ou même un élément particulier de cette page Web. L'idée est donc de rajouter une dimension descriptive aux ressources véhiculées sur le Web, cela en leur adjoignant des descriptifs explicites qui les rendront interprétables et utilisables par les outils du Web [40].

La spécification des schémas RDF est construite au moyen de RDF lui même.

Pour cette raison, nous allons tout d'abord présenter RDF, et nous montrerons ensuite comment RDF Schema permet de définir de nouveaux concepts et de les utiliser dans le cadre de la définition de modèles RDF [40] .

La définition du RDF, telle que publiée par le W3C, précise que RDF est un langage pour représenter l'information sur les ressources du Web, particulièrement adapté pour représenter des méta données attachées à des documents disponibles sur le Web comme le titre, l'auteur, la date de modification, le copyright, etc. Par extension, ce même langage permet de représenter de l'information sur des objets identifiés sur le Web, même s'ils n'y sont pas directement retrouvables [21].

3.2.4.2. Motivation et choix du RDF

En résumé, le choix de RDF a été motivé par les raisons suivantes :

- RDF est un modèle permettant de décrire les ressources « **toute** chose » à l'aide d'un mécanisme très basique, très simple et très proche du langage naturel : la phrase , dans sa forme la plus élémentaire dans le langage naturel est simple, elle correspond à une **proposition** (ou «phrase simple») constituée par l'association d'un **sujet** (ce dont on parle) et d'un **prédicat** (ce qu'on dit sur le sujet) qui inclut généralement un verbe et ses compléments c-a-d elle est composé d'un (sujet, verbe, complément) qui correspond en RDF aux différentes déclarations des triplets (subject, predicat , literal) qui est <*sujet, prédicat, objet*>.donc le langage RDF a la vertu

d'exprimer ou de décrire des applications exprimées dans le langage naturel , en effet les domaines complexes nécessitent plus qu'un « attributs –valeurs » pour les représenter ce qui est permis par le langage RDF, et plus particulièrement par le RDFS, il s'agit du langage RDFS,quand à lui ,est dédié à la spécification de schémas associé à RDF, permettant de mieux représenter et comprendre le domaine.

- Pour réaliser un web sémantique, on doit travailler par un langage qui supporte la représentation de la sémantique de l'information, On peut se demander pourquoi ne pas se contenter d'utiliser XML pour construire ce web sémantique, ou plutôt XML à la place de RDF, Il y a plusieurs raisons à cela. La première est que : ce langage ne possède pas de sémantique ce qui rend difficile la réalisation du web sémantique. ce qui est permis par le langage RDF. Ce dernier est doté d'une sémantique et permet le traitement automatique d'information. Le modèle RDF est dédié à la spécification des sémantiques des données basées sur XML d'une manière standardisée et interopérable [36]. Ainsi que le langage XML est certes un élément de réponse à la standardisation des données sur le Web, mais ce n'est pas la réponse complète, car si XML structure les données, il ne définit pas de standard pour exprimer les métadonnées. C'est là le rôle de RDF. Alors que XML peut être utilisé comme voie générale de transport des données sur le Web, à condition de connaître au préalable la forme spécifique des données transportées, RDF pose sur XML un cadre de définition des métadonnées pour une large catégorie des données. Quand les données de XML sont déclarées de format RDF, les applications pourront comprendre une grande partie de la traduction des données sans agencement antérieur. Ceci donne à des utilisateurs du Web l'avantage des outils génériques de traitement des métadonnées qui seront réutilisables à travers une variété de domaines d'application de métadonnées. C'est pour cela que RDF est qualifié de métalangage de métadonnées.

3.3.RDF et RDFS

3.3.1.Modèle RDF élémentaire

Le modèle de données RDF et la spécification de sa syntaxe est une recommandation du W3C [31]. Il est dédié à la description des ressources du Web. Tout objet du Web identifié par une URI (uniform ressource identifier) est une ressource et peut être décrite en RDF, ce peut aussi bien être un document textuel qu'une image ou encore un document sonore [19].

La fondation de RDF est un modèle pour représenter des propriétés et des valeurs de propriétés données. Le modèle RDF s'appuie sur des principes bien établis provenant des différentes communautés de représentation des données. Vous pouvez imaginer les propriétés RDF comme les attributs de ressources et en ce sens elles correspondent aux paires traditionnel attribut-valeur. Les propriétés RDF représentent également les relations entre les ressources.

Le modèle de donnée élémentaire consiste en trois types d'objets :

- Ressources :

  ➢ Toutes choses décrites par des expressions RDF sont appelées des *ressources*. Une ressource peut être une page Web entière : comme le document HTML "http://www.w3.org/Overview.html" par exemple. Une ressource peut être une partie d'une page Web : ex. un élément HTML ou XML spécifique à l'intérieur du document source. Une ressource peut être également une collection complète de pages : ex., un site web entier. Une ressource peut être également un objet qui n'est pas directement accessible par le Web : ex., un livre imprimé. Les ressources sont toujours nommées par des URIs avec des ancres ids optionnelles. Toute chose peut avoir une URI : L'extensibilité des URIs permet l'introduction d'identificateurs pour toute entité imaginable [32].

  ➢ Une ressource est une entité manipulée dans une annotation RDF et accessible par une URI. Il existe un moyen pour spécialiser et classifer en différentes catégories ces ressources [34].

- Propriétés :

  ➢ Une *propriété* est un aspect, une caractéristique, un attribut, ou une relation spécifique utilisée pour décrire une ressource. Chaque propriété possède une signification spécifique, définit ses valeurs permises, les types de ressources qu'elle peut décrire, et les relations qu'elle entretient avec les autres propriétés [32].

  ➢ Une propriété comme son nom l'indique, permet d'attacher des informations à une ressource. C'est donc une relation binaire entre des ressources et/ou des valeurs atomiques. Tout comme pour les ressources il est possible de spécialiser les propriétés [34].

- Déclarations :

  ➢ Une ressource spécifique associée à une propriété définie ainsi que la valeur de cette propriété pour cette ressource est une *déclaration* RDF. Ces trois parties individuelles d'une déclaration sont appelées, respectivement, le *sujet*, le *prédicat*, et l'*objet*. L'objet d'une déclaration (c.-à-d., la valeur de la propriété) peut être une autre ressource ou il peut être littéral ; c.-à-d., une ressource (spécifiée par une URI) ou une simple chaîne ou autre type de données primitif défini par XML. En termes RDF, un *littéral* pourrait avoir un contenu qui est un balisage XML mais qui n'est pas davantage évalué par le processeur RDF. Il y a des restrictions syntaxiques sur la façon dont le balisage dans les littéraux peut être exprimé [32]. Une valeur peut être simplement une chaîne de caractères, ou alors une ressource [34].

Les déclarations RDF peuvent être représentées sous la forme d'un graphe orienté. Une déclaration RDF est représentée dans la figure suivante:

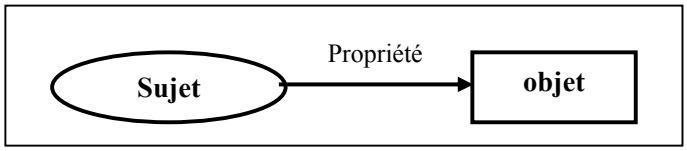


Figure 3. 1 : Représentation graphique d'une déclaration RDF

L'interprétation associée est : "*sujet* possède une propriété *propriété* qui a pour valeur *objet*". Un exemple présenté est la description d'une page Web (la ressource) comme ayant pour auteur (la propriété) une personne se nommant "NomAuteur" (l'objet) (Figure 3.2) :

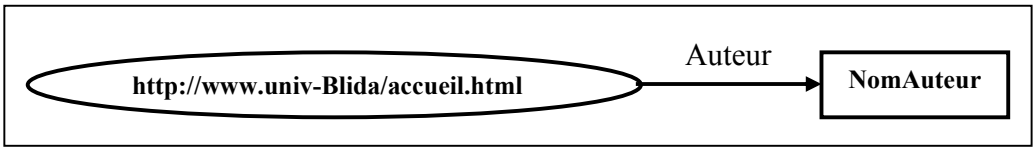


Figure 3. 2: Un exemple de déclaration RDF

3.3.1.1. Syntaxe du RDF

La syntaxe d'échange employée par RDF est celle de XML. Ainsi, les règles formulées par XML sont implicitement intégrées à la syntaxe RDF (documents bien formés, mécanisme de visibilité. La figure suivante présente l'exemple précédent sous sa forme textuelle :

```
<rdf:RDF>
     <rdf:Description about=" http://www.univ-Blida/accueil.html ">
          <schema:Auteur>NomAuteur</schema:Auteur>
     </rdf:Description>
</rdf:RDF>
```

Figure 3. 3: Une déclaration RDF/XML

3.3.1.2.Objectifs de RDF

Le développement de RDF par le W3C a été entre autres motivé par la perspective des applications suivantes :

• Manipulation et classification des métadonnées Web, afin de fournir des informations sur les ressources Web et les systèmes qui les utilisent.

• Développement de modèles d'information ouverts plutôt que fixés pour certaines applications (par exemple les activités de planification, de description de processus organisationnels, d'annotation de ressources web, etc) [27].

• Faire avec l'information traitable par machine, en permettant aux informations d'être manipulées en dehors de l'environnement particulier dans lequel elles ont été créées, éventuellement à l'échelle d'Internet. C'est le concept d'interopérabilité des savoirs.

• Optimisation de la coopération entre applications, en permettant de combiner les données de plusieurs applications, pour générer de nouvelles informations.

• Faciliter le traitement automatique de l'information du Web par des agents logiciels, transformant ainsi le web d'un regroupement d'informations uniquement destinées aux humains, en un état de réseau de processus en coopération.

3.3.1.3.Primitives additionnelles

Dans la spécification de RDF, d'autres primitives sont définies, qui s'ajoutent au modèle noyau que nous venons de décrire, pour faciliter la description des ressources du Web.

1) Les Containers : collection d'objets

Les *Containers* permettent de représenter des collections de ressources. Trois types de *Containers* sont définis :

- les *Bags* sont des multi-ensembles de ressources, qui peuvent contenir plusieurs fois la même ressource [19]. C'est une liste non ordonnée de ressources ou de littéraux. Bag est utilisé pour déclarer qu'une propriété possède plusieurs valeurs et qu'il n'y a pas de sens pour l'ordre dans lequel elles sont données. Les valeurs identiques sont permises [37].

- les *Sequences* sont des listes ordonnées de ressources ou de littéraux. Sequence est utilisé pour déclarer qu'une propriété a plusieurs valeurs et que l'ordre de ces valeurs a un sens. Les valeurs identiques sont permises [37].

- les *Alternatives* sont des listes de ressources représentant une disjonction ; l'application d'une propriété à une ressource de type *Alternative* signifie que cette propriété est appliquée à l'une des ressources de ce Container [19]. Alternative : Une liste de ressources ou de littéraux qui représentent des alternatives pour la valeur (unique) d'une propriété. Alternative pourra être utilisé pour fournir des traductions dans une autre langue pour le titre d'un travail, ou pour fournir une liste de sites Internet miroirs dans lesquels une ressource pourra être trouvée. Une application utilisant une propriété dont la valeur est une collection d'Alternative sait qu'elle peut choisir l'un des éléments de la liste lorsque c'est approprié [37].

Une utilisation du type collection est présentée (Figure 3.4) dans le cadre de la description d'une ressource de type "page HTML". La représentation RDF est la suivante :

```
<rdf:RDF>
  <rdf:Description about="http://www.univ-Blida.LABO/accueil.html">
     <schema:Createur>
         <rdf:Seq ID="NomMembresDelEquipe">
               <rdf:li>NomResponsable</rdf:li>
               <rdf:li>NomMembre</rdf:li>
         </rdf:Seq>
    </schema:Createur>
</rdf:Description>
</rdf:RDF>
```

Figure 3. 4: Les collections en RDF/XML

2) Les ressources anonymes

Le modèle RDF est dédié à la description des ressources du Web ; il est de ce fait centré sur la description de ressources *identifiées*. Il permet cependant une forme limitée de quantification existentielle grâce à la notion de ressource anonyme. Considérons par exemple le graphe RDF de la figure suivante et sa sérialisation XML.

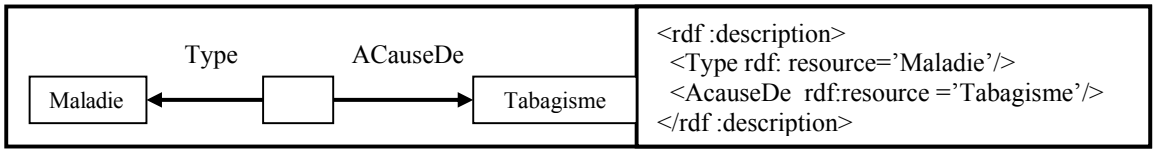


Figure 3. 5: Un graphe RDF contenant une ressource anonyme et sa sérialisation en XML

3) La réification

Le modèle RDF est pourvu d'un mécanisme de réification qui permet de considérer les déclarations RDF comme des ressources. Une déclaration (**r**, **p**, **v**) est réifiée en une ressource **s** décrite par les quatre propriétés suivantes : la propriété *subject* identifie la ressource **r**, la propriété *predicate* identifie la propriété originale **p**, la propriété *object* identifie la valeur **v** de **p** et la propriété *type* décrit le type de **s** - toutes les déclarations réifiées sont instances de la classe *Statement*. La figure suivante décrit la représentation en RDF de la déclaration suivante : *'Observer 3002 says that the rating of Newsletter 425 is seminal '*. Cette déclaration contient une déclaration réifiée : *'the rating of Newsletter 425 is seminal'*.

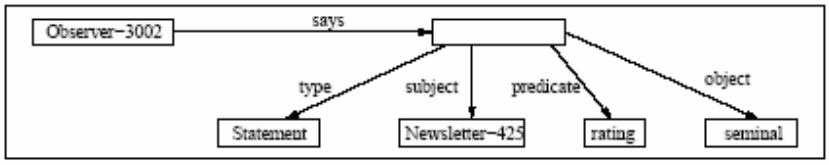


Figure 3. 6: Un exemple de graphe RDF comportant un triplet réifié

3.3.2.RDFS

RDF Schéma (RDFS) est une recommandation du W3C, il s'agit du langage de spécification de schémas associé à RDF. Il est dédié à la représentation des connaissances ontologiques utilisées dans des déclarations RDF. Un schéma RDF est un ensemble de déclarations de classes et de propriétés. Il s'apparente en cela à la partie terminologique d'une Logique de Description (LDs) ou sont déclarés les rôles et les concepts d'une base de connaissances. Il permet de définir les domaines et les codomaines (domaines de valeurs) des propriétés RDF ainsi que des hiérarchies de classes. Il permet également de représenter des hiérarchies de propriétés, celles-ci étant considérées sur le même plan que les classes.

Pour représenter les connaissances spécifiques à un domaine, un schéma RDF est défini par raffinement du schéma noyau RDFS. Les classes spécifiques au domaine sont déclarées comme

des instances de la classe *Class* et les propriétés spécifiques au domaine comme des instances de la classe *Property*. Les propriétés *subClassOf* et *subPropertyOf* permettent de définir des hiérarchies de classes et des hiérarchies de propriétés. Les hiérarchies de classes et de propriétés autorisent toutes deux l'héritage multiple. Les propriétés *range* et *domain* permettent de restreindre les domaines et codomaines des ressources de type *Property*. Dans la version actuelle du langage [31], la signature d'une propriété autorise un domaine à valeur unique : une propriété peut être appliquée à différentes classes mais sa valeur appartient à une classe unique.

Le métamodèle de RDFS est présenté sur la figure présentée par la suite. Il est défini comme un ensemble de déclarations RDF utilisant les trois propriétés RDFS fondamentales *subClassOf*, *subPropertyOf* et *type*. La propriété *type* permet de représenter la relation d'instanciation entre ressources et classes : une ressource **r** de *type* **C** appartient à l'ensemble des individus représentés par **C**.

Par exemple, sur la figure 3.7, dans l'annotation RDF, la ressource *Lirmm* est une instance de la classe *Institute* ; dans la définition du schéma RDFS, la propriété *activity* est une instance de la classe *Property*. Les propriétés *subClassOf* et *subPropertyOf* représentent les relations de subsomption qui permettent d'organiser les classes et les propriétés RDF en hiérarchies : une classe C subsume une classe D si C est plus générale que D, i.e. l'ensemble des ressources de *type* C contient l'ensemble des ressources de *type* D. Par exemple, sur la figure 3.7, la classe *InanimateEntity* subsume la classe *Institute*.

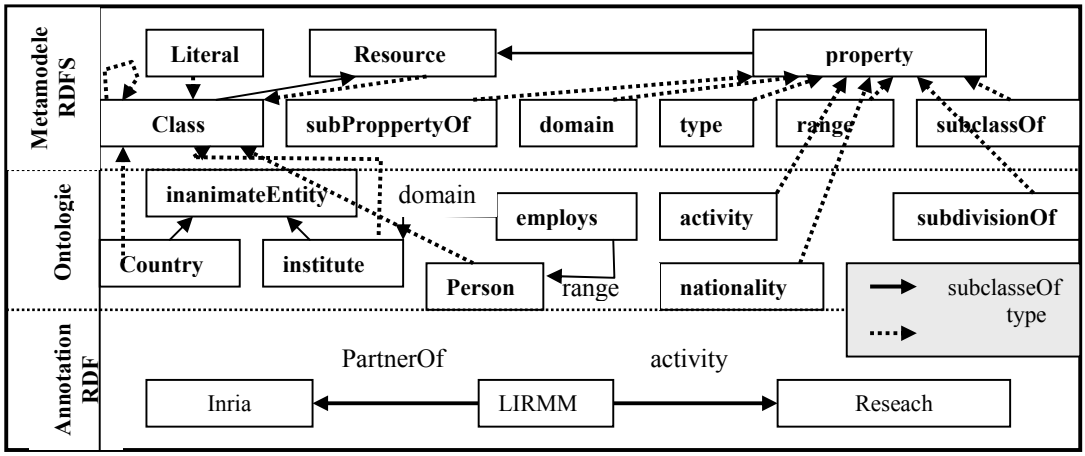


Figure 3. 7 : Le méta-modele de RDFS, une ontologie sous forme de schéma RDFS et une annotation RDF basée sur ce schéma

3.3.2.1. Vocabulaire de RDF et RDF Schema

Cette table présente l'ensemble de vocabulaire de RDF, réunissant le vocabulaire à l'origine défini dans les spécifications de modèle et de syntaxe de RDF avec les classes et les propriétés de RDF schéma.

1) Classes de RDF et RDFS

Class name	Comment
rdfs:Resource	The class resource, everything.
rdfs:Literal	The class of literal values, e.g. textual strings and integers.
Rdf:XMLLiteral	The class of XML literals values.
rdfs:Class	The class of classes.
Rdf:Property	The class of RDF properties.
rdfs:Datatype	The class of RDF datatypes.
Rdf:Statement	The class of RDF statements.
Rdf:Bag	The class of unordered containers.
Rdf:Seq	The class of ordered containers.
Rdf:Alt	The class of containers of alternatives.
rdfs:Container	The class of RDF containers.
rdfs:ContainerMembershipProperty	The class of container membership properties, rdf:_1, rdf:_2, ..., all of which are sub-properties of 'member'.
Rdf:List	The class of RDF Lists.

Table 3. 1: Classes de RDF [30]

2) Propriétés de RDF et RDFS

Property name	comment	domain	range
Rdf:type	The subject is an instance of a class.	rdfs:Resource	rdfs:Class
rdfs:subClassOf	The subject is a subclass of a class.	rdfs:Class	rdfs:Class
rdfs:subPropertyOf	The subject is a subproperty of a property.	rdf:Property	rdf:Property
rdfs:domain	A domain of the subject property.	rdf:Property	rdfs:Class
rdfs:range	A range of the subject property.	rdf:Property	rdfs:Class
rdfs:label	A human-readable name for the subject.	rdfs:Resource	rdfs:Literal
rdfs:comment	A description of the subject resource.	rdfs:Resource	rdfs:Literal
rdfs:member	A member of the subject resource.	rdfs:Resource	rdfs:Resource

Rdf:first	The first item in the subject RDF list.	rdf:List	rdfs:Resource
Rdf:rest	The rest of the subject RDF list after the first item.	rdf:List	rdf:List
rdfs:isDefinedBy	The definition of the subject resource.	rdfs:Resource	rdfs:Resource
Rdf:subject	The subject of the subject RDF statement.	rdf:Statement	rdfs:Resource
Rdf:predicate	The predicate of the subject RDF statement.	rdf:Statement	rdfs:Resource
Rdf:object	The object of the subject RDF statement.	rdf:Statement	rdfs:Resource

Table 3. 2 : Propriétés de RDF [30]

3.4.Conclusion

L'évolution d'un Web syntaxique vers un Web sémantique est un problème actuel. Le défi de cette évolution est de transformer le World Wide Web actuel, entièrement tourné vers la présentation des documents à des utilisateurs humains, vers un Web dont le contenu serait compréhensible aux machines.

L'utilisation du langage XML permet à chacun de créer ses propres balises qui entourent des pages Web ou des sections de texte. Les scripts ou les programmes peuvent alors utiliser ces balises de façon sophistiquée. En clair, XML permet aux utilisateurs d'ajouter des structures arbitraires à leurs documents sans rien dire de leur signification. A cet effet le langage RDF est introduit, il exprime le sens de ces balises. C'est encodé sous la forme de triplets, un peu comme sujet-verbe-complément d'une phrase très simple. Une description RDF consiste en un ensemble de déclarations spécifiant chacune la valeur d'une propriété d'une ressource. Une déclaration RDF est donc un triplet (ressource, propriété, valeur) où la valeur est soit une ressource soit un littéral. Cette technique des triplets permet aussi de créer des collections, c'est à dire des groupes de ressources auxquels s'appliquent diverses propriétés. La réification est enfin possible en transformant un triplet en ressource. Ce procédé permet donc d'attacher une propriété à un triplet.

RDF fournit un moyen pour annoter un document. C'est une application de XML dans le sens ou la syntaxe de RDF est définie en XML. RDF est donc un langage permettant de décrire les données d'un document. C'est-à-dire d'attacher des propriétés aux informations transportées.

Pour que une information soit utilisable par une machine, il faut auparavant la définir. Ceci est le rôle de RDF-Schemas, bientôt au stade de recommandation, qui permet de déclarer des classes de ressources, et d'indiquer la signature des propriétés, c'est à dire de contraindre les domaine et co-domaine des relations [34].

La vision du web sémantique s'appuie sur l'utilisation d'un langage de description commun, permettant d'exprimer à la fois des ontologies, rassemblant un ensemble de définitions de concepts et de rôles, et d'exprimer des annotations (par exemple sur les ressources du Web), utilisant le vocabulaire de ces ontologies, ce langage permettrait aux agents informatiques de comprendre les diverses annotations et de communiquer entre eux, en effectuant des raisonnements sur les concepts, Le langage qui semble aujourd'hui s'imposer est RDF : Ce langage possède à la fois une syntaxe graphique ainsi qu'une syntaxe du type triplet, ou ressource-attribut-valeur et correspond à peu près aux Graphes Conceptuels Simples. Cependant les mécanismes de définition de classes et de propriétés en RDFS sont élémentaires et ne permettent pas un réel raisonnement terminologique. RDFS permet essentiellement de créer des hiérarchies de classes et de propriétés, avec quelques contraintes sur la signature d'une propriété.

Sous les contraintes et les conditions de toutes ces exigences pour la réalisation du web sémantique, dont l'être humain peut pas agir à cause de la grande masse d'information, nous devons dans ce cas, utiliser des techniques d'apprentissage pour la construction des ontologies dans notre cas des ontologies RDFS pour permettre leur traitements automatique, en construisant les hiérarchies des concepts, et en utilisant les annotations RDF pour la description des ressources, ce que nous présentons dans le chapitre suivant .

CHAPITRE 04

ONTOLOGIES RDFS ET ANNOTATION DES RESSOURCES WEB PAR LE RDF.

4.1.Introduction

Dans le cadre du Web, les ontologies ont un rôle crucial à jouer pour le développement d'un Web sémantique fondé sur l'annotation sémantique des ressources du Web. Ce travail aborde précisément le problème de l'apprentissage d'ontologies à partir de RDF. De fait, plutôt que de faire appel à des techniques d'apprentissage qui exploitent les descriptions extraites de ces graphes, nous proposons après un passage depuis les graphes conceptuel vers le RDF et de tirer parti des annotations sémantiques des graphe RDF du Web et nous faisons appel à des techniques basées sur le regroupement conceptuel d'objets.

4.2.Correspondance entre les graphes conceptuels et le RDF

Le modèle de RDFS et celui des GCs partagent plusieurs caractéristiques communes, de sorte qu'une correspondance peut être établie entre RDF(S) et un large sous-ensemble du modèle des GCs Simples. Ceci n'empêche pas qu'ils existent certains points différents.

4.2.1.Les points communs

- Les connaissances terminologiques et celles assertionnelles sont séparées dans les deux modèles :

Examinons tout d'abord la partie terminologique des deux modèles :

La hiérarchie des classes (resp. propriétés) dans un schéma RDF correspond à la hiérarchie des types de concepts (resp. relations) dans un support de GCs ; de plus, et surtout, les propriétés RDFS sont des entités à part entière, au même titre que les classes RDFS, tout comme les types de relations dans le modèle des GCs sont déclarés indépendamment des types de concepts. C'est cette similitude des propriétés RDFS et des types de relations qui rend particulièrement pertinente la correspondance entre le modèle de RDF(S) et celui des GCs. Ce traitement similaire des

propriétés peut notamment être apposé à celui adopté dans les langages de représentation à objets où les propriétés sont déclarées à l'intérieur des classes

- Les connaissances assertionnelles dans les deux modèles, sont positives, conjonctives et existentielles.

- Une opération de réification est définie dans les deux modèles :

Elle permet de représenter des formules du langage par des objets : dans le modèle des GCs, un graphe G peut être réifié en un concept dont le marqueur est G; dans le modèle RDF, un triplet RDF peut être réifié en une ressource de type rdf : Statement et un ensemble de triplets en une ressource de type rdf : Bag regroupant des ressources de type rdf : Statement.

- Les connaissances assertionnelles sont représentées par des graphes orientés étiquetés dans les deux modèles. Un graphe RDF G peut être traduit en un graphe conceptuel GC comme suit :

– Un arc étiqueté par une propriété p dans G est traduit par un sommet relation de type dans GC.

– Un sommet étiqueté par une ressource identifiée dans G est traduit par un sommet concept individuel dans GC dont le marqueur est l'identificateur de la ressource ; le type de ce concept correspond à la classe à laquelle la ressource identifiée est liée par la propriété rdf :type dans G.

– Un sommet étiqueté par une ressource anonyme dans G est traduit par un sommet concept générique dans GC ; le type de ce concept correspond à la classe à laquelle la ressource identifiée est liée par la propriété rdf :type dans G.

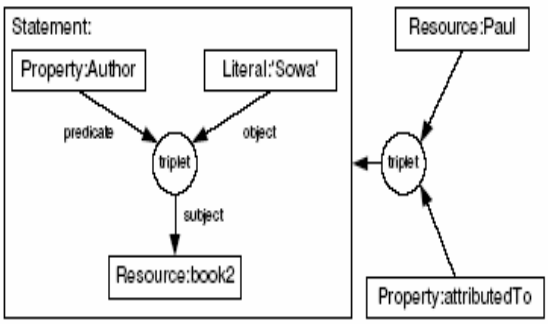
4.2.2.Les points différents

- Le modèle de données RDF ne supporte que des relations binaires tandis que le modèle des GCs supporte des relations n-aires. Cependant, une relation n-aire peut toujours être exprimée par des propriétés binaires en utilisant une ressource intermédiaire, les ressources reliées par la relation n-aire étant traduites par des ressources reliées à la ressource intermédiaire par des propriétés binaires .

- Le modèle de donnée RDF supporte la multi-instanciation tandis que celui des GCs ne supporte qu'une mono-instanciation. Cependant, la déclaration d'une ressource comme instance

de plusieurs classes dans le modèle RDF peut être traduite dans le modèle des GCs en générant le type de concept correspondant à la plus grande spécialisation des types de concepts traduisant ces classes.

Le tableau suivant récapitule les correspondances entre le GC et le RDF :

GC	RDF
types de concepts.	des classes.
types Relations : les types de relations dans le modèle des GCs sont déclarés indépendamment des types de concepts.	Propriétés : les propriétés RDFS sont des entités à part entière.
Réification : un graphe G peut être réifié en un concept dont le marqueur est G.  	Réification : un triplet RDF peut être réifié en une ressource de type rdf : Statement <rdf:Description> <rdf:subject resource='book2' /> <rdf:object resource='Sowa' /> <rdf:predicate resource='Author' /> <rdf:type resource='Statement' /> <a:attributedTo>Paul<a:attributedTo> </rdf:Description>
un sommet relation dans GC.	Un arc étiqueté par une propriété p dans G.
Un sommet concept individuel dans GC dont le marqueur ➔ l'identificateur de la ressource, le type de ce concept ➔ la classe à laquelle la ressource identifiée est liée par la propriété rdf :type dans G.	Un sommet étiqueté par une ressource identifiée dans G.
Un sommet concept générique dans GC, le type de ce concept ➔la classe	Un sommet étiqueté par une ressource anonyme dans G

à laquelle la ressource identifiée est liée par la propriété rdf :type dans G.	
le modèle des GCs supporte des relations n-aires.	Le modèle de données RDF ne supporte que des relations binaires.
GCs ne supporte qu'une mono-instanciation.	Le modèle de donnée RDF supporte la multi-instanciation.
le marqueur d'un concept est le nom de la ressource et le type de concept est le type de ressource. Type de ressouce: NomRessource	Le nom de ressource est le marqueur de concepts NomRessource
Type de ressource :*	Ressource anonyme

4.3.Description et représentation des concepts

4.3.1.Apprentissage des concepts

Les techniques de regroupement conceptuel essayent de former des classes rassemblant les objets similaires. La classification systématique a pour objectif de créer une hiérarchie contenant toutes les classes possibles, où les classes sont un couple intension et extension, les intensions sont formées à partir d'un langage de représentation donné. Dans notre travail le langage qui représenté les intensions est le langage RDF.

4.3.2.Apprentissage de concepts pour le Web Sémantique

Nous nous fixons pour objectif d'apprendre des concepts intéressants pour être ajoutés dans l'ontologie. La première différence avec les approches précédentes est la forme des données : au lieu d'un ensemble d'objets possédant chacun une description indépendante, les objets font ici partie d'un même graphe et n'ont pas de description propre. Il faut donc extraire pour chaque objet une description particulière à partir du graphe global. De plus, la présence d'une ontologie représentant la connaissance du domaine est à prendre en compte dans la recherche des concepts intéressants.

4.4.Présentation d'une ontologie par un schéma RDF

Un RDF Schema (RDFS) représente les connaissances ontologiques utilisées dans des déclarations RDF, c'est un ensemble de déclarations de classes et de propriétés. Chaque classe ou propriété est décrite par un ensemble de propriétés : parmi ces propriétés on trouve les plus fondamentales qui sont : subclassOf et type, pour représenter respectivement les relations de subsomption entre classes et les relations d'instanciation entre instances et classes. Dans la section suivante on représente quelques classe et propriété.

4.5.La création des RDFS

4.5.1.Les classes

Les ressources peuvent être divisées en groupes appelés classes, les membres d'une classe sont des instances de la classe .les classes sont elles même des ressources, il sont définie par [67], et peuvent être décrites en utilisant les propriétés de RDF. Rdf : type est une propriété utilisé pour savoir a quelle classe, cette ressource appartient.

4.5.1.1.Ressource

Toute choses décrit par RDF sont appelés ressources, et sont des instances de la classe rdfs : Ressource. C'est la classe de n'importe quelle chose. Toutes les autres classes sont des sous classes de cette classe.

```
<rdfs:Class rdf:about="http://www.w3.org/2000/01/rdf-
schema#Resource">
      <rdfs:isDefinedBy
    rdf:resource="http://www.w3.org/2000/01/rdf-  schema#"/>
      <rdfs:label>Resource</rdfs:label>
      <rdfs:comment>The class resource, everything.</rdfs:comment>
  </rdfs:Class>
```

4.5.1.2. Class : C'est la classe des ressources qui sont les classes de RDF.

```
<rdfs:Class rdf:about="http://www.w3.org/2000/01/rdf-
schema#Class">
  <rdfs:isDefinedBy
rdf:resource="http://www.w3.org/2000/01/rdf-      schema#"/>
  <rdfs:label>Class</rdfs:label>
  <rdfs:comment>The class of classes.</rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Resource"/>
    </rdfs:Class>
```

4.5.1.3. Statement

rdf:Statement est une instance de la classe rdfs:Class. Un RDF statement est une déclaration en utilisant les triplets de RDF (subject, prédicate, object) : subject est une instance de rdfs:Resource , predicate :est une instance de rdf:Property, object : est une instance de rdfs:Resource.

```
<rdfs:Class rdf:about="http://www.w3.org/1999/02/22-rdf-syntax-
ns#Statement">
      <rdfs:isDefinedBy   rdf:resource="http://www.w3.org/1999/02/22-
    rdf-syntax-ns#"/>
      <rdfs:label>Statement</rdfs:label>
      <rdfs:subClassOf   rdf:resource="http://www.w3.org/2000/01/rdf-
    schema#Resource"/>
      <rdfs:comment>The class of RDF statements.</rdfs:comment>
</rdfs:Class>
```

4.5.1.4. Bag

La classe rdf:Bag est la classe BAG des container RDF, est une sous classe de rdfs:Container , cette classe est utilisée conventionnellement pour indiquer que les éléments de cette classe sont désordonnées.

```
    <rdfs:Class rdf:about="http://www.w3.org/1999/02/22-rdf-syntax-
    ns#Bag">
      <rdfs:isDefinedBy rdf:resource="http://www.w3.org/1999/02/22-
    rdf-syntax-ns#"/>
      <rdfs:label>Bag</rdfs:label>
      <rdfs:comment>The class of unordered containers.</rdfs:comment>
      <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-
    schema#Container"/>
    </rdfs:Class>
```

4.5.2.Les propriétés

Une propriété RDF est définit comme une relation entre la ressource qui représente le subject et celle qui représente l'object.

4.5.2.1. Label

rdfs:label est une instance de rdf:Property , elle est utilisée pour indiquer un nom de la ressource. Le triplet de la forme : (R rdfs:label L) indique que L est un label pour R.

```
    <rdf:Property rdf:about="http://www.w3.org/2000/01/rdf-
    schema#label">
      <rdfs:isDefinedBy rdf:resource="http://www.w3.org/2000/01/rdf-
    schema#"/>
      <rdfs:label>label</rdfs:label>
      <rdfs:comment>A human-readable name for the
    subject.</rdfs:comment>
      <rdfs:domain rdf:resource="http://www.w3.org/2000/01/rdf-
    schema#Resource"/>
      <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-
    schema#Literal"/>
</rdf:Property>
```

4.5.2.2. Type

rdf:type est une instance de rdf:Property elle est utilisée pour indiquer que cette ressource est une instance d'une classe . Le triplet de la forme : (R rdf:type C) indique que R est une instance de C.

```
<rdf:Property rdf:about="http://www.w3.org/1999/02/22-rdf-syntax-
ns#type">
      <rdfs:isDefinedBy rdf:resource="http://www.w3.org/1999/02/22-
    rdf-syntax-ns#"/>
      <rdfs:label>type</rdfs:label>
      <rdfs:comment>The subject is an instance of a
    class.</rdfs:comment>
      <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-
    schema#Class"/>
      <rdfs:domain rdf:resource="http://www.w3.org/2000/01/rdf-
    schema#Resource"/>
</rdf:Property>
```

4.5.2.3.Domain

rdfs:domain est une instance de rdf:Property elle lie une propriété aux classes dont les membres peuvent avoir cette propriété.

```
    <rdf:Property rdf:about="http://www.w3.org/2000/01/rdf-
    schema#domain">
      <rdfs:isDefinedBy rdf:resource="http://www.w3.org/2000/01/rdf-
    schema#"/>
      <rdfs:label>domain</rdfs:label>
      <rdfs:comment>A domain of the subject property.</rdfs:comment>
     <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-
    schema#Class"/>
      <rdfs:domain rdf:resource="http://www.w3.org/1999/02/22-rdf-
    syntax-ns#Property"/>
</rdf:Property>
```

4.5.2.4. Range

rdfs:range est une instance de rdf:Property, elle lie une propriété a la classe dans la quelle cette propriété prend sa valeur .

```
    <rdf:Property rdf:about="http://www.w3.org/2000/01/rdf-
    schema#range">
      <rdfs:isDefinedBy rdf:resource="http://www.w3.org/2000/01/rdf-
    schema#"/>
      <rdfs:label>range</rdfs:label>
      <rdfs:comment>A range of the subject property.</rdfs:comment>
      <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-
    schema#Class"/>
      <rdfs:domain rdf:resource="http://www.w3.org/1999/02/22-rdf-
    syntax-ns#Property"/>
    </rdf:Property>
```

4.5.2.5. SubClassOf

rdfs:subClassOf est une instance de rdf:Property , elle est utilisé pour indiquer que toutes les instances d'une classe sont des instances d'une autre .

```
    <rdf:Property rdf:about="http://www.w3.org/2000/01/rdf-
    schema#subClassOf">
      <rdfs:isDefinedBy rdf:resource="http://www.w3.org/2000/01/rdf-
    schema#"/>
      <rdfs:label>subClassOf</rdfs:label>
      <rdfs:comment>The subject is a subclass of a
    class.</rdfs:comment>
      <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-
    schema#Class"/>
      <rdfs:domain rdf:resource="http://www.w3.org/2000/01/rdf-
    schema#Class"/>
</rdf:Property>
```

4.6.Présentation des graphes RDF

4.6.1.Extraction de descriptions de ressources

Dans le modèle RDF, l'ensemble des annotations déclarées sur le Web compose un graphe RDF *unique* qui regroupe les connaissances exprimées. Le modèle RDF ne permet donc pas d'exprimer des déclarations indépendantes les unes des autres. Il est de ce fait impossible d'identifier un sous-graphe de G qui représenterait la description d'une ressource.

On peut citer la définition de [19] qui a définit une description complète d'une ressource comme suit :

- Description complète. Une ressource R est *complètement décrite* par le sousgraphe de G contenant toutes les ressources qui peuvent être atteintes à partir de R par un chemin de propriétés. Formellement, la *description complète* d'une ressource R est le plus grand sous-graphe connexe de G contenant R. Elle peut être très large, potentiellement elle peut être égale au graphe RDF G entier.

4.6.2.Description de longueur n.

La description de longueur n d'une ressource R est le plus grand sous-graphe connexe de G contenant tous les chemins possibles de longueur égale ou inférieure à n, commençant ou finissant par R. La description Dn(R) d'une ressource R peut être définie par récurrence comme la jointure de Dn-1(R) avec les descriptions D1 des noeuds externes de Dn-1(R).

La Figure 4.1 présente l'extraction de deux descriptions partielles de la ressource CancerDuPomon à partir du graphe RDF que constitue l'ensemble des ressources, ces descriptions sont de longueurs 1 et 2.

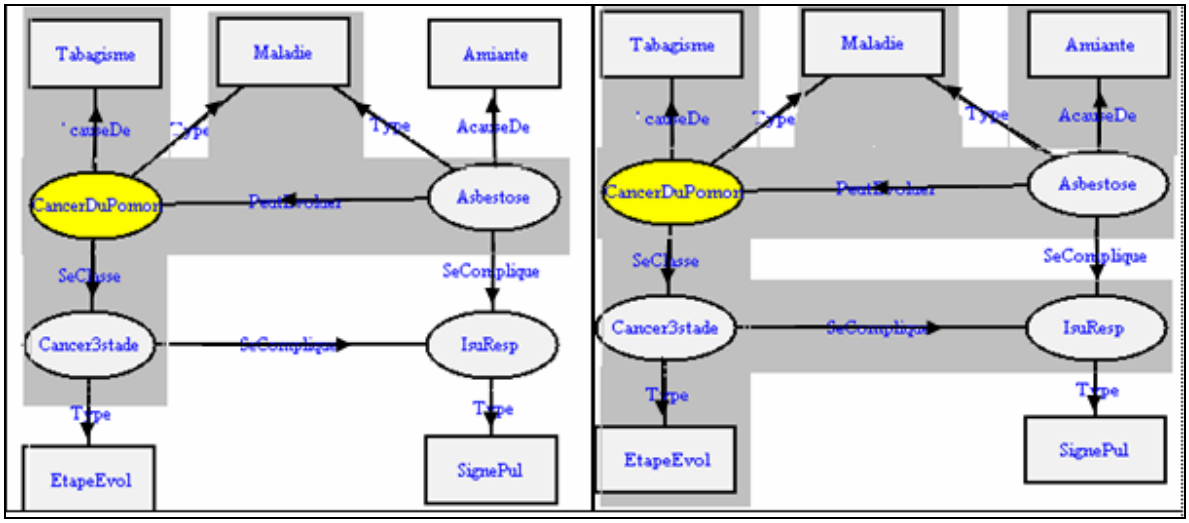


Figure 4. 1:descriptions partielles de la ressource CancerDuPomon

4.7.Conclusion

Le modèle de RDFS et celui des GCs ont plusieurs points communs, d'une manière à pouvoir établir un passage entre RDFS et le modèle des GCs Simples. Sans oublier qu'ils existent certains points différents.

Les schémas RDF permet de représenter des ontologies, à travers les définitions des classes et des relations qu'on peut trouvé dans un RDFS. Chacune des classes et des relations peut être décrite par des propriétés.

Dans notre méthode d'apprentissage on représente l'ontologie initial par un RDFS et un exemple d'apprentissage par un graphe RDF qui décrit un cas, pour définir ce graphe on doit utiliser les déclarations du RDFS initial. Une représentation détaillée de notre méthode d'apprentissage sera présentée dans le chapitre suivant.

CHAPITRE 05

LA CONCEPTUALISATION AUTOMATIQUE D'UNE ONTOLOGIE RDFS

5.1.Introduction

La construction d'ontologies demande la construction de la hiérarchie des concepts ainsi que celle des relations, nous nous intéressons par la hiérarchie des concepts, en utilisant l'apprentissage.

L'objectif de notre travail est la définition d'une méthodologie d'aide à la construction de taxonomie de concepts. L'élément principal de cette méthodologie est la construction automatique des concepts, nous nous intéressons essentiellement par les méthodes d'apprentissage dite de regroupement conceptuel permettant de construire une taxonomie à partir d'une taxonomie initial qui est présentée par le schéma RDF et un cas, c'est le graphe RDF définit en utilisant le schéma RDF.

Le but de ce chapitre est de présenter une méthode d'apprentissage d'ontologies à partir du langage RDF. Les hiérarchies apprises sont destinées à permettre au concepteur de l'ontologie initiale d'enrichir celle-ci en examinant les extensions des concepts appris.

5.2.L'algorithme d'apprentissage de concepts

Pour apprendre des concepts, il nous faut :

- choisir un langage de description de concepts. Pour permettre de présenter l'ontologie initiale, et tant que un de nos objectifs étant de pouvoir utiliser notre approche sur le Web Sémantique, le langage RDF est apparaît indispensable.
- extraire les descriptions des objets à partir du graphe initial global, c-a-d nous devons choisir des descriptions partielles des objets classés pour permettre de former des nouveaux concepts afin d'enrichir l'ontologie initiale.

Plus les descriptions extraites seront importantes, plus les concepts obtenus seront précis et spécifiques au domaine. Notre but est d'apprendre, à partir du graphe RDF que constitut une base d'annotations de ressources du Web, de nouvelles classes pour enrichir le schéma RDF représentant l'ontologie qui a été utilisé pour construire les annotations.

Dans ce travail la méthode d'apprentissage est incrémentale basant sur le principe Bottom-Up, à partir d'exemples. Nous montrerons qu'un algorithme Bottom-Up ayant un bon processus de spécialisation, qui lui garantit de rester dans l'espace des généralisations maximalement spécifiques, produira un meilleur comportement incrémental. Les méthodes incrémentales représentent une approche efficace vers le regroupement conceptuel, du coup elles sont bien adaptées au traitement de grands ensembles de données [54].

Notre approche de l'apprentissage d'ontologies consiste à considérer les concepts couvrant un ensemble de sommets ressources du graphe RDF. Chaque concept est défini en extension par un regroupement de ressources et sa définition en intension généralise les descriptions des ressources appartenant à son extension. Il s'agit de construire, à partir des descriptions de ressources extraites du graphe RDF, une hiérarchie H des concepts dont les extensions correspondent à tous les regroupements possibles de ces ressources.

5.3.Construction incrémentale d'une hiérarchie de généralisation

5.3.1.Construction de la hiérarchie H1

1. considérer initialement que chaque sommet ressource du graphe initial est une extension d'un concept.
2. extraction des graphes de description de longueur 1des sommets ressources, chaque concept est représenté en intension par son graphe de description.
3. calcul de la subsomption entre les graphes dont les sous graphes partiels sont de longueur 1 :

   3.1. extraction des sous graphes partiel du concept de longueur 1

   3.2. comparaison entre les sous graphes partiels :

   On dit que deux graphes peuvent être subsumé ssi :

   - il existe un sous graphe partiel qui est inclut dans les deux graphes initiaux ou

- ils existent deux sous graphes partiel qu'on peut les subsumés (généralisés) : soient G1 et G2 : ces deux sous graphes

- si P1=P2 alors Ps=P1, Si non on cherche Ps la propriété la plus spécifique qui subsume P1 et P2 en utilisant le shéma RDF

- Si V1 et V2 sont des classes, alors Vs est la classe la plus spécifique qui subsume V1 et V2 en utilisant le shéma RDF, sinon Vs=V1=V2 ou Vs est une ressource anonyme [19].

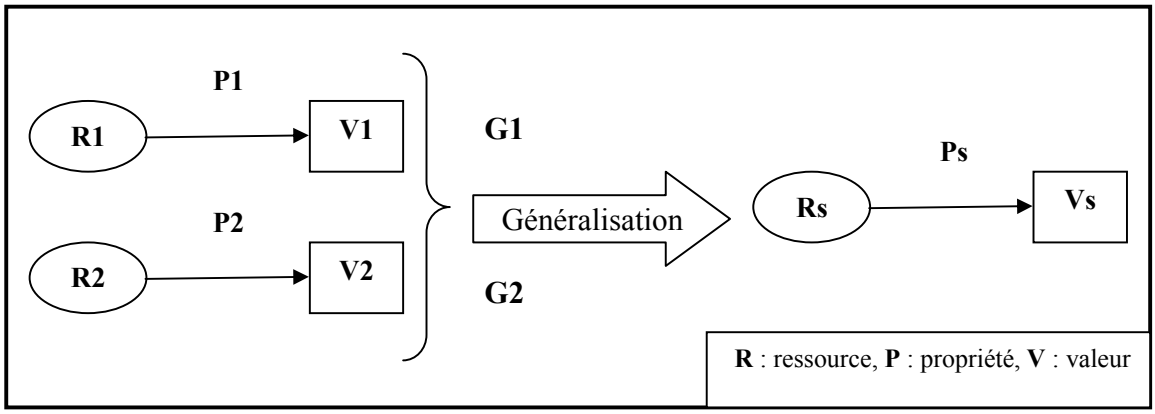


Figure 5. 1:Généralisation de deux sous graphes G1 et G2.

4. création d'un nouveau concept pour les graphes subsumés (longueur 1), le concept créé est dont l'extension est l'union des extensions du graphes initiaux, et dont l'intension est le graphe de subsumption.

5. jointure des graphes qui ont la même extension, pour chaque jointure : création d'un nouveau concept dont l'extension est l'une des deux extensions du concepts initiaux et dont l'intension est le graphe joint, élimination des concepts qui participent à la jointure.

6. construction de la hiérarchie entre les concepts par la relation d'inclusion entre les extensions.

5.3.2.Construction de la hiérarchie Hn

1. considérer initialement que chaque sommet ressource du graphe RDF initial est une extension d'un concept.

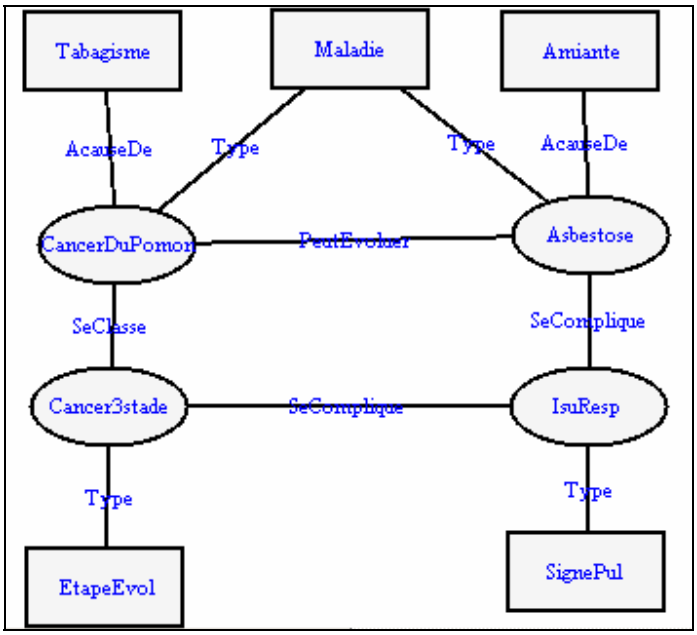
2. extraction des graphes de description de longueur n des sommets ressources, chaque concept est représenté en intension par son graphe de description.

3. calcul de la subsomption dont les sous graphes partiels sont de longueur n, entre les graphes dont leurs extensions appartient a uneextension d'un même concept dans la hiérarchie Hn-1.

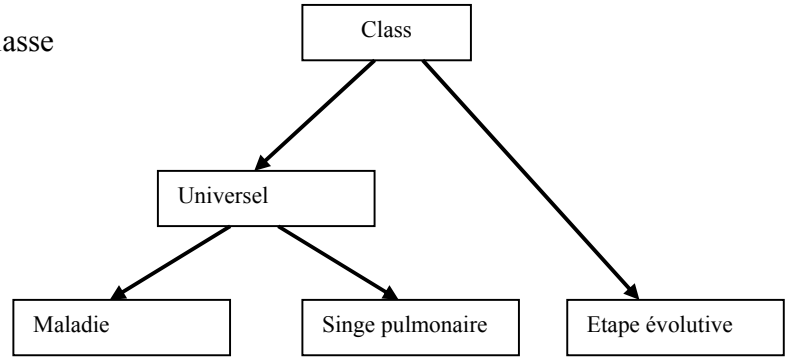
4. création de nouveau concept pour les graphes subsumés (longueur n), le concept créé est dont l'extension est l'union des extensions du graphes initiaux, et dont l'intension est le graphe de subsumption.
5. jointure des graphes des concepts créés avec les concepts de Hn-1, dont l'extension est la même (entre Hn et Hn-1). Pour chaque jointure : création d'un nouveaux concept, son extension est une des extensions des concepts joints et son intension et le graphe joint. Et élimination des concepts qui participent à la jointure.
6. construction de Hn : la même que Hn-1 dans les extensions et les intensions sont de Hn.

Nous présentons par la suite une application détaillée de cet algorithme sur le graphe et l'ontologie présentée par la hiérarchie de classes et de propriétés des figures suivante :

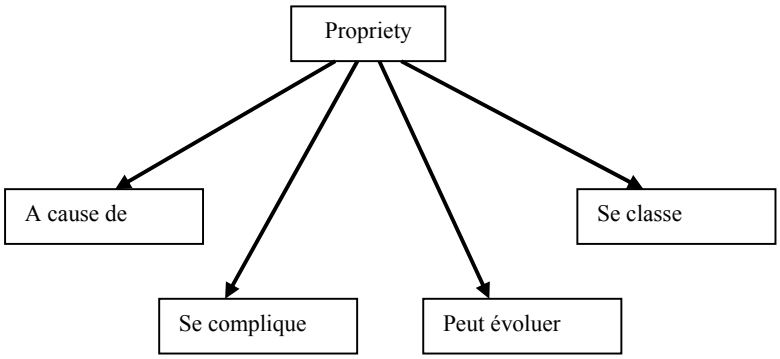
- Le graphe RDF



- Hiérarchie de classe



Hiérarchie des propriétés



5.4.Présentation détaillée de l'algorithme d'apprentissage

Dans cette section nous expliquons d'une manière détaillé notre algorithme on prend comme exemple le graphe RDF de la figure ci-dessus. Le principe de construction d'une hiérarchie de généralisation H1 à partir de descriptions de ressources de longueur 1 est le suivant :

5.4.2.Les étapes de H1

5.4.2.1.Formation des concepts de H1

1. considérer initialement que chaque sommet ressource du graphe initial est une extension d'un concept.

Dans notre graphe RDF présenté dans la figure ci-dessus, nous formons quatre concepts dont leurs extensions sont les sommets ressources :

Concept1 : {extension=Asbestose, intension= {}}

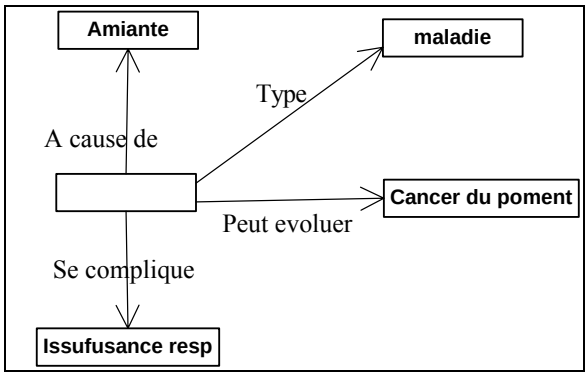
Concept2 : {extension=Cancer_Du_Poumon, intension= {}}

Concept3 : {extension=Cancer3stade, intension= {}}

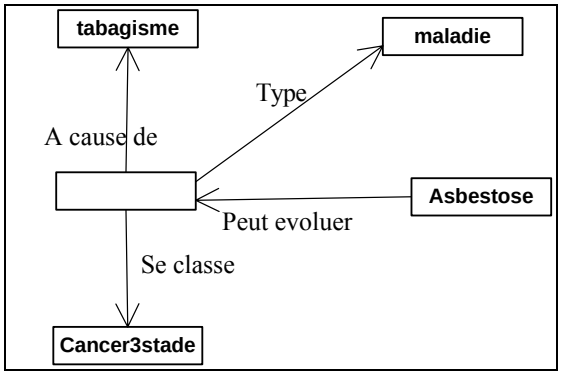
Concept4 : {extension= Insuffisance_Respiratoire, intension= {}}

2. construction des intensions des concepts formés à travers l'extraction des graphes de description de longueur 1 des sommets ressources, chaque concept est représenté en intension par son graphe de description.

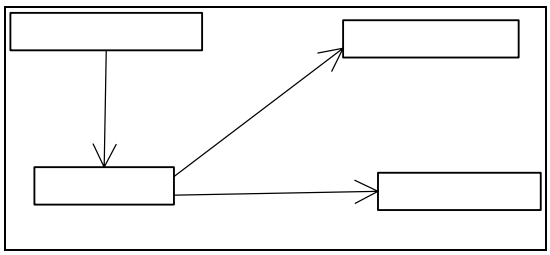
Le graphe de description de longueur 1 du Concept1 : {extension=Asbestose}



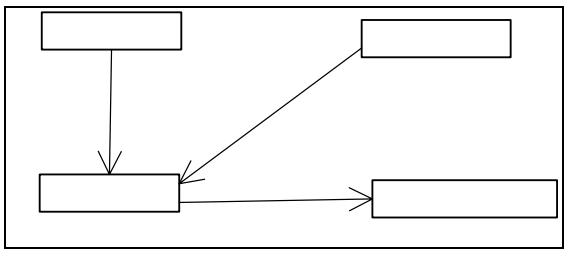
Concept2 : {extension= Cancer_Du_Poumon}



Concept 3 : {extension=Cancer3stade}



Concept4 : {extension=Insuffisance_Respiratoire}



3. calcul de la subsomption entre les graphes dont les sous graphes partiels sont de longueur 1 :

   3.1. extraction des sous graphes partiel du concept de longueur 1

   3.2. comparaison entre les sous graphes partiels :

Asbestose

On dit que deux graphes peuvent être subsumé ssi :

- il existe un sous graphe partiel qui est inclut dans les deux graphes initiaux ou

Se complique

- il existe deux sous graphes partiel qu'on peut les subsumés (généralisés)

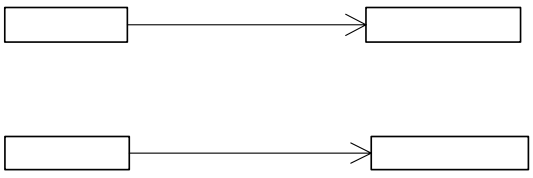
  - si p1=p2 alors ps=p1

  - Si non on cherche ps la propriété la plus spécifique qui subsume p1 et p2 en utilisant le shéma RDF

  - Si v1=v2 alors vs=v1

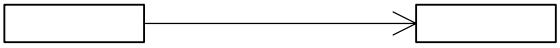
  - Si non vs est la classe la plus spécifique qui subsume v1 et v2 en utilisant le shéma RDF

Parmi les sous graphes du deux graphes de description du concept1 et concept2 (qui ont respectivement les extensions : Cancer_Du_Poumon et Asbestose), nous présentons seulement les sous graphes qu'on peut les généralisés Φ :

Sous graphes partiel	Extension	Graphe de subsomption
	Cancer_Du_Poumon Asbestose	

Donc création d'un nouveau concept5 :

Extension : Cancer_Du_Poumon, Asbestose

Intension : 

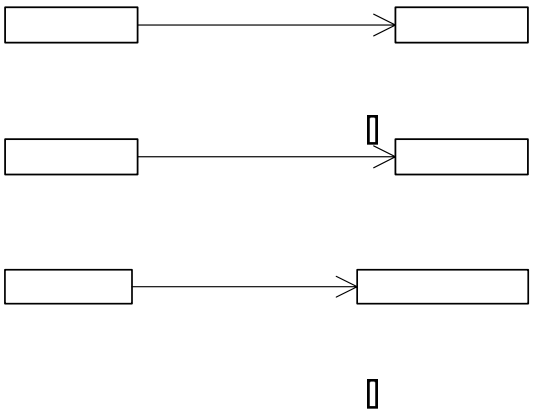
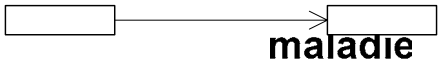
Sous graphes partiel	Extension	Graphe de subsomption
	Cancer_Du_Poumon Asbestose	

Donc création d'un nouveau concept6 :

Extension : Cancer_Du_Poumon, Asbestose

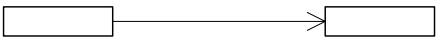
A cause de

Intension : 

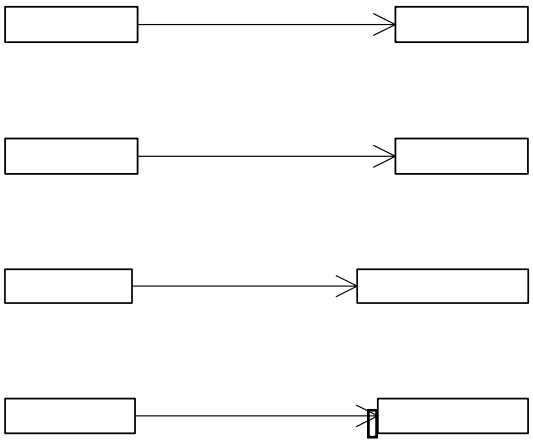
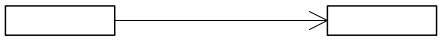
Sous graphes partiel	Extension	Graphe de subsomption
	Cancer_Du_Poumon Type Asbestose Insuffisance_Respiratoire Type	 maladie

Donc création d'un nouveau concept7 :

Extension : Cancer_Du_Poumon, Asbestose, Insuffisance_Respiratoire

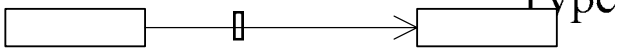
Intension : 

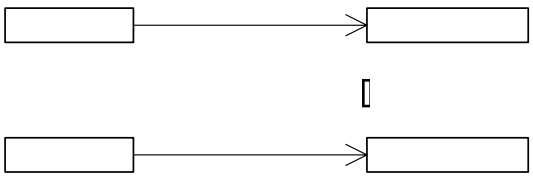
Type

Sous graphes partiel	Extension	Graphe de subsomption
	Cancer_Du_Poumon Asbestose Insuffisance_Respiratoire Type Cancer3stade	 maladie

Donc création d'un nouveau concept8 :

Extension : Cancer_Du_Poumon, Asbestose, Insuffisance_Respiratoire, Cancer3stade

Intension : 
Type
maladie

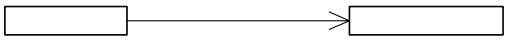
Sous graphes partiel	Extension	Graphe de subsomption
	Cancer3stade Type Asbestose	 SignePulmonaire

Donc création d'un nouveau concept9 :

Extension : Asbestose, Cancer3stade

Type

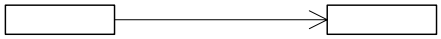
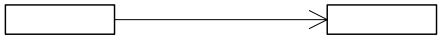
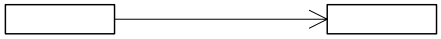
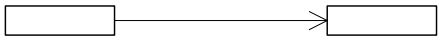
EtapeEvolutive

Intension : 

4. création d'un nouveau concept pour les graphes subsumés (longueur 1), le concept crée est dont l'extension est l'union des extensions du graphes initiaux, et dont l'intension est le graphe de subsumption.

Type

Les concepts formés a ce niveau :

Les concepts	Extension	Intension
Concept5	Cancer_Du_Poumon, Asbestose	
Concept6	Cancer_Du_Poumon, Asbestose	
Concept7	Cancer_Du_Poumon, Asbestose, Insuffisance_Respiratoire	
Concept8	Cancer_Du_Poumon, Asbestose, Insuffisance_Respiratoire, Cancer3stade	
Concept9	Asbestose, Cancer3stade	

5. jointure des graphes qui ont la même extension, pour chaque jointure : création d'un nouveau concept dont l'extension est l'une des deux extensions du concepts initiaux et dont l'intension est le graphe joint, élimination des concepts qui participent à la jointure. Si l'extension d'un concept est inclut dans une extension d'un autre, on fait la jointure et on créé un nouveau concept, mais on élimine seulement le concept qui a la plus petite extension et non pas le concept qui a la plus grande extension.

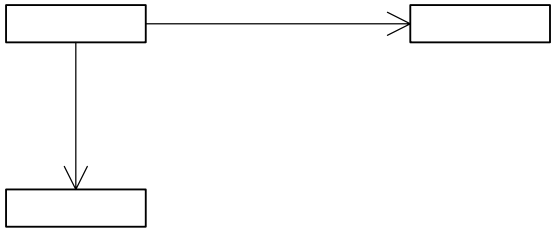
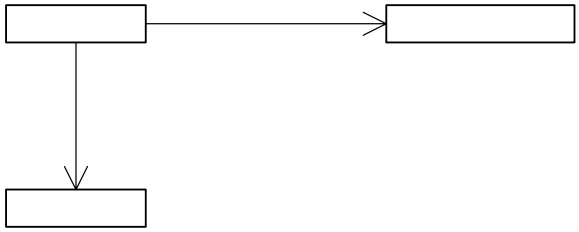
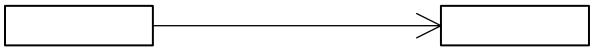
On prend un exemple de jointure entre les concepts : concept5 et concept6 (les deux concepts ont la même extension {Cancer_Du_Poumon, Asbestose}

Après jointure:

Elimination des concept concept5 et concept6 et création d'un nouveau concept : concept56 qui a :

L'extension : {Cancer_Du_Poumon, Asbestose} et l'intension : le graphe joint.

Apres l'application de toutes les jointures on obtient les concepts suivants :

Les concepts	Extension	Intension
Concept56	Cancer_Du_Poumon, Asbestose	
Concept89	Asbestose, Cancer3stade	
Concept7	Cancer_Du_Poumon, Asbestose, Insuffisance_Respiratoire	
Concept8	Cancer_Du_Poumon, Asbestose, Insuffisance_Respiratoire , Cancer3stade	

5.4.2.2.Construction de la hiérarchie entre les concepts formés

6. construction de la hiérarchie entre les concepts par la relation d'inclusion entre les extensions.

Suivant l'extension des concepts on trouve : Concept8, englobe toutes les ressources qui se trouvent dans les extensions des deux concepts : concept 7 et concept89, le concept89 englobe le

concept 56 et, à la fin les concepts de longueur 1 : concept1, concept2, concept3 et concept4. Suivant ces relations d'inclusion on trouve la hiérarchie de la figure suivante :

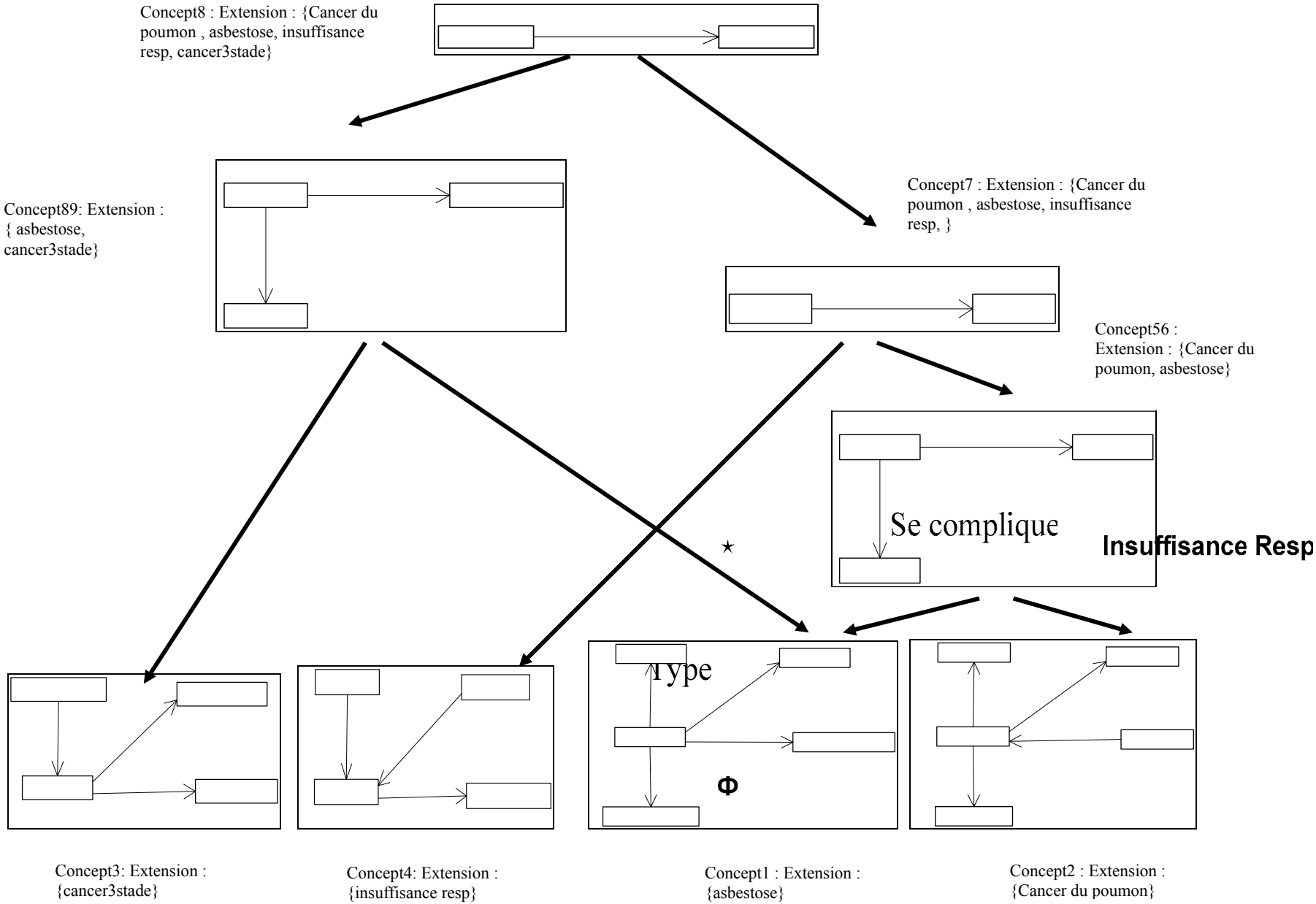


Figure 5. 2: La hiérarchie des concepts : longueur 1

Passant maintenant à la hiérarchie Hn:

5.4.3.Les étapes de Hn

5.4.3.1.Formation des concepts de Hn

1. considérer initialement que chaque sommet ressource du graphe RDF initial est une extension d'un concept.

Cette étape est similaire à la première partie de la hiérarchie H1 :

Concept1 : {extension=Asbestose, intension= {}}

Concept2 : {extension= Cancer_Du_Poumon, intension= {}}

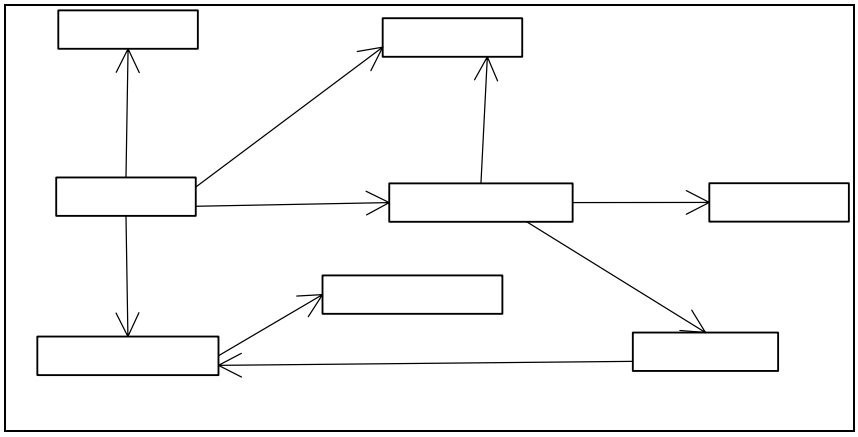
Concept3 : {extension=Cancer3stade, intension= {}}

Concept4 : {extension= Insuffisance_Respiratoire, intension= {}}

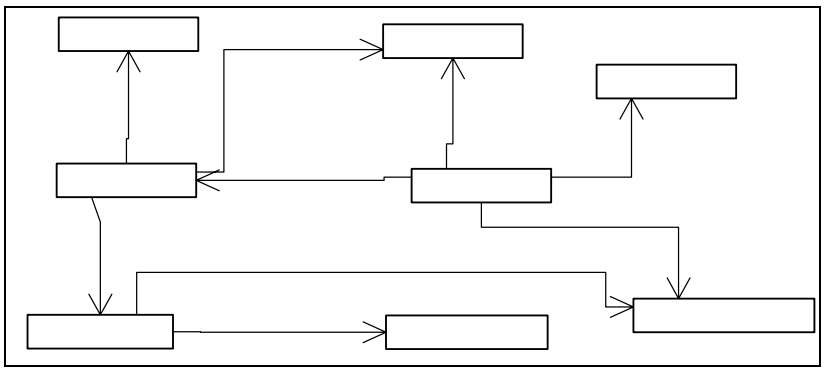
2. extraction des graphes de description de longueur n des sommets ressources, chaque concept est représenté en intension par son graphe de description.

Ici on parle de longueur 2 pour chaque sommet ressource :

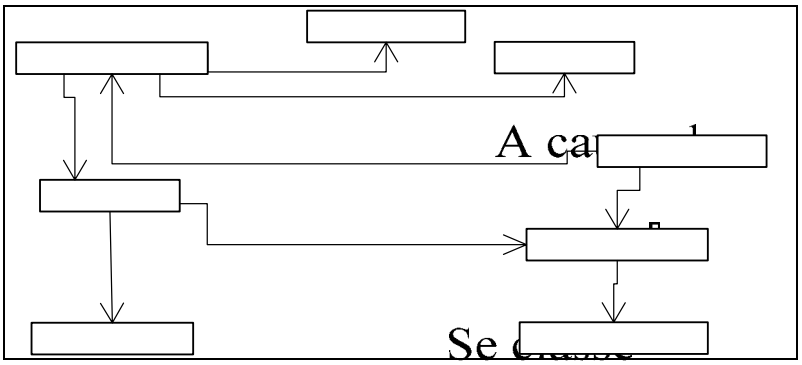
Le graphe de description de longueur 2 du Concept1 : {extension=Asbestose}



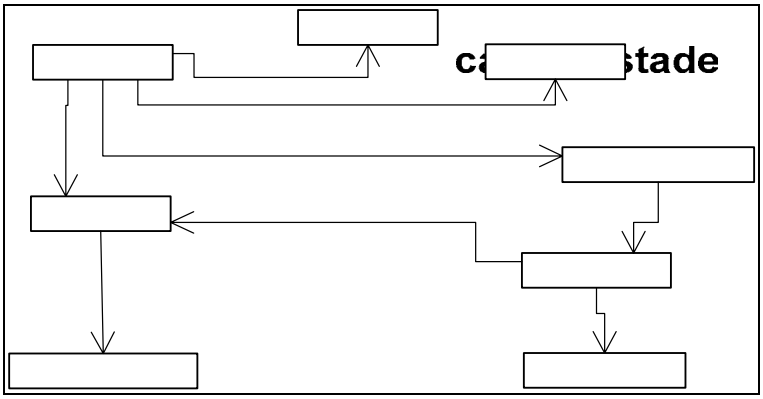
Concept2 : {extension= Cancer_Du_Poumon}



Concept 3 : {extension=Cancer3stade}



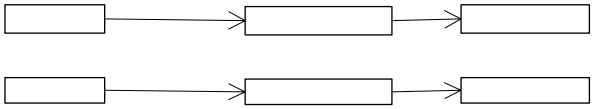
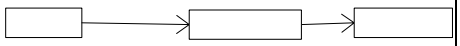
Concept4 : {extension=Insuffisance_Respiratoire}



3. calcul de la subsomption dont les sous graphes partiels sont de longueur n, entre les graphes dont leurs extensions appartiennent à une extension d'un même concept dans la hiérarchie Hn-1.

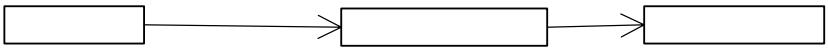
Apres la vérification entre les concepts qui sont dans leurs extension appartiennent à une même extension d'un concept de la hiérarchie H1, on trouve :

Le concept98 de la hiérarchie H1 : extension :{ Asbestose, Cancer3stade}, parmi les concept de la hiérarchie H2 qui ont une extension Asbestose et cancer3stade, si on cherche les graphes partiels de longueur 2 on trouve :

Sous graphes partiel	Extension	Graphe de subsomption
	Cancer3stade Asbestose	

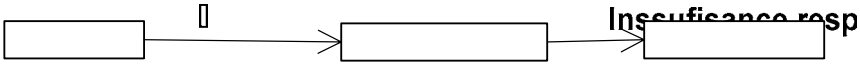
Donc création d'un nouveau concept5_2 :

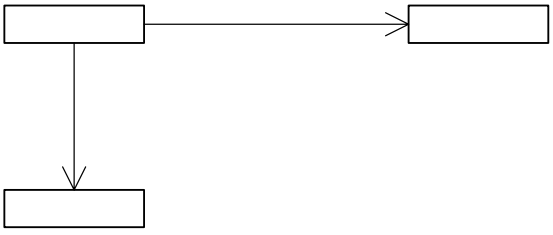
Extension : Cancer3stade, Asbestose

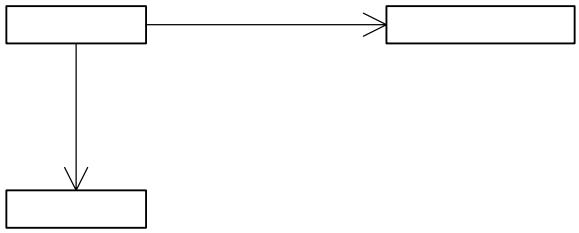
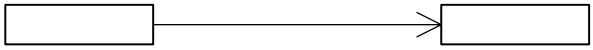
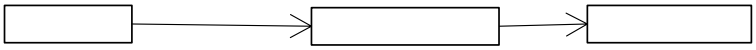
Intension : 

4. création de nouveau concept pour les graphes subsumé (longueur n), le concept crée est dont l'extension est l'union des extensions du graphes initiaux, et dont l'intension est le graphe de subsumption. Les concepts de Hn= les concepts de Hn-1 sauf qui ont dans leurs extension une seule ressource + les concepts créés.

On a un seul concept créé : concept5_2 : Extension : Cancer3stade, Asbestose

Intension : 

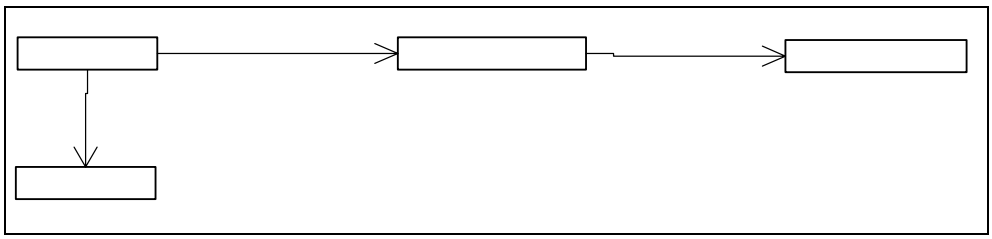
Les concepts	Extension	Intension
Concept56	Cancer_Du_Poumon, Asbestose	

Concept89	Asbestose, Cancer3stade	
Concept7	Cancer_Du_Poumon, Asbestose, Insuffisance_Respiratoire	
Concept8	Cancer_Du_Poumon, Asbestose, Insuffisance_Respiratoire, Cancer3stade	
Concept5_2	Asbestose, Cancer3stade	

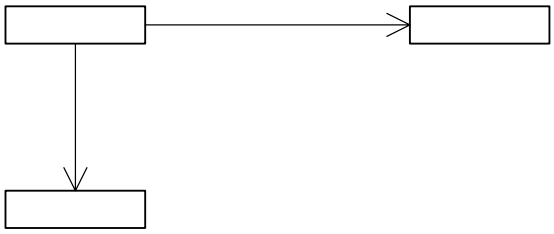
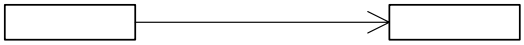
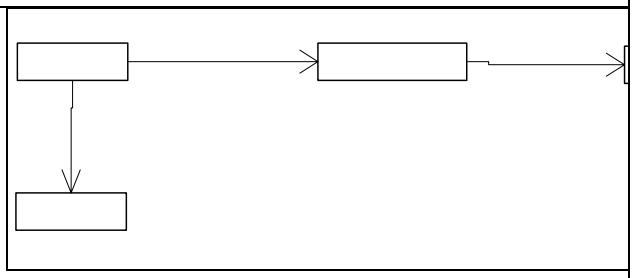
Typ

5. jointure des graphes des concepts créés avec les concepts de Hn-1, dont l'extension est la même (entre Hn et Hn-1). Pour chaque jointure : création d'un nouveaux concept, son extension est une des extensions des concepts joints et son intension et le graphe joint. Et élimination des concepts qui participent a la jointure.

Jointure des graphes de description pour les concepts : qui ont l'extension :{ Asbestose, Cancer3stade}



Création d'un nouveau concept Concept895_2 et élimination des concepts qui participent à la jointure le concept89 et le concept5_2, les concepts de H2 sont présentées dans le tableau suivant :

Les concepts	Extension	Intension
Concept56	Cancer_Du_Poumon, Asbestose	
Concept7	Cancer_Du_Poumon, Asbestose, Insuffisance_Respiratoire	
Concept8	Cancer_Du_Poumon, Asbestose, Insuffisance_Respiratoire , Cancer3stade	
Concept895$_2$	Asbestose, Cancer3stade	

5.4.3.2. Construction de Hn entre les concepts formés

6. construction de Hn : la hiérarchie de Hn est la même que Hn-1 dans les extensions des concepts et les intensions sont de Hn.

Les concepts de H2 sont les mêmes que de H 1dans les extensions mais dans les intensions, il y a 5 entre eux qui ont une intension plus complexe.

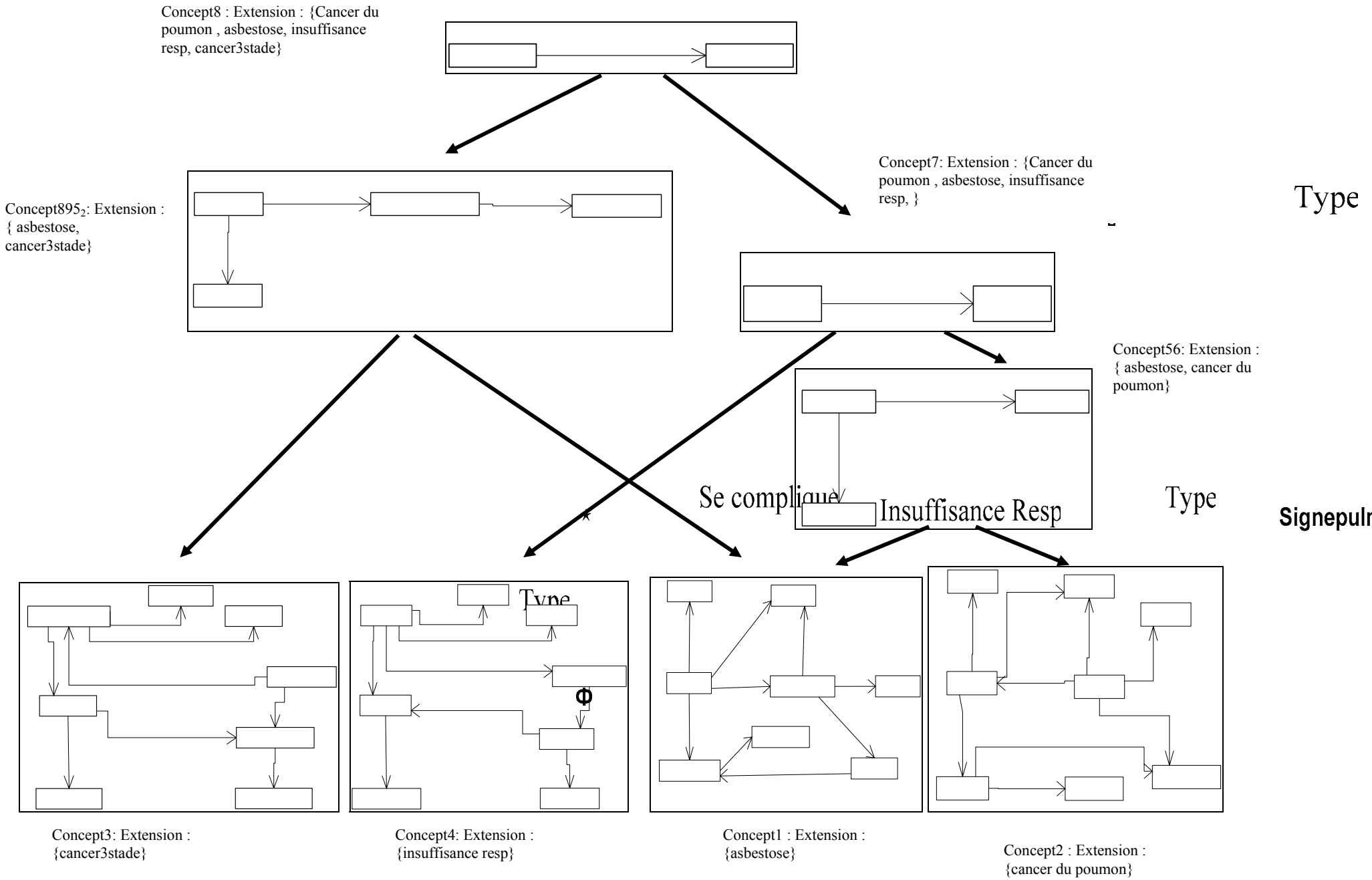


Figure 5. 3:La hiérarchie des concepts : longueur 2

5.5.Conclusion

L'apprentissage d'ontologies est un processus fondamental au sein des activités de maintenance. Il facilite en effet, l'acquisition de connaissances pour enrichir une ontologie et réduit, par voie de conséquence, le temps nécessaire à cette tâche.

Nous avons présenté un algorithme d'apprentissage de concepts permettant une conceptualisation automatique d'une ontologie. Cet algorithme est incrémental. Il consiste à construire une hiérarchie en utilisant des généralisations maximalement spécifiques entre les descriptions des concepts, ces description ce sont des annotations RDF extraites à partir un graphe RDF qui représente un cas d'apprentissage, pour enrichir l'ontologie RDFS initial par des concepts plus complexes et plus spécifique au domaine.

CHAPITRE 06

PRESENTATION DE L'OUTIL DE DEFINITION ET D'APPRENTISSAGE D'ONTOLOGIES (DLOT)

6.1.Introduction

La définition de méthodes et d'outils d'aide à la construction d'ontologies constitue un enjeu important pour les applications du Web Sémantique. Un tel outil est implémenté par apprentissage peut assurer de grands services et des réductions de coût aux ontologistes. Nous nous intéressons par l'apprentissage des concepts pour la construction des ontologies, c'est à partir d'une ontologie initiale exprimé en RDFS nous construisons un graphe RDF pour un cas d'apprentissage, puis nous appliquons l'algorithme d'apprentissage décrite dans le chapitre précédent pour obtenir une nouvelle classification des concepts et enrichir l'ontologie initiale.

Parmi les étapes de construction des ontologies, l'étape de conceptualisation est la plus importante, car elle nous permet d'obtenir la hiérarchie des concepts.

Dans ce travail, nous développons un outil de définition et d'apprentissage des concepts qui servira pour la construction des ontologies,

Pour réaliser notre outil d'apprentissage nous avons procédé comme suit:

Dans une première étape nous avons commencé par la spécification des besoins, dans cette étape nous devons développer des certains points important vu dans le premier chapitre, entre autre de limiter les fonctionnalités de l'outil que nous devons implémenter par la détermination des utilisateurs et des cas d'utilisations.

Puis nous passons à la présentation de notre outil DLOT en montrant les différentes interfaces pour les fonctionnalités offertes par notre outil tels que la définition et l'apprentissage des ontologies. L'implémentation de DLOT est réalisée en utilisant le langage C#.

6.2.La spécification des besoins

Comme il est cité au premier chapitre dans le processus de construction d'ontologies, dans la première étape il faut décrire: l'objectif opérationnel, le domaine de connaissance, les utilisateurs, la portée d'ontologie et les sources d'information où il faut décrire les sources de connaissances. Cette section a pour but de décrire chacun de ces éléments :

- L'objectif opérationnel

Notre thème de recherche se situe dans la problématique de l'annotation d'information. Notre but est d'annoter les ressources du Web, ces annotations sont extraites à partir de graphe RDF que constitue l'ensemble des annotations de ces ressources,ceci pour rendre le contenu de ces ressources compréhensible et exploitable par les machines. Ces annotations sémantiques permettrent plusieurs utilisations entre autre :

- l'amélioration de la pertinence des résultats obtenus suite à une recherche sur Internet via un moteur de recherche, la recherche sera plus précise qu'une recherche dite classique (recherche par mots clefs par exemple) car elle combine à la fois la structure du document et son contenu.
- L'utilisation de l'ontologie apprisse par des agents suivant leurs divers besoins. On peut donc parler du Web sémantique comme d'une extension du Web actuel qui permet aux humains et aux machines de travailler en coopération.
- La communication antre agent, en effet deux agents différents peuvent utilisée la même ontologie d'un domaine, et de même la communication entre agent logiciel et humains

Nous proposons, à partir d'ontologies contenant des concepts primitifs, d'exploiter les annotations utilisant ces ontologies pour repérer de nouveaux concepts définis. Il s'agit de former à partir de la base d'objets des concepts d'inclure dans l'ontologie, et d'organiser ces concepts dans une structure hiérarchique suivant la relation de généralisation définie sur les intensions et les extensions.

- Le domaine de connaissance

Notre travail est l'apprentissage des concepts à partir d'un schéma RDF et d'un graphe RDF qui utilise ce schéma, l'outil qui doit être implémenté permet la création d'une nouvelle ontologie, ainsi qu'un nouvel graphe RDF, donc l'utilisateur de notre outil se charge de les créer, d'où le choix pour n'importe quel domaine d'application.

Pour la connaissance exprimée dans l'ontologie elle est de nature connaissance de domaine du fait qu'elle exprime les concepts du domaine et les relations entre eux. Ces concepts reflètent ce que le domaine peut contenir.

- La portée d'ontologie

Dans cet élément de spécification nous devons citer les termes de l'ontologie, et tant que notre outil est destiné pour la création des nouvelles ontologies, donc nous (en tant que concepteur) ne pouvons pas déterminer la liste des termes à priori que doit contenir l'ontologie, en effet c'est la tâche de l'utilisateur

- Les sources d'information : annotation RDF

Le but de notre travail est l'évolution des ontologies, en effet, c'est à partir d'un schéma RDF créé par un utilisateur ainsi qu'un graphe RDF, que l'évolution ou l'enrichissement se s'effectue. En utilisant l'apprentissage pour annoter les ressources à partir des descriptions de ces ressources extraites du graphe RDF. Donc les sources d'information sont l'ontologie RDFS initiale et le graphe RDF utilisé pour extraire les descriptions des ressources.

- Les utilisateurs

Pour déterminer les utilisateurs de notre outil nous avons suivé la notation UML pour la détermination des cas d'utilisations et les utilisateurs.

6.2.1.Les Cas d'utilisation

L'étude des cas d'utilisation débute par la détermination des acteurs (catégories d'utilisateurs) du système.

6.2.1.1. Définition des acteurs

Notre système "l'outil de définition et d'apprentissage des ontologies DLOT [5]" est destiné à un utilisateur qui souhaite effectuer l'apprentissage des concepts pour la construction des ontologies :

- soit à partir de 0, en créant le Schéma RDF qui représente l'ensemble des classes et des propriétés du domaine, dans ce cas l'acteur chargé de cette fonction est un *ontologiste* ou un *ingénieur de connaissance* (un spécialiste du langage RDFS).
- soit, pour enrichir une ontologie décrite par un Shema RDF, par de nouveaux concepts, l'enrichissement de l'ontologie nécessite l'introduction d'un graphe RDF définit en

[5]DLOT **:**Definition and Learning Ontology Tool

utilisant le Schema RDF initiale, ce cas se réalise par un *expert* de domaine pour la définition de graphe RDF, car la construction de ce graphe demande la connaissance des concepts d'un domaine.

Ainsi, un autre cas d'utilisation est la consultation de l'ontologie pour l'utiliser dans divers usages soit par des spécialistes (communauté des spécialistes humains) du domaine, par exemple pour savoir la définition d'un certain concepts, soit par une communauté d'agents pour la communication entre eux, ils peuvent utilisés la même ontologie, ce qui induit la présence d'un autre rôle, donc acteur, qui configure le système. Il s'agit alors d'un acteur secondaire que nous nommons *opérateur*.

6.2.1.2. Diagramme de cas d'utilisation

Les différentes fonctionnalités offertes par notre outil forment ainsi un ensemble de *cas d'utilisation* (*"Use Case"*), exprimés par le diagramme suivant :

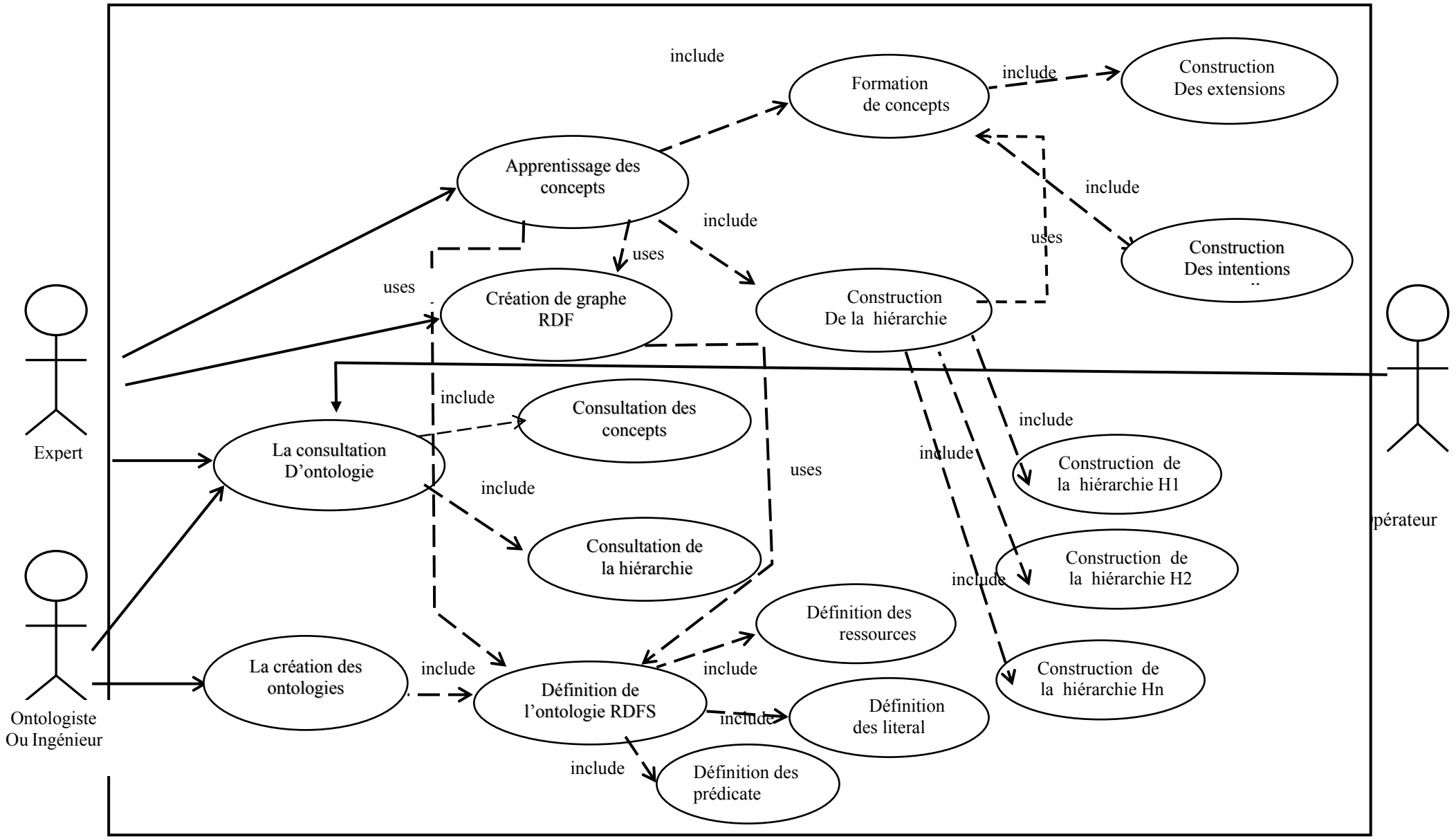


Figure 6. 1: Le diagramme de cas d'utilisation de DLOT

6.3.Présentation de DLOT

Notre outil de définition des ontologie permet de définir des ontologies de type RDFS, et par conséquence la définition des classes et des propriétés et les propriétés de chacune des classes et des propriétés.

6.3.1.L'ontologie RDFS

6.3.1.1. Création des ontologies RDFS

La définition d'une ontologie RDFS, consiste à définir un namespace, en quelque sorte on va définir un schéma en définissant l'ensemble des classes de ce schéma, ainsi que ses propriété. La définition des namespaces est réalisée comme il est apparu dans la figure suivante :

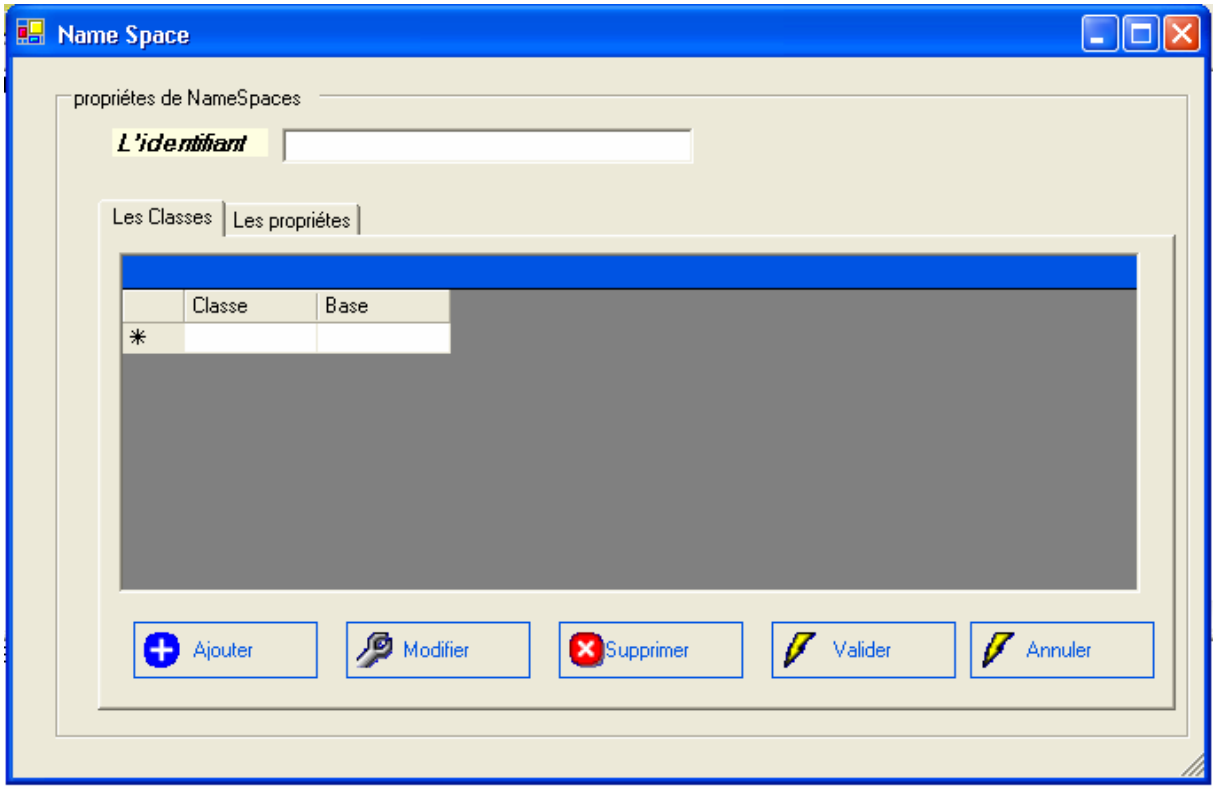


Figure 6. 2 : La définition des namesspaces

Premièrement on donne un nom pour le namespace (l'ontologie RDFS), on a aussi deux onglets, une pour les classes et autre pour les propriétés, on peut ajouter, modifier, supprimer des classe et des propriétés. Les classes et les propriétés sont enregistrées dans des tables (la classe/la propriété et la base de cette classe/ propriété c-a-d la super classe/ propriété)

6.3.1.2. La définition des classes et des propriétés

Pour définir des classes on doit donner un nom pour la classe à définir, et des valeurs pour leurs propriétés comme il est illustré dans les figures suivantes :

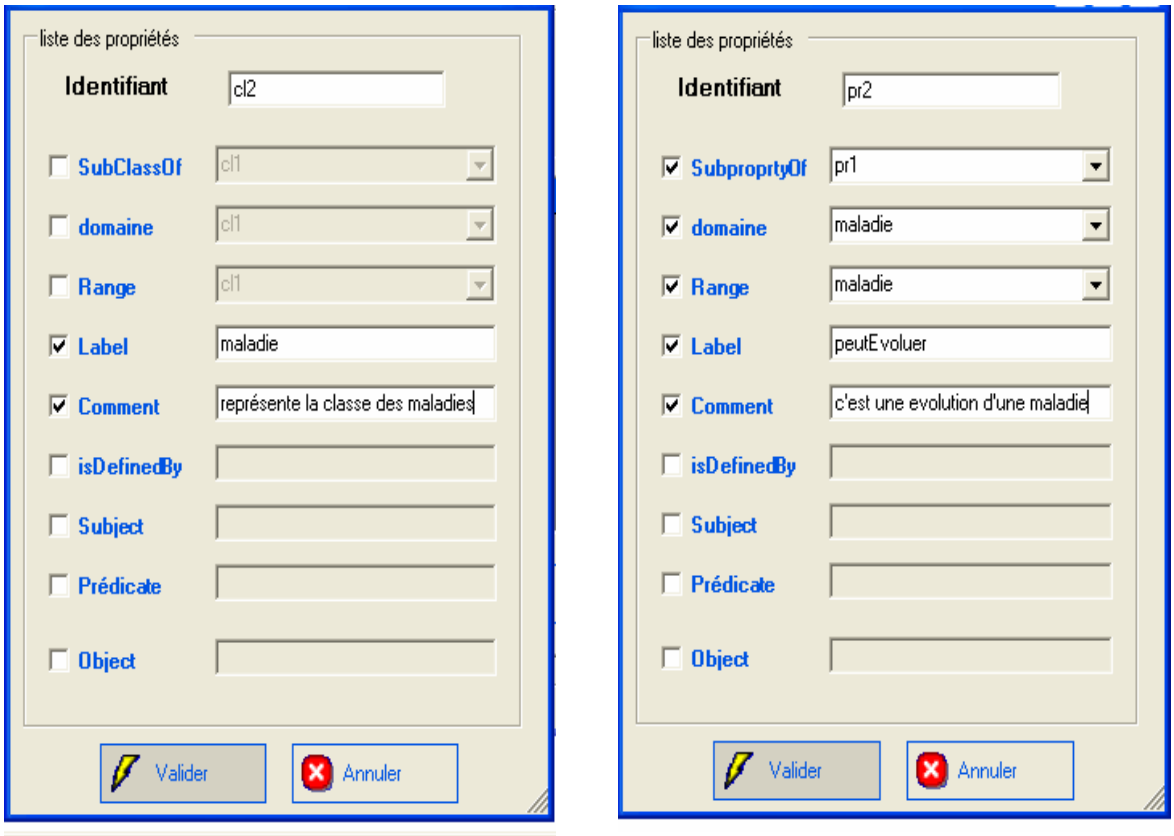


Figure 6. 3: La définition des classes et des propriétés.

6.3.1.3. Enregistrement des ontologies RDFS

L'enregistrement des ontologies RDFS est effectué par notre outil sous format *.Srdf , la syntaxe de ce fichier se base sur la syntaxe de XML, en ajoutant les spécificité de RDF. Nous présentons par la suite un exemple d'ontologie RDFS :

```
<?xml version="1.0"?>
<Environement le_nom="Fichier RDF " unite="21">
  <RDFS>
    <name_space le_nom="S_Med">
      <Classes>
        <Classe ID="universel">
          <Label>classe universel</Label>
          <Comment>represente une entité universel</Comment>
        </Classe>
        <Classe ID="signe-pulmonaire" SubClassOf="universel">
          <Label>la classe des signe pulmonaire</Label>
        </Classe>
```

```
        <Classe ID="maladie" SubClassOf="universel">
          <Label>la classe malidie</Label>
          <Comment>reppresente la classe des maladie</Comment>
        </Classe>
        <Classe ID="Etape_Evolutive">
          <Label>Etape Evolutive</Label>
          <Comment>represente les deffirentes etapes d'evolution des maladies</Comment>
        </Classe>
      </Classes>
      <Prorietes>
        <Propriete ID="Se_Classe">
          <Domain>maladie</Domain>
          <Range>Etape_Evolutive</Range>
        </Propriete>
        <Propriete ID="A_Cause_de">
          <Domain>maladie</Domain>
          <Label>a cause de </Label>
          <Comment>decrit les cause des maladies</Comment>
        </Propriete>
        <Propriete ID="se_Complique">
          <Domain>maladie</Domain>
          <Label>se complique</Label>
          <Comment>decrit la complication d'une maladie</Comment>
        </Propriete>
        <Propriete ID="Peut_Evoluer">
          <Domain>maladie</Domain>
          <Range>maladie</Range>
          <Label>peut evoluer</Label>
          <Comment>decrit les evolution possibles d'une maladie</Comment>
        </Propriete>
        <Propriete ID="Type">
          <Domain>universel</Domain>
          <Range>universel</Range>
          <Label>type</Label>
          <Comment>les types des objet</Comment>
          <IsDefinedBy>www.w3c-rdf.com</IsDefinedBy>
        </Propriete>
      </Prorietes>
    </name_space>
  </RDFS>
</Environement>
```

Figure 6. 4: Un exemple d'une ontologie RDFS

6.3.1.4.Ouverture des ontologies RDFS

Une fois l'ontologie est enregistrée, on peut l'ouvrir pour de multiples utilisations, l'ouverture de cette ontologie permet de la représenter dans un explorateur qui contient le namepace, les classes, les propriétés et leurs attributs avec leurs valeurs comme présentent les figures suivantes :

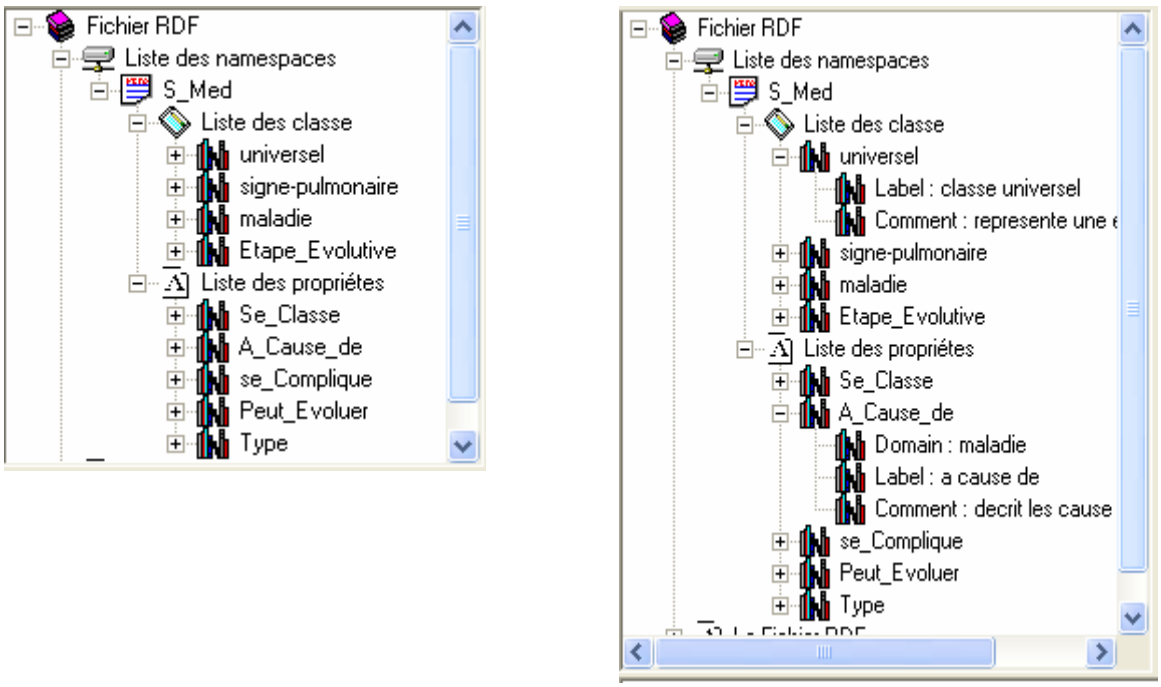


Figure 6. 5: Une ontologie RDFS présenté dans l'explorateur de l'outil.

6.3.2. Les graphes RDF

6.3.2.1.Construction des graphes RDF

Pour construire un graphe RDF, on doit utiliser une ontologie RDFS pour permettre d'instancier ses différentes classes et propriétés. Un graphe RDF est une représentation graphique d'un ensemble des triplet (ressource, propriété, littéral), pour chaque élément de ce triplet nous avons développé un composant : composant *Classe* qui présente une ressource, composant *Predicate* : c'est une transition entre la classe et le litteral, et composant *Litteral*.

- Construction des classes

Pour construire des classes nous devons utiliser le composant *Classe* en cliquant sur ce composant et le placer sur le panneau d'affichage des graphe, une boite de dialogue s'affiche pour permettre à l'utilisateur de choisir les propriétés de cette classe, il s'agit du nom de ce composant et la classe de l'ontologie RDFS dont notre composant est une instance, comme présente la figure :

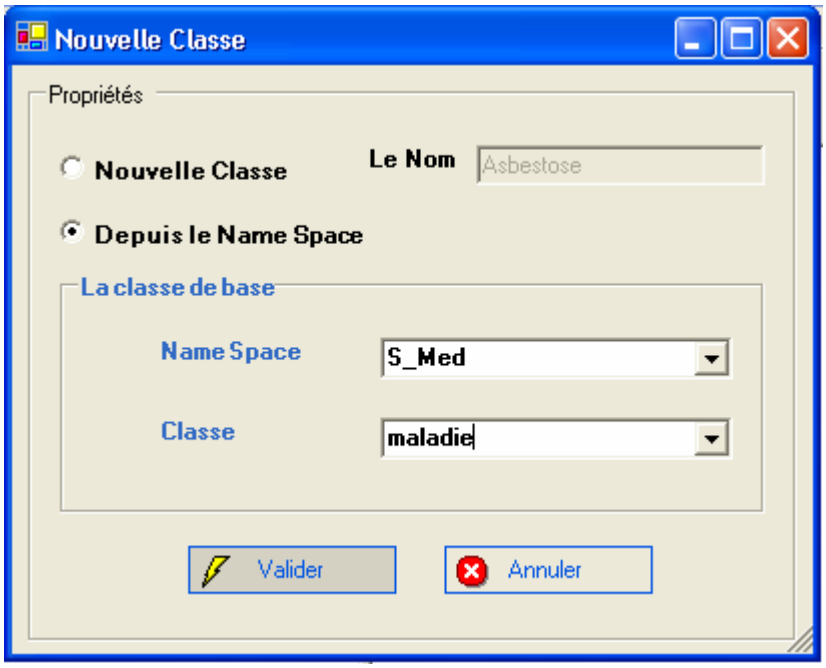


Figure 6. 6 : Propriétés du composant Classe.

Une fois l'utilisateur choisit les propriétés nécessaires, la classe sera affichée sur le panneau en spécifiant son nom. L'affichage de cette classe sera comme suit :

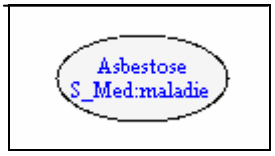


Figure 6. 7: Affichage d'une classe.

- Construction des Litteral

Pour construire un litteral nous utilisons le composant *Litteral*, en cliquant sur ce composant et le placer sur le panneau d'affichage des graphe, une boite de dialogue s'affiche pour permettre à l'utilisateur de choisir les propriétés de ce litteral, il s'agit du nom de ce composant et la classe de l'ontologie RDFS dont notre composant est une instance. Comme présente la figure :

Figure 6. 8 : Propriétés du composant Litteral.

Une fois l'utilisateur choisit les propriétés nécessaires, le litteral sera affiché sur le panneau en spécifiant son nom, l'ontologie RDFS utilisée et la classe instanciée. L'affichage de ce litteral sera comme suit :

Figure 6. 9: Affichage d'un litteral.

- Construction des predicate

Une predicate est une transition permettant de relier un composant Classe à un composant Litteral. Pour construire une predicate nous utilisons le composant ***Predicate*** en cliquant sur ce composant puis glisser la sourie du premier composant qui est la classe vers le deuxième composant qui est le litteral, une boite de dialogue s'affiche pour permettre à l'utilisateur de choisir les propriétés de cette predicate, il s'agit du nom de ce composant et la propriété de l'ontologie RDFS dont notre composant est une instance. Comme illustre la figure :

Figure 6. 10: Propriétés du composant Predicate

Une fois l'utilisateur choisit les attributs nécessaires, la predicate sera affiché sur le panneau en spécifiant son nom, l'ontologie RDFS utilisée. L'affichage de cette predicate sera comme suit :

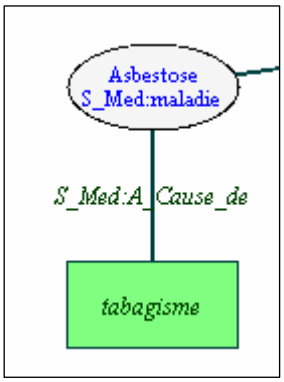


Figure 6. 11: Affichage d'une Predicate.

Une fois l'utilisateur édite son graphe en créant l'ensemble des classes, des litteral et des predicate nécessaires, il peut visualiser son graphe, comme le montre la figure :

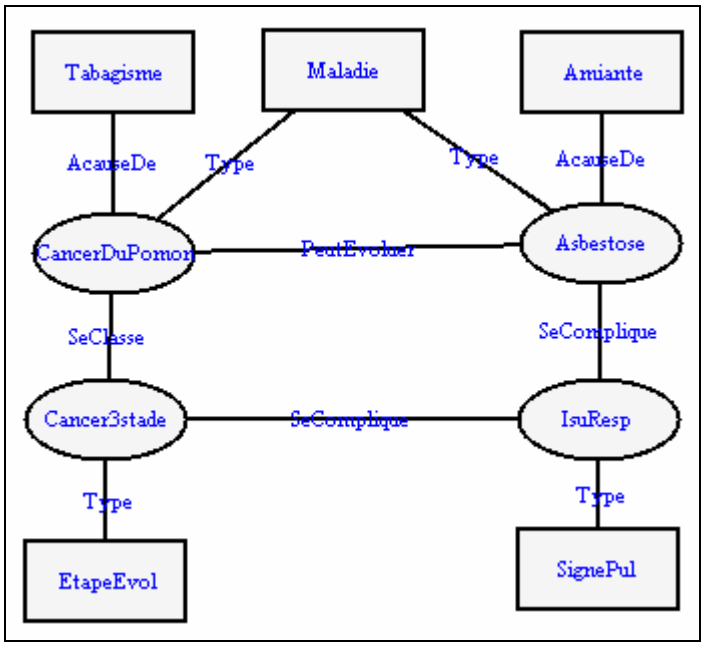


Figure 6. 12: Affichage d'un graphe RDF.

6.3.2.2. L'enregistrement des graphes RDF

L'enregistrement de graphes RDF est effectué soit d'une manière graphique ou textuelle. Graphiquement comme il est présenté dans la figure ci-dessus, et textuellement en sous format *.Grdf, la syntaxe de ce fichier se base sur la syntaxe de XML. Nous présentons par la suite un exemple textuel du même graphe de la figure 6.12 :

```
<?xml version="1.0"?>
<Environement le_nom="Fichier RDF " unite="21">
  <RDF m_nom="">
    <Classes>
      <Classe m_namespace="S_Med" Caption="Cancer-poment S_Med:maladie">
        <Centre>104;165</Centre>
        <Font>Times New Roman;8;false;false;false;false</Font>
        <Type>Cycle</Type>
        <Couleur_Interface>255;179;254;129</Couleur_Interface>
        <Couleur_Trace>255;0;128;0</Couleur_Trace>
        <Couleur_text>255;0;0;160</Couleur_text>
      </Classe>
      <Classe m_namespace="S_Med" Caption="Asbestose S_Med:maladie">
        <Centre>370;164</Centre>
        <Font>Times New Roman;8;false;false;false;false</Font>
        <Type>Cycle</Type>
        <Couleur_Interface>255;179;254;129</Couleur_Interface>
        <Couleur_Trace>255;0;128;0</Couleur_Trace>
        <Couleur_text>255;0;0;160</Couleur_text>
      </Classe>
      <Classe m_namespace="S_Med" Caption="Cancer3stde S_Med:maladie">
        <Centre>104;271</Centre>
        <Font>Times New Roman;8;false;false;false;false</Font>
        <Type>Cycle</Type>
        <Couleur_Interface>255;179;254;129</Couleur_Interface>
        <Couleur_Trace>255;0;128;0</Couleur_Trace>
        <Couleur_text>255;0;0;160</Couleur_text>
      </Classe>
      <Classe m_namespace="S_Med" Caption="Inssufisance-res S_Med:maladie">
        <Centre>370;269</Centre>
        <Font>Times New Roman;8;false;false;false;false</Font>
```

```
      <Type>Cycle</Type>
      <Couleur_Interface>255;179;254;129</Couleur_Interface>
      <Couleur_Trace>255;0;128;0</Couleur_Trace>
      <Couleur_text>255;0;0;160</Couleur_text>
    </Classe>
  </Classes>
  <Terminales>
    <Terminale Caption="tabagisme">
      <Centre>103;60</Centre>
      <Font>Times New Roman;8;false;false;false;false</Font>
      <Type>Rectangle</Type>
      <Couleur_Interface>255;242;209;77</Couleur_Interface>
      <Couleur_Trace>255;255;0;128</Couleur_Trace>
      <Couleur_text>255;128;0;255</Couleur_text>
    </Terminale>
    <Terminale Caption="Amiante">
      <Centre>371;62</Centre>
      <Font>Times New Roman;8;false;false;false;false</Font>
      <Type>Rectangle</Type>
      <Couleur_Interface>255;242;209;77</Couleur_Interface>
      <Couleur_Trace>255;255;0;128</Couleur_Trace>
      <Couleur_text>255;128;0;255</Couleur_text>
    </Terminale>
    <Terminale Caption="Etape-evol">
      <Centre>105;371</Centre>
      <Font>Times New Roman;8;false;false;false;false</Font>
      <Type>Rectangle</Type>
      <Couleur_Interface>255;242;209;77</Couleur_Interface>
      <Couleur_Trace>255;255;0;128</Couleur_Trace>
      <Couleur_text>255;128;0;255</Couleur_text>
    </Terminale>
    <Terminale Caption="Signe-pul">
      <Centre>370;372</Centre>
      <Font>Times New Roman;8;false;false;false;false</Font>
      <Type>Rectangle</Type>
      <Couleur_Interface>255;242;209;77</Couleur_Interface>
      <Couleur_Trace>255;255;0;128</Couleur_Trace>
      <Couleur_text>255;128;0;255</Couleur_text>
    </Terminale>
    <Terminale Caption="maladie">
      <Centre>236;61</Centre>
      <Font>Times New Roman;8;false;false;false;false</Font>
      <Type>Rectangle</Type>
      <Couleur_Interface>255;250;189;5</Couleur_Interface>
      <Couleur_Trace>255;128;0;64</Couleur_Trace>
      <Couleur_text>255;128;0;255</Couleur_text>
    </Terminale>
  </Terminales>
  <Proprites>
    <Propriete m_namespace="S_Med" Caption="1 S_Med:A_Cause_de">
      <font>Times New Roman;8;false;false;false;false</font>
      <Couleur_Trace>255;0;0;255</Couleur_Trace>
      <Couleur_text>255;128;0;0</Couleur_text>
      <Type_Debut>Cycle</Type_Debut>
      <Debut>Cancer-poment S_Med:maladie</Debut>
      <Type_Fin>Rectangle</Type_Fin>
      <Fin>tabagisme</Fin>
    </Propriete>
    <Propriete m_namespace="S_Med" Caption=" S_Med:A_Cause_de">
      <font>Times New Roman;8;false;false;false;false</font>
      <Couleur_Trace>255;0;0;255</Couleur_Trace>
      <Couleur_text>255;128;0;0</Couleur_text>
      <Type_Debut>Cycle</Type_Debut>
      <Debut>Asbestose S_Med:maladie</Debut>
      <Type_Fin>Rectangle</Type_Fin>
      <Fin>Amiante</Fin>
    </Propriete>
    <Propriete m_namespace="S_Med" Caption=" S_Med:Peut_Evoluer">
```

```
      <font>Times New Roman;8;false;false;false;false</font>
      <Couleur_Trace>255;0;0;255</Couleur_Trace>
      <Couleur_text>255;128;0;0</Couleur_text>
      <Type_Debut>Cycle</Type_Debut>
      <Debut>Asbestose S_Med:maladie</Debut>
      <Type_Fin>Cycle</Type_Fin>
      <Fin>Cancer-poment S_Med:maladie</Fin>
    </Propriete>
    <Propriete m_namespace="S_Med" Caption=" S_Med:Se_Classe">
      <font>Times New Roman;8;false;false;false;false</font>
      <Couleur_Trace>255;0;0;255</Couleur_Trace>
      <Couleur_text>255;128;0;0</Couleur_text>
      <Type_Debut>Cycle</Type_Debut>
      <Debut>Cancer-poment S_Med:maladie</Debut>
      <Type_Fin>Cycle</Type_Fin>
      <Fin>Cancer3stde S_Med:maladie</Fin>
    </Propriete>
    <Propriete m_namespace="S_Med" Caption=" S_Med:se_Complique">
      <font>Times New Roman;8;false;false;false;false</font>
      <Couleur_Trace>255;0;0;255</Couleur_Trace>
      <Couleur_text>255;128;0;0</Couleur_text>
      <Type_Debut>Cycle</Type_Debut>
      <Debut>Cancer3stde S_Med:maladie</Debut>
      <Type_Fin>Cycle</Type_Fin>
      <Fin>Inssufisance-res S_Med:maladie</Fin>
    </Propriete>
    <Propriete m_namespace="S_Med" Caption="1 S_Med:se_Complique">
      <font>Times New Roman;8;false;false;false;false</font>
      <Couleur_Trace>255;0;0;255</Couleur_Trace>
      <Couleur_text>255;128;0;0</Couleur_text>
      <Type_Debut>Cycle</Type_Debut>
      <Debut>Asbestose S_Med:maladie</Debut>
      <Type_Fin>Cycle</Type_Fin>
      <Fin>Inssufisance-res S_Med:maladie</Fin>
    </Propriete>
    <Propriete m_namespace="S_Med" Caption=" S_Med:Type">
      <font>Times New Roman;8;false;false;false;false</font>
      <Couleur_Trace>255;0;0;255</Couleur_Trace>
      <Couleur_text>255;128;0;0</Couleur_text>
      <Type_Debut>Cycle</Type_Debut>
      <Debut>Cancer3stde S_Med:maladie</Debut>
      <Type_Fin>Rectangle</Type_Fin>
      <Fin>Etape-evol</Fin>
    </Propriete>
    <Propriete m_namespace="S_Med" Caption="1 S_Med:Type">
      <font>Times New Roman;8;false;false;false;false</font>
      <Couleur_Trace>255;0;0;255</Couleur_Trace>
      <Couleur_text>255;128;0;0</Couleur_text>
      <Type_Debut>Cycle</Type_Debut>
      <Debut>Inssufisance-res S_Med:maladie</Debut>
      <Type_Fin>Rectangle</Type_Fin>
      <Fin>Signe-pul</Fin>
    </Propriete>
    <Propriete m_namespace="S_Med" Caption="2 S_Med:Type">
      <font>Times New Roman;8;false;false;false;false</font>
      <Couleur_Trace>255;0;0;255</Couleur_Trace>
      <Couleur_text>255;128;0;64</Couleur_text>
      <Type_Debut>Cycle</Type_Debut>
      <Debut>Cancer-poment S_Med:maladie</Debut>
      <Type_Fin>Rectangle</Type_Fin>
      <Fin>maladie</Fin>
    </Propriete>
    <Propriete m_namespace="S_Med" Caption="3 S_Med:Type">
      <font>Times New Roman;8;false;false;false;false</font>
      <Couleur_Trace>255;0;0;255</Couleur_Trace>
      <Couleur_text>255;128;0;64</Couleur_text>
      <Type_Debut>Cycle</Type_Debut>
      <Debut>Asbestose S_Med:maladie</Debut>
```

```
        <Type_Fin>Rectangle</Type_Fin>
        <Fin>maladie</Fin>
      </Propriete>
    </Proprites>
  </RDF>
</Environement>
```

6.3.3. L'apprentissage des ontologies RDFS

Une foie l'utilisateur édite l'ontologie initiale et le graphe RDF qui présente un cas d'apprentissage, il peut lancer l'apprentissage de cette ontologie suivant des longueurs différentes en raffinant à chaque étape l'ontologie par une longueur supérieure à la longueur de l'étape précédente. L'interface de la hiérarchie résultante est la suivante :

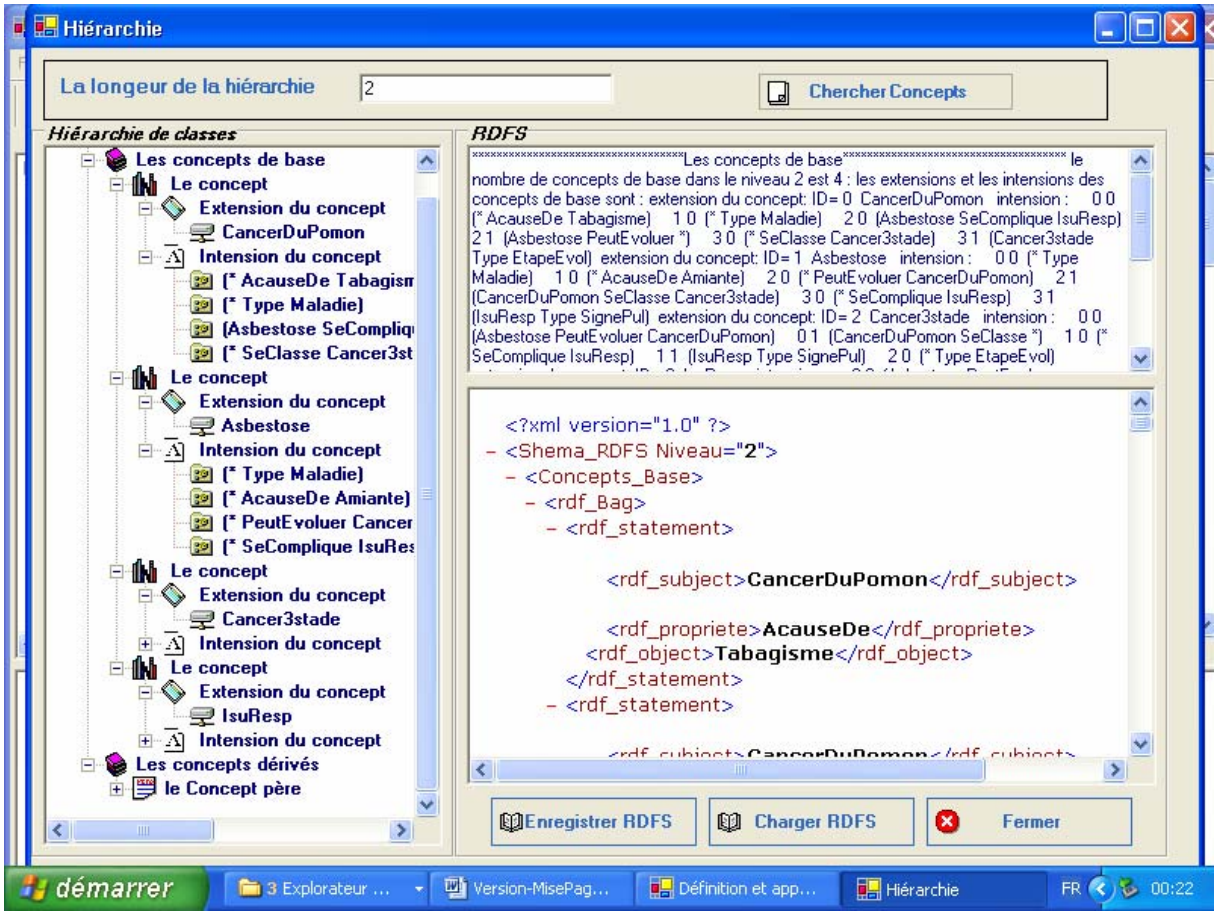
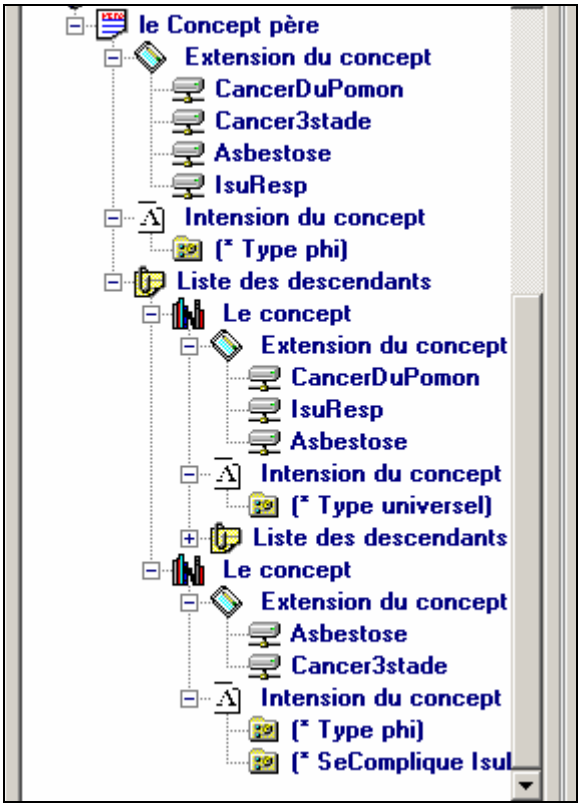


Figure 6. 13: L'interface de l'apprentissage.

L'utilisateur précise la longueur voulu de l'ontologie dans la case spécifique de la longueur de la hiérarchie situé au-dessus de l'interface, comme il est apparu dans cette interface la longueur est 2, puis il click sur le bouton *Chercher Concepts* pour voir quels sont les concepts de base, les concepts dérivés du longueur choisi et la hiérarchie entre ces concepts, il peut aussi enregistrer l'ontologie résultante dans un fichier .rdfs et le charger pour l'afficher.

6.3.3.1. La hiérarchie des concepts

Notre outil DLOT permet de construire la hiérarchie entre les nouveaux concepts obtenus par apprentissage, suivant la longueur sélectionnée. Pour chaque concept des concepts dérivés on trouve l'extension, l'intension et la liste des descendants de ce concept. On prend par exemple le concept père de l'ontologie résultante à partir de l'ontologie et le graphe RDF présenté au par avant :



L'extension d'un concept est présentée par un ensemble de ressources qui ont des certains ressemblances, ces ressemblances sont des triplets de la forme (sujet, propriété, objet) qui annotent l'ensemble des ressources existantes dans l'extension de ce concept. Si on prend l'exemple du concept père il a comme extension l'ensemble des ressources suivants : CancerDuPomon, Asbestose, Cancer3stade, Insuffisance respiratoire, et comme intension le triplet (*, type, phi). Ce concept a aussi deux descendants, comme il apparu dans l'interface de la hiérarchie.

6.3.3.2. L'enregistrement des ontologies RDFS résultats

Une foie on trouve la hiérarchie entre les concepts, les extensions des concepts et ses intensions, l'utilisateur peut enregistrer cette hiérarchie sous forme .rdfs, et il peut aussi le charger pour l'afficher directement. L'affichage de l'ontologie résultante de la longueur 2 de l'exemple précédent est comme suit :

```
<?xml version="1.0"?>
<Shema RDFS Niveau="2">
  <Concepts Base>
    <Class>
      <rdf:Bag Concept="0" rdf:li="statement0" rdf:li="statement1" rdf:li="statement2" rdf:li="statement3">
        <statement ID="Statement0">
          <rdf:Subject>CancerDuPomon</rdf:Subject>
          <rdf:predicate>AcauseDe</rdf:predicate>
          <rdf:object>Tabagisme</rdf:object>
        </statement>
        <statement ID="Statement1">
          <rdf:Subject>CancerDuPomon</rdf:Subject>
          <rdf:predicate>Type</rdf:predicate>
          <rdf:object>Maladie</rdf:object>
        </statement>
        <statement ID="Statement2">
          <rdf:subject>Asbestose</rdf:subject>
          <rdf:predicate>PeutEvoluer</rdf:predicate>
          <rdf:object>CancerDuPomon</rdf:object>
        </statement>
        <statement ID="Statement3">
          <rdf:Subject>CancerDuPomon</rdf:Subject>
          <rdf:predicate>SeClasse</rdf:predicate>
          <rdf:object>Cancer3stade</rdf:object>
        </statement>
      </rdf:Bag>
    </Class>
    <Class>
      <rdf:Bag Concept="1" rdf:li="statement0" rdf:li="statement1" rdf:li="statement2" rdf:li="statement3">
        <statement ID="Statement0">
          <rdf:Subject>Asbestose</rdf:Subject>
          <rdf:predicate>Type</rdf:predicate>
          <rdf:object>Maladie</rdf:object>
        </statement>
        <statement ID="Statement1">
          <rdf:Subject>Asbestose</rdf:Subject>
          <rdf:predicate>AcauseDe</rdf:predicate>
          <rdf:object>Amiante</rdf:object>
        </statement>
        <statement ID="Statement2">
          <rdf:Subject>Asbestose</rdf:Subject>
          <rdf:predicate>PeutEvoluer</rdf:predicate>
          <rdf:object>CancerDuPomon</rdf:object>
        </statement>
        <statement ID="Statement3">
          <rdf:Subject>Asbestose</rdf:Subject>
          <rdf:predicate>SeComplique</rdf:predicate>
          <rdf:object>IsuResp</rdf:object>
        </statement>
      </rdf:Bag>
    </Class>
    <Class>
      <rdf:Bag Concept="2" rdf:li="statement0" rdf:li="statement1" rdf:li="statement2">
        <statement ID="Statement0">
          <rdf:subject>CancerDuPomon</rdf:subject>
          <rdf:predicate>SeClasse</rdf:predicate>
          <rdf:object>Cancer3stade</rdf:object>
        </statement>
        <statement ID="Statement1">
          <rdf:Subject>Cancer3stade</rdf:Subject>
          <rdf:predicate>SeComplique</rdf:predicate>
          <rdf:object>IsuResp</rdf:object>
        </statement>
        <statement ID="Statement2">
          <rdf:Subject>Cancer3stade</rdf:Subject>
          <rdf:predicate>Type</rdf:predicate>
          <rdf:object>EtapeEvol</rdf:object>
        </statement>
      </rdf:Bag>
    </Class>
    <Class>
      <rdf:Bag Concept="3" rdf:li="statement0" rdf:li="statement1" rdf:li="statement2">
```

```
        <statement ID="Statement0">
          <rdf:subject>Asbestose</rdf:subject>
          <rdf:predicate>SeComplique</rdf:predicate>
          <rdf:object>IsuResp</rdf:object>
        </statement>
        <statement ID="Statement1">
          <rdf:subject>Cancer3stade</rdf:subject>
          <rdf:predicate>SeComplique</rdf:predicate>
          <rdf:object>IsuResp</rdf:object>
        </statement>
        <statement ID="Statement2">
          <rdf:Subject>IsuResp</rdf:Subject>
          <rdf:predicate>Type</rdf:predicate>
          <rdf:object>SignePul</rdf:object>
        </statement>
      </rdf:Bag>
    </Class>
  </Concepts Base>
  <Concepts Dérivés>
    <Class>
      <rdf:Bag Concept="2" rdf:li="statement0" rdf:li="statement1">
        <statement ID="Statement0">
          <Subject>
            <rdf:ALT>
              <rdf:li>Asbestose</rdf:li>
              <rdf:li>Cancer3stade</rdf:li>
            </rdf:ALT>
          </Subject>
          <rdf:predicate>Type</rdf:predicate>
          <rdf:object>phi</rdf:object>
        </statement>
        <statement ID="Statement1">
          <Subject>
            <rdf:ALT>
              <rdf:li>Asbestose</rdf:li>
              <rdf:li>Cancer3stade</rdf:li>
            </rdf:ALT>
          </Subject>
          <rdf:predicate>SeComplique</rdf:predicate>
          <rdf:object>IsuResp</rdf:object>
        </statement>
      </rdf:Bag>
    </Class>
    <Class>
      <rdf:Bag Concept="1" rdf:li="statement0" rdf:li="statement1">
        <statement ID="Statement0">
          <Subject>
            <rdf:ALT>
              <rdf:li>CancerDuPomon</rdf:li>
              <rdf:li>Asbestose</rdf:li>
            </rdf:ALT>
          </Subject>
          <rdf:predicate>AcauseDe</rdf:predicate>
          <rdf:object>phi</rdf:object>
        </statement>
        <statement ID="Statement1">
          <Subject>
            <rdf:ALT>
              <rdf:li>CancerDuPomon</rdf:li>
              <rdf:li>Asbestose</rdf:li>
            </rdf:ALT>
          </Subject>
          <rdf:predicate>Type</rdf:predicate>
          <rdf:object>Maladie</rdf:object>
        </statement>
      </rdf:Bag>
    </Class>
    <Class>
      <rdf:Bag Concept="1" rdf:li="statement0">
        <statement ID="Statement0">
          <Subject>
            <rdf:ALT>
```

```
            <rdf:li>CancerDuPomon</rdf:li>
            <rdf:li>Cancer3stade</rdf:li>
            <rdf:li>Asbestose</rdf:li>
            <rdf:li>IsuResp</rdf:li>
          </rdf:ALT>
        </Subject>
        <rdf:predicate>Type</rdf:predicate>
        <rdf:object>phi</rdf:object>
      </statement>
    </rdf:Bag>
  </Class>
  <Class>
    <rdf:Bag Concept="1" rdf:li="statement0">
      <statement ID="Statement0">
        <Subject>
          <rdf:ALT>
            <rdf:li>CancerDuPomon</rdf:li>
            <rdf:li>IsuResp</rdf:li>
            <rdf:li>Asbestose</rdf:li>
          </rdf:ALT>
        </Subject>
        <rdf:predicate>Type</rdf:predicate>
        <rdf:object>universel</rdf:object>
      </statement>
    </rdf:Bag>
  </Class>
 </Concepts Dérivés>
</Shema RDFS>
```

Chaque concepts est un bag : ensemble de Statement, un Statement est triplet (Subjectif, property, object). Dans le cas où le sujet (subject) ou l'objet (object) sont des ensembles ayant plus d'une ressource nous avons proposé de les représenter par un ALT entre les ressources de ce concept.

6.4. Conclusion

Nous avons présenté, au cours de ce chapitre, l'outil de définition et d'apprentissage des ontologies. Cet outil permet comme son nom indique, la définition des ontologies en définissant l'ensemble des classes et des propriétés que constitue l'ontologie, puis de définir le cas d'apprentissage qui est un graphe RDF.

Une fois l'utilisateur définit l'ontologie et le graphe RDF, il peut procéder à l'apprentissage suivant une longueur choisi, pour trouver les concepts de base, les concepts dérivés et la hiérarchie entre ces concepts. L'ontologie résultante peut être enregistrée sous forme .rdfs.

CHAPITRE 07
TESTS ET RESULTATS

7.1. Introduction

Dans ce chapitre nous verrons un test détaillé avec les résultats obtenus pour l'exemple que nous avons présenté dans la partie théorique en utilisant l'outil que nous avons développé DLOT. Dans ce test nous présentons les nouveaux concepts extraits : de base et dérivé pour chaque longueur et les hiérarchies obtenues suivant les longueurs choisissent.

7.2.Présentation des tests

7.2.1.Le premier test : application médecine

La médecine est un des domaines d'applications privilégiés du Web sémantique comme elle l'a été, à une autre époque, des techniques de l'Intelligence Artificielle, en particulier les systèmes experts. C'est en effet un domaine complexe où les informations à partager sont nombreuses. Ainsi, un des principaux mécanismes du Web sémantique qui est la description de ressources via des annotations est de la plus grande importance en bioinformatique. Le besoins en médecine est motivé par :

La recherche d'informations en médecine concerne de multiples utilisateurs du monde médical. Trouver rapidement sur le Web, avec le minimum de bruit et de silence possible, une information scientifique récente, a un intérêt non seulement pour le chercheur qui doit accéder à des bases hétérogènes et réitérer régulièrement les interrogations sur ces bases, pour les patients à la recherche d'informations, mais aussi dans la pratique médicale quotidienne où médecins et industriels pharmaceutiques sont amenés à rechercher de l'information. Une étude menée en 2000 aux Etats-Unis montrait que :

(1) plus d'utilisateurs utilisent le Web pour rechercher de l'information de santé que pour faire des achats, consulter des résultats sportifs ou des informations boursières,

(2) les gens recherchent huit fois plus souvent de l'information sur une maladie que de l'information en nutrition ou autre.

Les moteurs de recherche existants basés sur des mots-clé, peuvent retourner des documents non pertinents, à cause par exemple d'homonymie ou de mauvais contexte, ou rater des documents à cause de l'utilisation de mots différents (synonymie), de mots plus spécifiques, ou plus généraux (hypo-hyperonymie). La définition d'ontologies avec les langages formels du Web Sémantique devrait contribuer à une recherche plus « intelligente ».

Le problème ne se limite pas seulement au partage des données médicales. L'enjeu est aussi l'utilisation de ces données dans le cadre de l'aide à la décision. Ceci recouvre encore une fois de multiples aspects. Citons par exemple l'accès à des protocoles et guides de bonnes pratiques (par exemple prise en charge).

Les nouveaux langages, méthodes, outils développés pour un Web Sémantique, devraient contribuer de façon notable à mieux répondre à ces besoins en facilitant, l'indexation, la recherche et le partage d'informations médicales en vue d'une meilleure utilisation de ces données.

7.2.1.1.L'ontologie et le graphe initiaux

Dans l'ontologie initiale de médecine on représente les classes et les propriétés du domaine qui forment l'ontologie, on distingue la hiérarchie des classes et celle des propriétés : ces deux hiérarchies sont représentées graphiquement comme suit :

- Hiérarchie de classe

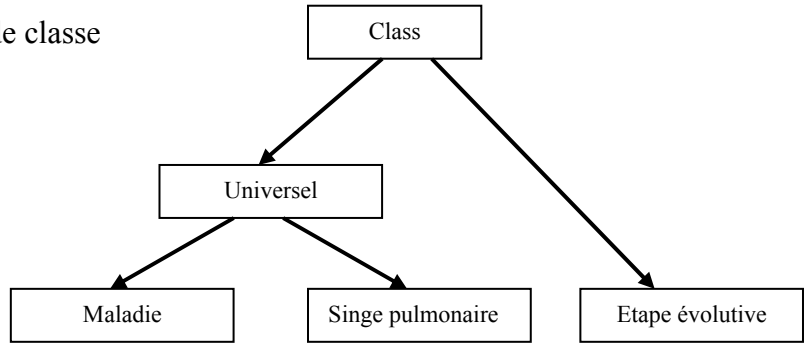


Figure 7. 1: Hiérarchie des concepts pour l'ontologie de médecine.

- Hiérarchie des propriétés

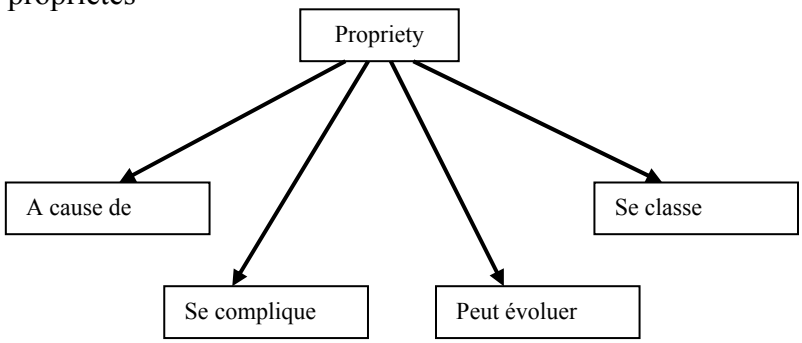


Figure 7. 2: Hiérarchie des propriétés pour l'ontologie de médecine.

Par notre outil DLOT un schema RDF est présenté dans un explorateur, qui contient l'ensemble des classes et des propriétés du domaine. Chaque classe et propriété est définit par plusieurs attributs est les valeur de ces attributs. Comme le montre la figure suivante:

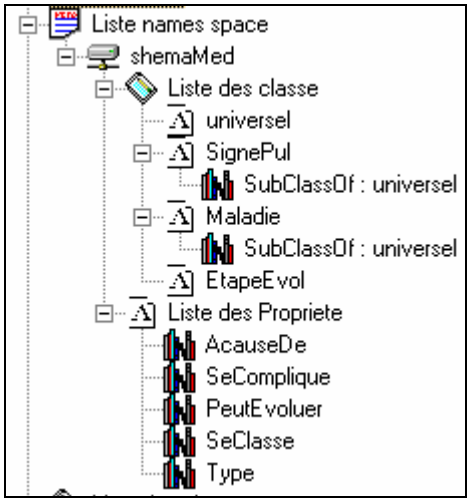


Figure 7. 3: L'interface de l'explorateur de l'ontologie initiale.

- Le cas d'apprentissage

Une foie l'ontologie est définit on peut éditer le graphe RDF en utilisant cette ontologie, la figure suivante montre le graphe RDF de l'exemple présenté dans la partie théorique en utilisant DLOT.

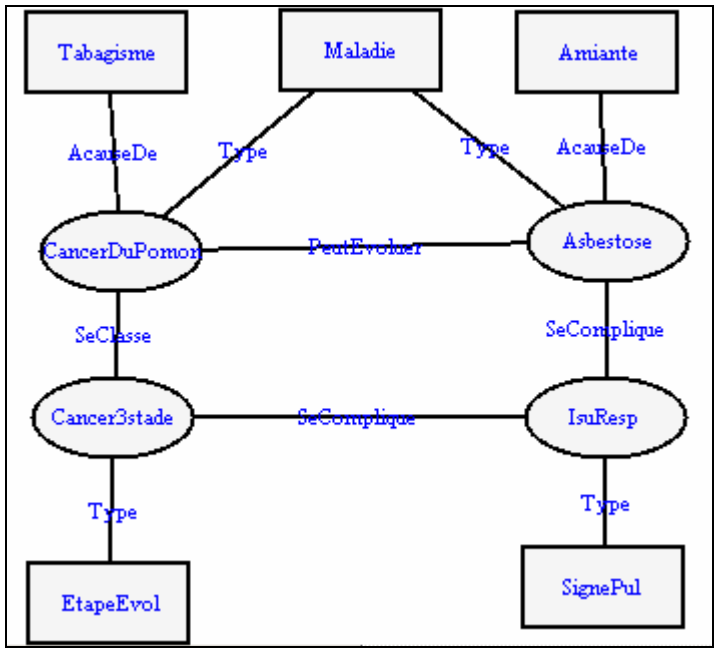


Figure 7. 4 : Le cas d'apprentissage : un graphe RDF de l'ontologie de médecine**.**

On peut enregistrer l'ontologie et le graphe RDF sous une forme textuelle, la figure suivante présente une partie de l'ensemble ontologie et graphe RDF :

```
<?xml version="1.0"?>

<Environnement m_nom="Nouveau Rdf" unite="20">
  <RDFS>
    <Schema m_nom="shemaMed">
      <Classes>
        <Classe ID="universel" />
        <Classe ID="SignePul">
          <SubClassOf>universel</SubClassOf>
        </Classe>
        <Classe ID="Maladie">
          <SubClassOf>universel</SubClassOf>
        </Classe>
        <Classe ID="EtapeEvol" />
      </Classes>
      <Proprietes>
        <Propriete ID="AcauseDe" />
        <Propriete ID="SeComplique" />
        <Propriete ID="PeutEvoluer" />
        <Propriete ID="SeClasse" />
        <Propriete ID="Type" />
      </Proprietes>
    </Schema>
  </RDFS>
  <RDF m_nom="graphe">
    <Classes>
      <Classe Caption="CancerDuPomon">
        <Centre>98;123</Centre>
        <Font>Times New Roman;8;false;false;false;false</Font>
        <Type>Cycle</Type>
        <Couleur_Interface>255;245;245;245</Couleur_Interface>
        <Couleur_Trace>255;0;0;0</Couleur_Trace>
        <Couleur_text>255;0;0;255</Couleur_text>
      </Classe>
      <Classe Caption="Asbestose">
        <Centre>341;118</Centre>
        <Font>Times New Roman;8;false;false;false;false</Font>
        <Type>Cycle</Type>
        <Couleur_Interface>255;245;245;245</Couleur_Interface>
        <Couleur_Trace>255;0;0;0</Couleur_Trace>
        <Couleur_text>255;0;0;255</Couleur_text>
      </Classe>
      <Classe Caption="Cancer3stade">
        <Centre>94;206</Centre>
        <Font>Times New Roman;8;false;false;false;false</Font>
        <Type>Cycle</Type>
        <Couleur_Interface>255;245;245;245</Couleur_Interface>
        <Couleur_Trace>255;0;0;0</Couleur_Trace>
        <Couleur_text>255;0;0;255</Couleur_text>
      </Classe>
     </Classes>
    <Terminales>
      <Terminale Caption="Tabagisme">
        <Centre>91;24</Centre>
        <Font>Times New Roman;8;false;false;false;false</Font>
        <Type>Rectangle</Type>
        <Couleur_Interface>255;245;245;245</Couleur_Interface>
        <Couleur_Trace>255;0;0;0</Couleur_Trace>
        <Couleur_text>255;0;0;255</Couleur_text>
      </Terminale>
      <Terminale Caption="Maladie">
```

```
            <Centre>218;23</Centre>
            <Font>Times New Roman;8;false;false;false;false</Font>
            <Type>Rectangle</Type>
            <Couleur_Interface>255;245;245;245</Couleur_Interface>
            <Couleur_Trace>255;0;0;0</Couleur_Trace>
            <Couleur_text>255;0;0;255</Couleur_text>
          </Terminale>
        </Terminales>
        <Proprites>
          <Propriete m_namespace="shemaMed" Caption="AcauseDe">
            <font>Times New Roman;8;false;false;false;false</font>
            <Couleur_Trace>255;0;0;0</Couleur_Trace>
            <Couleur_text>255;0;0;255</Couleur_text>
            <Type_Debut>Cycle</Type_Debut>
            <Debut>CancerDuPomon</Debut>
            <Type_Fin>Rectangle</Type_Fin>
            <Fin>Tabagisme</Fin>
          </Propriete>
          <Propriete m_namespace="shemaMed" Caption="SeClasse">
            <font>Times New Roman;8;false;false;false;false</font>
            <Couleur_Trace>255;0;0;0</Couleur_Trace>
            <Couleur_text>255;0;0;255</Couleur_text>
            <Type_Debut>Cycle</Type_Debut>
            <Debut>CancerDuPomon</Debut>
            <Type_Fin>Cycle</Type_Fin>
            <Fin>Cancer3stade</Fin>
          </Propriete>
          <Propriete m_namespace="shemaMed" Caption="SeComplique">
            <font>Times New Roman;8;false;false;false;false</font>
            <Couleur_Trace>255;0;0;0</Couleur_Trace>
            <Couleur_text>255;0;0;255</Couleur_text>
            <Type_Debut>Cycle</Type_Debut>
            <Debut>Asbestose</Debut>
            <Type_Fin>Cycle</Type_Fin>
            <Fin>IsuResp</Fin>
          </Propriete>
          <Propriete m_namespace="shemaMed" Caption="SeComplique">
            <font>Times New Roman;8;false;false;false;false</font>
            <Couleur_Trace>255;0;0;0</Couleur_Trace>
            <Couleur_text>255;0;0;255</Couleur_text>
            <Type_Debut>Cycle</Type_Debut>
            <Debut>Cancer3stade</Debut>
            <Type_Fin>Cycle</Type_Fin>
            <Fin>IsuResp</Fin>
          </Propriete>
          <Propriete m_namespace="shemaMed" Caption="Type">
            <font>Times New Roman;8;false;false;false;false</font>
            <Couleur_Trace>255;0;0;0</Couleur_Trace>
            <Couleur_text>255;0;0;255</Couleur_text>
            <Type_Debut>Cycle</Type_Debut>
            <Debut>IsuResp</Debut>
            <Type_Fin>Rectangle</Type_Fin>
            <Fin>SignePul</Fin>
          </Propriete>
        </Proprites>
      </RDF>
    </Enevirenement>
```

7.2.1.2. Résultats d'apprentissage

Commençant premièrement par la longueur 1 :

- Les concepts de base

**********Les concepts de base********** le nombre de concepts de base dans le niveau 1 est 4 : les extensions et les intensions des concepts de base sont : extension du concept: ID= 0 CancerDuPomon intension : 0 0 (* AcauseDe Tabagisme) 1 0 (* Type Maladie) 2 0 (Asbestose PeutEvoluer *) 3 0 (* SeClasse Cancer3stade) extension du concept: ID= 1 Asbestose intension : 0 0 (* Type Maladie) 1 0 (* AcauseDe Amiante) 2 0 (* PeutEvoluer CancerDuPomon) 3 0 (* SeComplique IsuResp) extension du concept: ID= 2 Cancer3stade intension : 0 0 (CancerDuPomon SeClasse *) 1 0 (* SeComplique IsuResp) 2 0 (* Type EtapeEvol) extension du concept: ID= 3 IsuResp intension : 0 0 (Asbestose SeComplique *) 1 0 (Cancer3stade SeComplique *) 2 0 (* Type SignePul) fin niveau 1

- Les concepts dérives

********** **********concepts derivé********** debut niveau 1 extension du Concept: Classe CancerDuPomon , Classe Asbestose , 0 0 , (* AcauseDe phi) 1 0 , (* Type Maladie) extension du Concept: Classe CancerDuPomon , Classe Cancer3stade , Classe Asbestose , Classe IsuResp , 0 0 , (* Type phi) extension du Concept: Classe CancerDuPomon , Classe IsuResp , Classe Asbestose , 0 0 , (* Type universel) extension du Concept: Classe Asbestose , Classe Cancer3stade , 0 0 , (* Type phi) 1 0 , (* SeComplique IsuResp) fin niveau 1

- La hiérarchie

Nous verrons dans la figure suivante les concepts de base : chaque concept est défini par extension qui est un ensemble de ressources et par intension qui est un ensemble de triplets décrivant les ressources de l'extension:

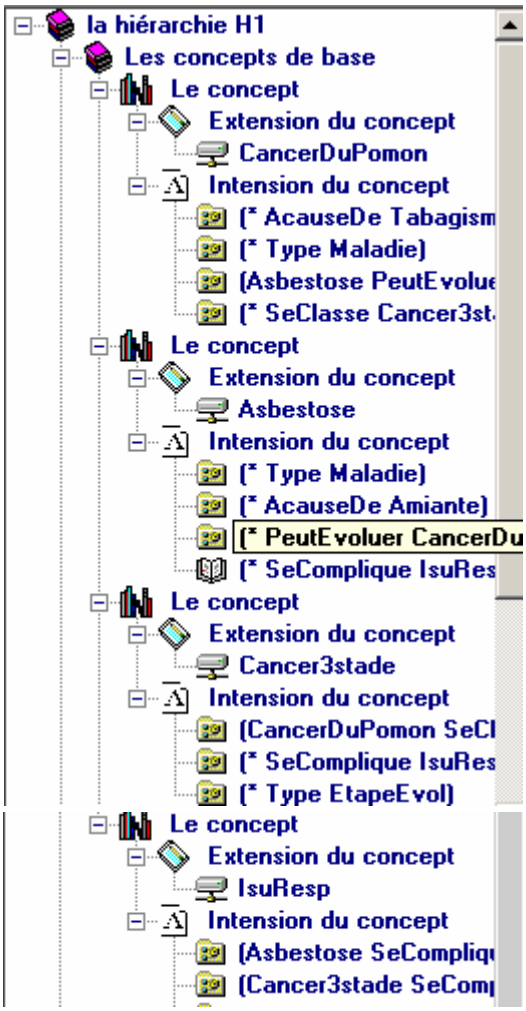


Figure 7. 5: La hiérarchie entre les concepts de base.

La figure suivante présente la hiérarchie entre les concepts dérivés :

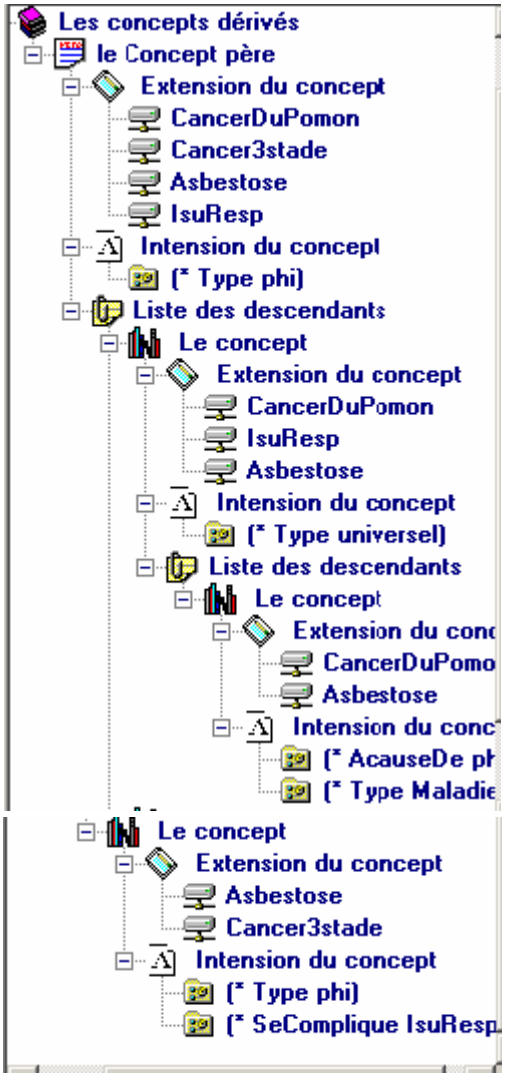


Figure 7. 6: La hiérarchie entre les concepts dérives.

La figure suivante présente l'ontologie RDFS résultante de longueur 1

```
<?xml version="1.0" ?>
- <Shema_RDFS Niveau="1">
- <Concepts_Base>
- <rdf_Bag>
- <rdf_statement>
  <rdf_subject>CancerDuPomon</rdf_subject>
  <rdf_propriete>AcauseDe</rdf_propriete>
  <rdf_object>Tabagisme</rdf_object>
    </rdf_statement>
- <rdf_statement>
  <rdf_subject>CancerDuPomon</rdf_subject>
  <rdf_propriete>Type</rdf_propriete>
  <rdf_object>Maladie</rdf_object>
    </rdf_statement>
- <rdf_statement>
  <rdf_subject>Asbestose</rdf_subject>
  <rdf_propriete>PeutEvoluer</rdf_propriete>
  <rdf_object>CancerDuPomon</rdf_object>
    </rdf_statement>
```

- <rdf_statement>
<rdf_subject>CancerDuPomon</rdf_subject>
<rdf_propriete>SeClasse</rdf_propriete>
<rdf_object>Cancer3stade</rdf_object>
</rdf_statement>
</rdf_Bag>
- <rdf_Bag>
- <rdf_statement>
<rdf_subject>Asbestose</rdf_subject>
<rdf_propriete>Type</rdf_propriete>
<rdf_object>Maladie</rdf_object>
</rdf_statement>
- <rdf_statement>
<rdf_subject>Asbestose</rdf_subject>
<rdf_propriete>AcauseDe</rdf_propriete>
<rdf_object>Amiante</rdf_object>
</rdf_statement>
- <rdf_statement>
<rdf_subject>Asbestose</rdf_subject>
<rdf_propriete>PeutEvoluer</rdf_propriete>
<rdf_object>CancerDuPomon</rdf_object>
</rdf_statement>
- <rdf_statement>
<rdf_subject>Asbestose</rdf_subject>
<rdf_propriete>SeComplique</rdf_propriete>
<rdf_object>IsuResp</rdf_object>
</rdf_statement>
</rdf_Bag>
- <rdf_Bag>
- <rdf_statement>
<rdf_subject>CancerDuPomon</rdf_subject>
<rdf_propriete>SeClasse</rdf_propriete>
<rdf_object>Cancer3stade</rdf_object>
</rdf_statement>
- <rdf_statement>
<rdf_subject>Cancer3stade</rdf_subject>
<rdf_propriete>SeComplique</rdf_propriete>
<rdf_object>IsuResp</rdf_object>
</rdf_statement>
- <rdf_statement>
<rdf_subject>Cancer3stade</rdf_subject>
<rdf_propriete>Type</rdf_propriete>
<rdf_object>EtapeEvol</rdf_object>
</rdf_statement>
</rdf_Bag>
- <rdf_Bag>
- <rdf_statement>
<rdf_subject>Asbestose</rdf_subject>
<rdf_propriete>SeComplique</rdf_propriete>
<rdf_object>IsuResp</rdf_object>
</rdf_statement>
- <rdf_statement>
<rdf_subject>Cancer3stade</rdf_subject>
<rdf_propriete>SeComplique</rdf_propriete>
<rdf_object>IsuResp</rdf_object>
</rdf_statement>
- <rdf_statement>
<rdf_subject>IsuResp</rdf_subject>
<rdf_propriete>Type</rdf_propriete>
<rdf_object>SignePul</rdf_object>
</rdf_statement>

```
    </rdf_Bag>
    </Concepts_Base>
- <Concepts_Dérivés>
- <rdf_Bag>
- <rdf_Bag>
- <rdf_statement>
- <rdf_subject>
- <rdf_Alt>
  <rdf_li>CancerDuPomon</rdf_li>
  <rdf_li>Asbestose</rdf_li>
    </rdf_Alt>
    </rdf_subject>
  <rdf_propriete>AcauseDe</rdf_propriete>
  <rdf_object>phi</rdf_object>
    </rdf_statement>
- <rdf_statement>
- <rdf_subject>
- <rdf_Alt>
  <rdf_li>CancerDuPomon</rdf_li>
  <rdf_li>Asbestose</rdf_li>
    </rdf_Alt>
    </rdf_subject>
  <rdf_propriete>Type</rdf_propriete>
  <rdf_object>Maladie</rdf_object>
    </rdf_statement>
    </rdf_Bag>
    </rdf_Bag>
- <rdf_Bag>
- <rdf_Bag>
- <rdf_statement>
- <rdf_subject>
- <rdf_Alt>
  <rdf_li>CancerDuPomon</rdf_li>
  <rdf_li>Cancer3stade</rdf_li>
  <rdf_li>Asbestose</rdf_li>
  <rdf_li>IsuResp</rdf_li>
    </rdf_Alt>
    </rdf_subject>
  <rdf_propriete>Type</rdf_propriete>
  <rdf_object>phi</rdf_object>
    </rdf_statement>
    </rdf_Bag>
    </rdf_Bag>
- <rdf_Bag>
- <rdf_Bag>
- <rdf_statement>
- <rdf_subject>
- <rdf_Alt>
  <rdf_li>CancerDuPomon</rdf_li>
  <rdf_li>IsuResp</rdf_li>
  <rdf_li>Asbestose</rdf_li>
    </rdf_Alt>
    </rdf_subject>
  <rdf_propriete>Type</rdf_propriete>
  <rdf_object>universel</rdf_object>
    </rdf_statement>
    </rdf_Bag>
    </rdf_Bag>
- <rdf_Bag>
- <rdf_Bag>
- <rdf_statement>
```

- <rdf_subject>
- <rdf_Alt>
<rdf_li>Asbestose</rdf_li>
<rdf_li>Cancer3stade</rdf_li>
</rdf_Alt>
</rdf_subject>
<rdf_propriete>Type</rdf_propriete>
<rdf_object>phi</rdf_object>
</rdf_statement>
- <rdf_Bag>
- <rdf_statement>
- <rdf_subject>
- <rdf_Alt>
<rdf_li>Asbestose</rdf_li>
<rdf_li>Cancer3stade</rdf_li>
</rdf_Alt>
</rdf_subject>
<rdf_propriete>SeComplique</rdf_propriete>
<rdf_object>IsuResp</rdf_object>
</rdf_statement>
- <rdf_statement>
<rdf_subject>IsuResp</rdf_subject>
<rdf_propriete>Type</rdf_propriete>
<rdf_object>SignePul</rdf_object>
</rdf_statement>
</rdf_Bag>
</rdf_Bag>
</rdf_Bag>
</Concepts_Dérivés>
</Shema_RDFS>

Passant maintenant au longueur 2 :

- Les concepts de base

**********Les concepts de base********** le nombre de concepts de base dans le niveau 2 est 4 : les extensions et les intensions des concepts de base sont : extension du concept: ID= 0 CancerDuPomon intension : 0 0 (* AcauseDe Tabagisme) 1 0 (* Type Maladie) 2 0 (Asbestose SeComplique IsuResp) 2 1 (Asbestose PeutEvoluer *) 3 0 (* SeClasse Cancer3stade) 3 1 (Cancer3stade Type EtapeEvol) 4 0 (* SeClasse Cancer3stade) 4 1 (Cancer3stade SeComplique IsuResp) 5 0 (Asbestose ACauseDe Amiante) 5 1 (Asbestose PeutEvoluer *) 6 0 (Asbestose Type Maladie) 6 1 (Asbestose PeutEvoluer *) extension du concept: ID= 1 Asbestose intension : 0 0 (* Type Maladie) 1 0 (* AcauseDe Amiante) 2 0 (* PeutEvoluer CancerDuPomon) 2 1 (CancerDuPomon SeClasse Cancer3stade) 3 0 (* SeComplique IsuResp) 3 1 (IsuResp Type SignePul) 4 0 (* PeutEvoluer CancerDuPomon) 4 1 (CancerDuPomon Type Maladie) 5 0 (* PeutEvoluer CancerDuPomon) 5 1 (CancerDuPomon AcauseDe Tabagisme) extension du concept: ID= 2 Cancer3stade intension : 0 0 (Asbestose PeutEvoluer CancerDuPomon) 0 1 (CancerDuPomon SeClasse *) 1 0 (* SeComplique IsuResp) 1 1 (IsuResp Type SignePul) 2 0 (* Type EtapeEvol) 3 0 (CancerDuPomon Type Maladie) 3 1 (CancerDuPomon SeClasse *) 4 0 (CancerDuPomon ACauseDe Tabagisme) 4 1 (CancerDuPomon SeClasse *) extension du concept: ID= 3 IsuResp intension : 0 0 (Asbestose PeutEvoluer CancerDuPomon) 0 1 (Asbestose SeComplique *) 1 0 (Cancer3stade Type EtapeEvol) 1 1 (Cancer3stade SeComplique *) 2 0 (* Type SignePul) 3 0 (Asbestose Type Maladie) 3 1 (Asbestose SeComplique *) 4 0 (Asbestose ACause Amiante) 4 1 (Asbestose SeComplique *)fin niveau 2
****

- Les concepts dérives

**********concepts derivé********** debut niveau 2 extension du Concept: Classe Asbestose , Classe Cancer3stade , 0 0 , (* Type phi) 1 0 , (* SeComplique IsuResp) 1 1 , (IsuResp Type SignePul) extension du Concept: Classe CancerDuPomon , Classe Asbestose , 0 0 , (* AcauseDe phi) 1 0 , (* Type Maladie) extension du Concept: Classe CancerDuPomon , Classe Cancer3stade , Classe Asbestose , Classe IsuResp , 0 0 , (* Type phi) extension du Concept: Classe CancerDuPomon , Classe IsuResp , Classe Asbestose , 0 0 , (* Type universel) fin niveau 2

- La hiérarchie

La hiérarchie entre les concepts de longueur 2 est la même que la hiérarchie de longueur 1, en effet les concepts sont les mêmes dans les extensions, la différence se trouve seulement dans les intensions des concepts.

- RDFS résultat

```
<?xml version="1.0" ?>
- <Shema_RDFS Niveau="2">
  - <Concepts_Base>
    - <rdf_Bag>
      - <rdf_statement>
          <rdf_subject>CancerDuPomon</rdf_subject>
          <rdf_propriete>AcauseDe</rdf_propriete>
          <rdf_object>Tabagisme</rdf_object>
      </rdf_statement>
      - <rdf_statement>
          <rdf_subject>CancerDuPomon</rdf_subject>
          <rdf_propriete>Type</rdf_propriete>
          <rdf_object>Maladie</rdf_object>
      </rdf_statement>
      - <rdf_Bag>
        - <rdf_statement>
            <rdf_subject>Asbestose</rdf_subject>
            <rdf_propriete>SeComplique</rdf_propriete>
            <rdf_object>IsuResp</rdf_object>
        </rdf_statement>
        - <rdf_statement>
            <rdf_subject>Asbestose</rdf_subject>
            <rdf_propriete>PeutEvoluer</rdf_propriete>
            <rdf_object>CancerDuPomon</rdf_object>
        </rdf_statement>
      </rdf_Bag>
      - <rdf_Bag>
        - <rdf_statement>
            <rdf_subject>CancerDuPomon</rdf_subject>
            <rdf_propriete>SeClasse</rdf_propriete>
            <rdf_object>Cancer3stade</rdf_object>
        </rdf_statement>
        - <rdf_statement>
            <rdf_subject>Cancer3stade</rdf_subject>
            <rdf_propriete>Type</rdf_propriete>
            <rdf_object>EtapeEvol</rdf_object>
        </rdf_statement>
      </rdf_Bag>
    </rdf_Bag>
    - <rdf_Bag>
      - <rdf_statement>
          <rdf_subject>Asbestose</rdf_subject>
          <rdf_propriete>Type</rdf_propriete>
          <rdf_object>Maladie</rdf_object>
      </rdf_statement>
      - <rdf_statement>
          <rdf_subject>Asbestose</rdf_subject>
          <rdf_propriete>AcauseDe</rdf_propriete>
          <rdf_object>Amiante</rdf_object>
      </rdf_statement>
      - <rdf_Bag>
        - <rdf_statement>
            <rdf_subject>Asbestose</rdf_subject>
            <rdf_propriete>PeutEvoluer</rdf_propriete>
            <rdf_object>CancerDuPomon</rdf_object>
        </rdf_statement>
        - <rdf_statement>
```

```
            <rdf_subject>CancerDuPomon</rdf_subject>
            <rdf_propriete>SeClasse</rdf_propriete>
            <rdf_object>Cancer3stade</rdf_object>
        </rdf_statement>
    </rdf_Bag>
    - <rdf_Bag>
        - <rdf_statement>
            <rdf_subject>Asbestose</rdf_subject>
            <rdf_propriete>SeComplique</rdf_propriete>
            <rdf_object>IsuResp</rdf_object>
        </rdf_statement>
        - <rdf_statement>
            <rdf_subject>IsuResp</rdf_subject>
            <rdf_propriete>Type</rdf_propriete>
            <rdf_object>SignePul</rdf_object>
        </rdf_statement>
    </rdf_Bag>
</rdf_Bag>
- <rdf_Bag>
    - <rdf_Bag>
        - <rdf_statement>
            <rdf_subject>Asbestose</rdf_subject>
            <rdf_propriete>PeutEvoluer</rdf_propriete>
            <rdf_object>CancerDuPomon</rdf_object>
        </rdf_statement>
        - <rdf_statement>
            <rdf_subject>CancerDuPomon</rdf_subject>
            <rdf_propriete>SeClasse</rdf_propriete>
            <rdf_object>Cancer3stade</rdf_object>
        </rdf_statement>
    </rdf_Bag>
    - <rdf_Bag>
        - <rdf_statement>
            <rdf_subject>Cancer3stade</rdf_subject>
            <rdf_propriete>SeComplique</rdf_propriete>
            <rdf_object>IsuResp</rdf_object>
        </rdf_statement>
        - <rdf_statement>
            <rdf_subject>IsuResp</rdf_subject>
            <rdf_propriete>Type</rdf_propriete>
            <rdf_object>SignePul</rdf_object>
        </rdf_statement>
    </rdf_Bag>
    - <rdf_statement>
        <rdf_subject>Cancer3stade</rdf_subject>
        <rdf_propriete>Type</rdf_propriete>
        <rdf_object>EtapeEvol</rdf_object>
    </rdf_statement>
</rdf_Bag>
- <rdf_Bag>
    - <rdf_Bag>
        - <rdf_statement>
            <rdf_subject>Asbestose</rdf_subject>
            <rdf_propriete>PeutEvoluer</rdf_propriete>
            <rdf_object>CancerDuPomon</rdf_object>
        </rdf_statement>
        - <rdf_statement>
            <rdf_subject>Asbestose</rdf_subject>
            <rdf_propriete>SeComplique</rdf_propriete>
            <rdf_object>IsuResp</rdf_object>
        </rdf_statement>
    </rdf_Bag>
    - <rdf_Bag>
        - <rdf_statement>
            <rdf_subject>Cancer3stade</rdf_subject>
            <rdf_propriete>Type</rdf_propriete>
            <rdf_object>EtapeEvol</rdf_object>
        </rdf_statement>
```

```
            - <rdf_statement>
                <rdf_subject>Cancer3stade</rdf_subject>
                <rdf_propriete>SeComplique</rdf_propriete>
                <rdf_object>IsuResp</rdf_object>
              </rdf_statement>
          </rdf_Bag>
            - <rdf_statement>
                <rdf_subject>IsuResp</rdf_subject>
                <rdf_propriete>Type</rdf_propriete>
                <rdf_object>SignePul</rdf_object>
          </rdf_statement>
      </rdf_Bag>
  </Concepts_Base>
    - <Concepts_Dérivés>
        - <rdf_Bag>
            - <rdf_Bag>
                - <rdf_statement>
                    - <rdf_subject>
                        - <rdf_Alt>
                            <rdf_li>Asbestose</rdf_li>
                            <rdf_li>Cancer3stade</rdf_li>
                        </rdf_Alt>
                    </rdf_subject>
                        <rdf_propriete>Type</rdf_propriete>
                        <rdf_object>phi</rdf_object>
                </rdf_statement>
                - <rdf_Bag>
                    - <rdf_statement>
                        - <rdf_subject>
                            - <rdf_Alt>
                                <rdf_li>Asbestose</rdf_li>
                                <rdf_li>Cancer3stade</rdf_li>
                            </rdf_Alt>
                        </rdf_subject>
                            <rdf_propriete>SeComplique</rdf_propriete>
                            <rdf_object>IsuResp</rdf_object>
                    </rdf_statement>
                    - <rdf_statement>
                            <rdf_subject>IsuResp</rdf_subject>
                            <rdf_propriete>Type</rdf_propriete>
                            <rdf_object>SignePul</rdf_object>
                    </rdf_statement>
                </rdf_Bag>
            </rdf_Bag>
        </rdf_Bag>
        - <rdf_Bag>
            - <rdf_Bag>
                - <rdf_statement>
                    - <rdf_subject>
                        - <rdf_Alt>
                            <rdf_li>CancerDuPomon</rdf_li>
                            <rdf_li>Asbestose</rdf_li>
                        </rdf_Alt>
                    </rdf_subject>
                        <rdf_propriete>AcauseDe</rdf_propriete>
                        <rdf_object>phi</rdf_object>
                </rdf_statement>
                - <rdf_statement>
                    - <rdf_subject>
                        - <rdf_Alt>
                            <rdf_li>CancerDuPomon</rdf_li>
                            <rdf_li>Asbestose</rdf_li>
                        </rdf_Alt>
                    </rdf_subject>
                        <rdf_propriete>Type</rdf_propriete>
                        <rdf_object>Maladie</rdf_object>
                </rdf_statement>
            </rdf_Bag>
```

```
            </rdf_Bag>
              - <rdf_Bag>
                 - <rdf_Bag>
                    - <rdf_statement>
                       - <rdf_subject>
                          - <rdf_Alt>
                                <rdf_li>CancerDuPomon</rdf_li>
                                <rdf_li>Cancer3stade</rdf_li>
                                <rdf_li>Asbestose</rdf_li>
                                <rdf_li>IsuResp</rdf_li>
                            </rdf_Alt>
                       </rdf_subject>
                    <rdf_propriete>Type</rdf_propriete>
                    <rdf_object>phi</rdf_object>
                 </rdf_statement>
              </rdf_Bag>
        </rdf_Bag>
- <rdf_Bag>
- <rdf_Bag>
  - <rdf_statement>
     - <rdf_subject>
        - <rdf_Alt>
           <rdf_li>CancerDuPomon</rdf_li>
           <rdf_li>IsuResp</rdf_li>
           <rdf_li>Asbestose</rdf_li>
         </rdf_Alt>
      </rdf_subject>
      <rdf_propriete>Type</rdf_propriete>
     <rdf_object>universel</rdf_object>
  </rdf_statement>
  </rdf_Bag>
  </rdf_Bag>
  </Concepts_Dérivés>
  </Shema_RDFS
```

Nous présentons par la suite les résultats pour la longueur 3 :

- Les concepts de base

```
**********Les concepts de base********** le nombre de concepts de base dans le niveau 3 est 4 : les extensions et les intensions des concepts de base sont : extension du concept: ID= 0 CancerDuPomon intension : 0 0 (* AcauseDe Tabagisme) 1 0 (* Type Maladie) 2 0 (Asbestose PeutEvoluer CancerDuPomon) 2 1 (Asbestose SeComplique IsuResp) 2 2 (Asbestose PeutEvoluer *) 3 0 (* SeClasse Cancer3stade) 3 1 (Cancer3stade Type EtapeEvol) extension du concept: ID= 1 Asbestose intension : 0 0 (* Type Maladie) 1 0 (* AcauseDe Amiante) 2 0 (* PeutEvoluer CancerDuPomon) 2 1 (CancerDuPomon SeClasse Cancer3stade) 2 2 (Cancer3stade Type EtapeEvol) 3 0 (* SeComplique IsuResp) 3 1 (IsuResp Type SignePul) extension du concept: ID= 2 Cancer3stade intension : 0 0 (Asbestose SeComplique IsuResp) 0 1 (Asbestose PeutEvoluer CancerDuPomon) 0 2 (CancerDuPomon SeClasse *) 1 0 (* SeComplique IsuResp) 1 1 (IsuResp Type SignePul) 2 0 (* Type EtapeEvol) extension du concept: ID= 3 IsuResp intension : 0 0 (Asbestose SeComplique IsuResp) 0 1 (Asbestose PeutEvoluer CancerDuPomon) 0 2 (Asbestose SeComplique *) 1 0 (Cancer3stade SeComplique IsuResp) 1 1 (Cancer3stade Type EtapeEvol) 1 2 (Cancer3stade SeComplique *) 2 0 (* Type SignePul) fin niveau 3 **********
```

- Les concepts dérives

```
**********concepts derivé********** debut niveau 3 extension du Concept: Classe Asbestose , Classe Cancer3stade , 0 0 , (* Type phi) 1 0 , (* SeComplique IsuResp) 1 1 , (IsuResp Type SignePul) extension du Concept: Classe CancerDuPomon , Classe Asbestose , 0 0 , (* AcauseDe phi) 1 0 , (* Type Maladie) extension du Concept: Classe CancerDuPomon , Classe Cancer3stade , Classe Asbestose , Classe IsuResp , 0 0 , (* Type phi) extension du Concept: Classe CancerDuPomon , Classe IsuResp , Classe Asbestose , 0 0 , (* Type universel) fin niveau 3
```

- RDFS résultat

```
<?xml version="1.0" ?>
- <Shema_RDFS Niveau="3">
+ <Concepts_Base>
```

- <Concepts_Dérivés>
- <rdf_Bag>
- <rdf_Bag>
- <rdf_statement>
- <rdf_subject>
- <rdf_Alt>
<rdf_li>Asbestose</rdf_li>
<rdf_li>Cancer3stade</rdf_li>
</rdf_Alt>
</rdf_subject>
<rdf_propriete>Type</rdf_propriete>
<rdf_object>phi</rdf_object>
</rdf_statement>
- <rdf_Bag>
- <rdf_statement>
- <rdf_subject>
- <rdf_Alt>
<rdf_li>Asbestose</rdf_li>
<rdf_li>Cancer3stade</rdf_li>
</rdf_Alt>
</rdf_subject>
<rdf_propriete>SeComplique</rdf_propriete>
<rdf_object>IsuResp</rdf_object>
</rdf_statement>
- <rdf_statement>
<rdf_subject>IsuResp</rdf_subject>
<rdf_propriete>Type</rdf_propriete>
<rdf_object>SignePul</rdf_object>
</rdf_statement>
</rdf_Bag>
</rdf_Bag>
</rdf_Bag>
- <rdf_Bag>
- <rdf_Bag>
- <rdf_statement>
- <rdf_subject>
- <rdf_Alt>
<rdf_li>CancerDuPomon</rdf_li>
<rdf_li>Asbestose</rdf_li>
</rdf_Alt>
</rdf_subject>
<rdf_propriete>AcauseDe</rdf_propriete>
<rdf_object>phi</rdf_object>
</rdf_statement>
- <rdf_statement>
- <rdf_subject>
- <rdf_Alt>
<rdf_li>CancerDuPomon</rdf_li>
<rdf_li>Asbestose</rdf_li>
</rdf_Alt>
</rdf_subject>
<rdf_propriete>Type</rdf_propriete>
<rdf_object>Maladie</rdf_object>
</rdf_statement>
</rdf_Bag>
</rdf_Bag>
- <rdf_Bag>
- <rdf_Bag>
- <rdf_statement>
- <rdf_subject>
- <rdf_Alt>
<rdf_li>CancerDuPomon</rdf_li>
<rdf_li>Cancer3stade</rdf_li>
<rdf_li>Asbestose</rdf_li>
<rdf_li>IsuResp</rdf_li>
</rdf_Alt>
</rdf_subject>
<rdf_propriete>Type</rdf_propriete>

<rdf_object>**phi**</rdf_object>
</rdf_statement>
</rdf_Bag>
</rdf_Bag>
- <rdf_Bag>
- <rdf_Bag>
- <rdf_statement>
- <rdf_subject>
- <rdf_Alt>
<rdf_li>**CancerDuPomon**</rdf_li>
<rdf_li>**IsuResp**</rdf_li>
<rdf_li>**Asbestose**</rdf_li>
</rdf_Alt>
</rdf_subject>
<rdf_propriete>**Type**</rdf_propriete>
<rdf_object>**universel**</rdf_object>
</rdf_statement>
</rdf_Bag>
</rdf_Bag>
</Concepts_Dérivés>
</Shema_RDFS>

Nous verrons par la suite la longueur 4

- Les concepts de base

**********Les concepts de base********** le nombre de concepts de base dans le niveau 4 est 4 : les extensions et les intensions des concepts de base sont : extension du concept: ID= 0 CancerDuPomon intension : 0 0 (* AcauseDe Tabagisme) 1 0 (* Type Maladie) 2 0 (Asbestose SeComplique IsuResp) 2 1 (Asbestose PeutEvoluer CancerDuPomon) 2 2 (Asbestose PeutEvoluer *) 3 0 (* SeClasse Cancer3stade) 3 1 (Cancer3stade Type EtapeEvol) extension du concept: ID= 1 Asbestose intension : 0 0 (* Type Maladie) 1 0 (* AcauseDe Amiante) 2 0 (* PeutEvoluer CancerDuPomon) 2 1 (CancerDuPomon SeClasse Cancer3stade) 2 2 (Cancer3stade Type EtapeEvol) 3 0 (* SeComplique IsuResp) 3 1 (IsuResp Type SignePul) extension du concept: ID= 2 Cancer3stade intension : 0 0 (Asbestose PeutEvoluer CancerDuPomon) 0 1 (Asbestose SeComplique IsuResp) 0 2 (CancerDuPomon SeClasse *) 1 0 (* SeComplique IsuResp) 1 1 (IsuResp Type SignePul) 2 0 (* Type EtapeEvol) extension du concept: ID= 3 IsuResp intension : 0 0 (Asbestose PeutEvoluer CancerDuPomon) 0 1 (Asbestose SeComplique IsuResp) 0 2 (Asbestose SeComplique *) 1 0 (Cancer3stade Type EtapeEvol) 1 1 (Cancer3stade SeComplique IsuResp) 1 2 (Cancer3stade SeComplique *) 2 0 (* Type SignePul) fin niveau 4 **********

- Les concepts dérives

**********concepts derivé********** debut niveau 4 extension du Concept: Classe Asbestose , Classe Cancer3stade , 0 0 , (* Type phi) 1 0 , (* SeComplique IsuResp) 1 1 , (IsuResp Type SignePul) extension du Concept: Classe CancerDuPomon , Classe Asbestose , 0 0 , (* AcauseDe phi) 1 0 , (* Type Maladie) extension du Concept: Classe CancerDuPomon , Classe Cancer3stade , Classe Asbestose , Classe IsuResp , 0 0 , (* Type phi) extension du Concept: Classe CancerDuPomon , Classe IsuResp , Classe Asbestose , 0 0 , (* Type universel) fin niveau 4

- RDFS résultat

<?xml version="1.0" ?>
- <Shema_RDFS Niveau="**4**">
- <Concepts_Base>
- <rdf_Bag>
- <rdf_statement>
<rdf_subject>**CancerDuPomon**</rdf_subject>
<rdf_propriete>**AcauseDe**</rdf_propriete>
<rdf_object>**Tabagisme**</rdf_object>
</rdf_statement>
- <rdf_statement>
<rdf_subject>**CancerDuPomon**</rdf_subject>
<rdf_propriete>**Type**</rdf_propriete>
<rdf_object>**Maladie**</rdf_object>
</rdf_statement>
- <rdf_Bag>
- <rdf_statement>
<rdf_subject>**Asbestose**</rdf_subject>

<rdf_propriete>SeComplique</rdf_propriete>
<rdf_object>IsuResp</rdf_object>
</rdf_statement>
- <rdf_statement>
<rdf_subject>Asbestose</rdf_subject>
<rdf_propriete>PeutEvoluer</rdf_propriete>
<rdf_object>CancerDuPomon</rdf_object>
</rdf_statement>
- <rdf_statement>
<rdf_subject>Asbestose</rdf_subject>
<rdf_propriete>PeutEvoluer</rdf_propriete>
<rdf_object>CancerDuPomon</rdf_object>
</rdf_statement>
</rdf_Bag>
- <rdf_Bag>
- <rdf_statement>
<rdf_subject>CancerDuPomon</rdf_subject>
<rdf_propriete>SeClasse</rdf_propriete>
<rdf_object>Cancer3stade</rdf_object>
</rdf_statement>
- <rdf_statement>
<rdf_subject>Cancer3stade</rdf_subject>
<rdf_propriete>Type</rdf_propriete>
<rdf_object>EtapeEvol</rdf_object>
</rdf_statement>
</rdf_Bag>
</rdf_Bag>
- <rdf_Bag>
- <rdf_statement>
<rdf_subject>Asbestose</rdf_subject>
<rdf_propriete>Type</rdf_propriete>
<rdf_object>Maladie</rdf_object>
</rdf_statement>
- <rdf_statement>
<rdf_subject>Asbestose</rdf_subject>
<rdf_propriete>AcauseDe</rdf_propriete>
<rdf_object>Amiante</rdf_object>
</rdf_statement>
- <rdf_Bag>
- <rdf_statement>
<rdf_subject>Asbestose</rdf_subject>
<rdf_propriete>PeutEvoluer</rdf_propriete>
<rdf_object>CancerDuPomon</rdf_object>
</rdf_statement>
- <rdf_statement>
<rdf_subject>CancerDuPomon</rdf_subject>
<rdf_propriete>SeClasse</rdf_propriete>
<rdf_object>Cancer3stade</rdf_object>
</rdf_statement>
- <rdf_statement>
<rdf_subject>Cancer3stade</rdf_subject>
<rdf_propriete>Type</rdf_propriete>
<rdf_object>EtapeEvol</rdf_object>
</rdf_statement>
</rdf_Bag>
- <rdf_Bag>
- <rdf_statement>
<rdf_subject>Asbestose</rdf_subject>
<rdf_propriete>SeComplique</rdf_propriete>
<rdf_object>IsuResp</rdf_object>
</rdf_statement>
- <rdf_statement>
<rdf_subject>IsuResp</rdf_subject>
<rdf_propriete>Type</rdf_propriete>
<rdf_object>SignePul</rdf_object>
</rdf_statement>
</rdf_Bag>
</rdf_Bag>

- <rdf_Bag>
- <rdf_Bag>
- <rdf_statement>
<rdf_subject>Asbestose</rdf_subject>
<rdf_propriete>PeutEvoluer</rdf_propriete>
<rdf_object>CancerDuPomon</rdf_object>
</rdf_statement>
- <rdf_statement>
<rdf_subject>Asbestose</rdf_subject>
<rdf_propriete>SeComplique</rdf_propriete>
<rdf_object>IsuResp</rdf_object>
</rdf_statement>
- <rdf_statement>
<rdf_subject>CancerDuPomon</rdf_subject>
<rdf_propriete>SeClasse</rdf_propriete>
<rdf_object>Cancer3stade</rdf_object>
</rdf_statement>
</rdf_Bag>
- <rdf_Bag>
- <rdf_statement>
<rdf_subject>Cancer3stade</rdf_subject>
<rdf_propriete>SeComplique</rdf_propriete>
<rdf_object>IsuResp</rdf_object>
</rdf_statement>
- <rdf_statement>
<rdf_subject>IsuResp</rdf_subject>
<rdf_propriete>Type</rdf_propriete>
<rdf_object>SignePul</rdf_object>
</rdf_statement>
</rdf_Bag>
- <rdf_statement>
<rdf_subject>Cancer3stade</rdf_subject>
<rdf_propriete>Type</rdf_propriete>
<rdf_object>EtapeEvol</rdf_object>
</rdf_statement>
</rdf_Bag>
- <rdf_Bag>
- <rdf_Bag>
- <rdf_statement>
<rdf_subject>Asbestose</rdf_subject>
<rdf_propriete>PeutEvoluer</rdf_propriete>
<rdf_object>CancerDuPomon</rdf_object>
</rdf_statement>
- <rdf_statement>
<rdf_subject>Asbestose</rdf_subject>
<rdf_propriete>SeComplique</rdf_propriete>
<rdf_object>IsuResp</rdf_object>
</rdf_statement>
- <rdf_statement>
<rdf_subject>Asbestose</rdf_subject>
<rdf_propriete>SeComplique</rdf_propriete>
<rdf_object>IsuResp</rdf_object>
</rdf_statement>
</rdf_Bag>
- <rdf_Bag>
- <rdf_statement>
<rdf_subject>Cancer3stade</rdf_subject>
<rdf_propriete>Type</rdf_propriete>
<rdf_object>EtapeEvol</rdf_object>
</rdf_statement>
- <rdf_statement>
<rdf_subject>Cancer3stade</rdf_subject>
<rdf_propriete>SeComplique</rdf_propriete>
<rdf_object>IsuResp</rdf_object>
</rdf_statement>
- <rdf_statement>
<rdf_subject>Cancer3stade</rdf_subject>
<rdf_propriete>SeComplique</rdf_propriete>

```
  <rdf_object>IsuResp</rdf_object>
     </rdf_statement>
     </rdf_Bag>
- <rdf_statement>
  <rdf_subject>IsuResp</rdf_subject>
  <rdf_propriete>Type</rdf_propriete>
  <rdf_object>SignePul</rdf_object>
     </rdf_statement>
     </rdf_Bag>
     </Concepts_Base>
+ <Concepts_Dérivés>
     </Shema_RDFS>
```

7.2.2.Le deuxième test : l'ontologie de la géométrie projective

L'ontologie de la géométrie projective construite par D. HILBERT exposée dans l'ouvrage *« Les fondements de la géométrie »* [66] publié initialement en 1899. D. HILBERT fonde la géométrie sur 5 groupes d'axiomes regroupés selon la notion fondamentale qu'ils décrivent : les axiomes d'appartenance, les axiomes d'ordre, les axiomes de congruence, les axiomes des parallèles et les axiomes de continuité.

7.2.2.1. Le niveau terminologique

D. HILBERT précise les objets de l'ontologie de la géométrie projective: les points, les droites et les plans. Les relations entre ces objets sont ensuite formalisées par les axiomes. Trois concepts doivent donc déjà être inclus dans notre ontologie : *Point*, *Droite* et *Plan*. La hiérarchie qui intègre ces concepts a pour racine *Objet Géométrique*, regroupant l'ensemble des concepts du domaine. Le concept *Objet Géométrique* a pour fils *Point* et *Ensemble de Points*. *Ensemble de Points* a pour fils *Droite* et *Plan*. Regrouper des concepts dans un concept *Ensemble de Points* va permettre de définir des relations génériques de nature ensembliste. La hiérarchie présentée par la suite est une partie extraite à partir de la hiérarchie de l'ontologie de la géométrie projective.

- La hiérarchie des concepts

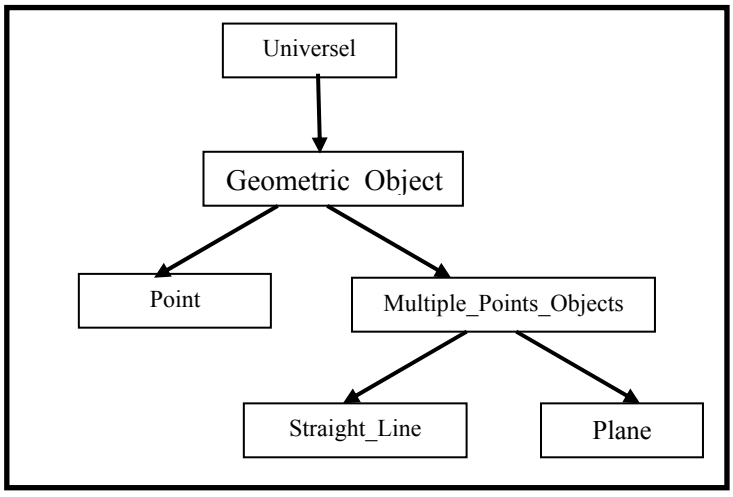


Figure 7. 7: La hiérarchie des concepts de l'ontologie de la géométrie projective.

- La hiérarchie des relations

La hiérarchie des relations est présentée dans la figure suivante :

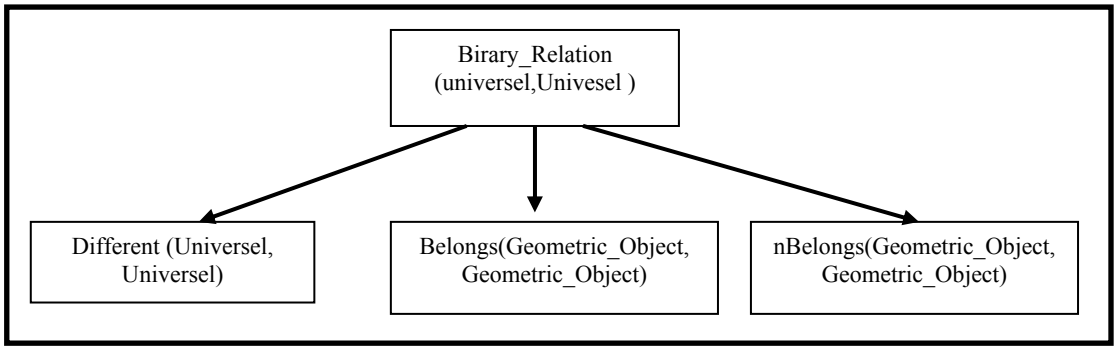


Figure 7. 8: La hiérarchie des propriétés de l'ontologie de la géométrie projective.

- Le cas d'apprentissage

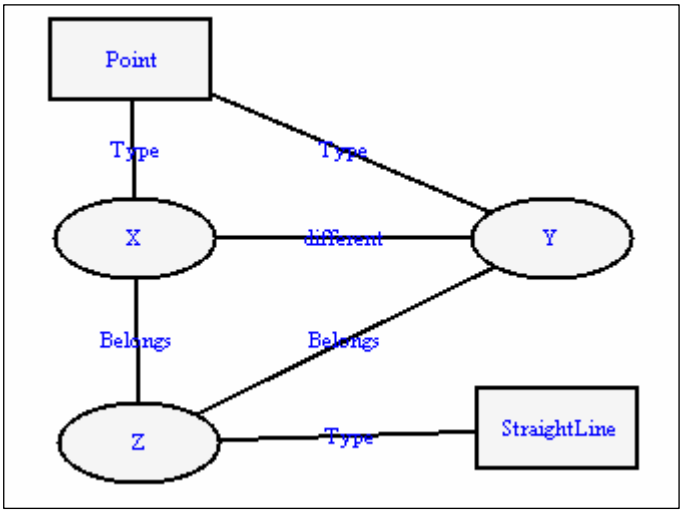


Figure 7. 9: Le cas d'apprentissage pour l'ontologie de la géométrie projective.

« Il existe une droite Z liée à deux points donnés X et Y à laquelle appartiennent ces deux points ». Ce qui est interprété par : étant donnés deux points X et Y, il existe une droite Z passant par ces deux points.

7.2.2.2.Résultats d'apprentissage

- La longueur 1 : les concepts de base

**********Les concepts de base********** le nombre de concepts de base dans le niveau 1 est 3 : les extensions et les intensions des concepts de base sont : extension du concept: ID= 0 X intension : 0 0 (* Type Point) 1 0 (* different Y) 2 0 (* Belongs Z) extension du concept: ID= 1 Y intension :

0 0 (* Type Point) 1 0 (X different *) 2 0 (* Belongs Z) extension du concept: ID= 2 Z intension : 0 0 (X Belongs *) 1 0 (Y Belongs *) 2 0 (* Type StraightLine) fin niveau 1

- Les concepts dérivés

********** **********concepts derivé********** debut niveau 1 extension du Concept: Classe X , Classe Y , 0 0 , (* Type Point) 1 0 , (* BinaryRelation phi) 2 0 , (* Belongs Z) 3 0 , extension du Concept: Classe X , Classe Y , Classe Z , 0 0 , (* Type phi) extension du Concept: Classe Y , Classe Z , 0 0 , (* Type phi) 1 0 , (X BinaryRelation *) 2 0 , fin niveau 1

La hiérarchie entre les concepts dérivés est présentée dans la figure suivante :

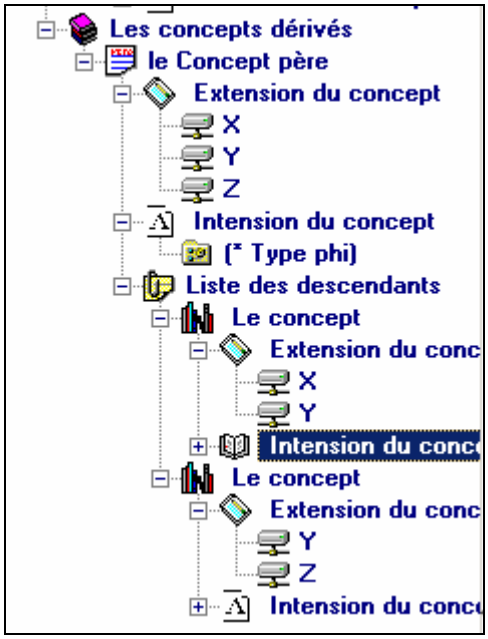


Figure 7. 10 : La hiérarchie des concepts dérivés de l'ontologie résultante.

- Ontologie RDFS de la longueur 1

```
<?xml version="1.0" ?>
- <Shema_RDFS Niveau="1">
- <Concepts_Base>
- <rdf_Bag>
- <rdf_statement>
<rdf_subject>X</rdf_subject>
<rdf_propriete>Type</rdf_propriete>
<rdf_object>Point</rdf_object>
</rdf_statement>
- <rdf_statement>
<rdf_subject>X</rdf_subject>
<rdf_propriete>Different</rdf_propriete>
<rdf_object>Y</rdf_object>
</rdf_statement>
- <rdf_statement>
<rdf_subject>X</rdf_subject>
<rdf_propriete>Belongs</rdf_propriete>
<rdf_object>Z</rdf_object>
</rdf_statement>
</rdf_Bag>
- <rdf_Bag>
- <rdf_statement>
<rdf_subject>Y</rdf_subject>
<rdf_propriete>Type</rdf_propriete>
<rdf_object>Point</rdf_object>
</rdf_statement>
```

- <rdf_statement>
  <rdf_subject>X</rdf_subject>
  <rdf_propriete>Different</rdf_propriete>
  <rdf_object>Y</rdf_object>
    </rdf_statement>
- <rdf_statement>
  <rdf_subject>Y</rdf_subject>
  <rdf_propriete>Belongs</rdf_propriete>
  <rdf_object>Z</rdf_object>
    </rdf_statement>
    </rdf_Bag>
- <rdf_Bag>
- <rdf_statement>
  <rdf_subject>Z</rdf_subject>
  <rdf_propriete>Type</rdf_propriete>
  <rdf_object>Straight_Line</rdf_object>
    </rdf_statement>
- <rdf_statement>
  <rdf_subject>X</rdf_subject>
  <rdf_propriete>Belongs</rdf_propriete>
  <rdf_object>Z</rdf_object>
    </rdf_statement>
- <rdf_statement>
  <rdf_subject>Y</rdf_subject>
  <rdf_propriete>Belongs</rdf_propriete>
  <rdf_object>Z</rdf_object>
    </rdf_statement>
    </rdf_Bag>
    </Concepts_Base>
- <Concepts_Dérivés>
- <rdf_Bag>
- <rdf_Bag>
- <rdf_statement>
- <rdf_subject>
- <rdf_Alt>
  <rdf_li>X</rdf_li>
  <rdf_li>Y</rdf_li>
    </rdf_Alt>
    </rdf_subject>
  <rdf_propriete>Type</rdf_propriete>
  <rdf_object>Point</rdf_object>
    </rdf_statement>
- <rdf_statement>
- <rdf_subject>
- <rdf_Alt>
  <rdf_li>X</rdf_li>
  <rdf_li>Y</rdf_li>
    </rdf_Alt>
    </rdf_subject>
  <rdf_propriete>Birary_Relation</rdf_propriete>
  <rdf_object>phi</rdf_object>
    </rdf_statement>
- <rdf_Bag>
- <rdf_statement>
- <rdf_subject>
- <rdf_Alt>
  <rdf_li>X</rdf_li>
  <rdf_li>Y</rdf_li>
    </rdf_Alt>
    </rdf_subject>
  <rdf_propriete>Belongs</rdf_propriete>
  <rdf_object>Z</rdf_object>
    </rdf_statement>
- <rdf_statement>
  <rdf_subject>Z</rdf_subject>

```
  <rdf_propriete>Type</rdf_propriete>
  <rdf_object>Straight_Line</rdf_object>
     </rdf_statement>
     </rdf_Bag>
- <rdf_statement>
- <rdf_subject>
- <rdf_Alt>
  <rdf_li>X</rdf_li>
  <rdf_li>Y</rdf_li>
     </rdf_Alt>
     </rdf_subject>
  <rdf_propriete>Type</rdf_propriete>
  <rdf_object>phi</rdf_object>
     </rdf_statement>
     </rdf_Bag>
     </rdf_Bag>
- <rdf_Bag>
- <rdf_Bag>
- <rdf_statement>
- <rdf_subject>
- <rdf_Alt>
  <rdf_li>X</rdf_li>
  <rdf_li>Z</rdf_li>
  <rdf_li>Y</rdf_li>
     </rdf_Alt>
     </rdf_subject>
  <rdf_propriete>Type</rdf_propriete>
  <rdf_object>phi</rdf_object>
     </rdf_statement>
     </rdf_Bag>
     </rdf_Bag>
- <rdf_Bag>
- <rdf_Bag>
- <rdf_statement>
- <rdf_subject>
- <rdf_Alt>
  <rdf_li>Y</rdf_li>
  <rdf_li>Z</rdf_li>
     </rdf_Alt>
     </rdf_subject>
  <rdf_propriete>Type</rdf_propriete>
  <rdf_object>phi</rdf_object>
     </rdf_statement>
- <rdf_statement>
  <rdf_subject>X</rdf_subject>
  <rdf_propriete>Birary_Relation</rdf_propriete>
- <rdf_object>
- <rdf_Alt>
  <rdf_li>Y</rdf_li>
  <rdf_li>Z</rdf_li>
     </rdf_Alt>
     </rdf_object>
     </rdf_statement>
- <rdf_statement>
  <rdf_subject>phi</rdf_subject>
  <rdf_propriete>Birary_Relation</rdf_propriete>
- <rdf_object>
- <rdf_Alt>
  <rdf_li>Y</rdf_li>
  <rdf_li>Z</rdf_li>
     </rdf_Alt>
     </rdf_object>
     </rdf_statement>
     </rdf_Bag>
     </rdf_Bag>
```

```
</Concepts_Dérivés>
</Shema_RDFS>
```

- Longueur 2 : Les concepts de base

```
**********Les concepts de base********** le nombre de concepts de base dans le niveau 2 est 3 : les extensions et les intensions des concepts de base sont : extension du concept: ID= 0 X intension : 0 0 (* Type Point) 1 0 (* different Y) 1 1 (Y Belongs Z) 2 0 (* Belongs Z) 2 1 (Z Type StraightLine) extension du concept: ID= 1 Y intension : 0 0 (* Type Point) 1 0 (X Belongs Z) 1 1 (X different *) 2 0 (* Belongs Z) 2 1 (Z Type StraightLine) extension du concept: ID= 2 Z intension : 0 0 (X different Y) 0 1 (X Belongs *) 1 0 (X different Y) 1 1 (Y Belongs *) 2 0 (* Type StraightLine) fin niveau 2
```

- Les concepts dérivés

```
********** **********concepts derivé********** debut niveau 2 extension du Concept: Classe X , Classe Y , 0 0 , (* Type Point) 1 0 , (* BinaryRelation phi) 2 0 , (* Belongs Z) 2 1 , (Z Type StraightLine) 3 0 , (* Type phi) extension du Concept: Classe X , Classe Y , Classe Z , 0 0 , (* Type phi) extension du Concept: Classe Y , Classe Z , 0 0 , (* Type phi) 1 0 , (X BinaryRelation *) 2 0 , (phi BinaryRelation *) fin niveau 2
```

- L'ontologie RDFS résultante de longueur 2

```
<?xml version="1.0" ?>
- <Shema_RDFS Niveau="2">
+ <Concepts_Base>
- <Concepts_Dérivés>
- <rdf_Bag>
- <rdf_Bag>
- <rdf_statement>
- <rdf_subject>
- <rdf_Alt>
  <rdf_li>X</rdf_li>
  <rdf_li>Y</rdf_li>
    </rdf_Alt>
    </rdf_subject>
  <rdf_propriete>Type</rdf_propriete>
  <rdf_object>Point</rdf_object>
    </rdf_statement>
- <rdf_statement>
- <rdf_subject>
- <rdf_Alt>
  <rdf_li>X</rdf_li>
  <rdf_li>Y</rdf_li>
    </rdf_Alt>
    </rdf_subject>
  <rdf_propriete>Birary_Relation</rdf_propriete>
  <rdf_object>phi</rdf_object>
    </rdf_statement>
- <rdf_Bag>
- <rdf_statement>
- <rdf_subject>
- <rdf_Alt>
  <rdf_li>X</rdf_li>
  <rdf_li>Y</rdf_li>
    </rdf_Alt>
    </rdf_subject>
  <rdf_propriete>Belongs</rdf_propriete>
  <rdf_object>Z</rdf_object>
    </rdf_statement>
- <rdf_statement>
  <rdf_subject>Z</rdf_subject>
  <rdf_propriete>Type</rdf_propriete>
  <rdf_object>Straight_Line</rdf_object>
    </rdf_statement>
```

```
        </rdf_Bag>
- <rdf_statement>
- <rdf_subject>
- <rdf_Alt>
  <rdf_li>X</rdf_li>
  <rdf_li>Y</rdf_li>
        </rdf_Alt>
        </rdf_subject>
  <rdf_propriete>Type</rdf_propriete>
  <rdf_object>phi</rdf_object>
        </rdf_statement>
        </rdf_Bag>
        </rdf_Bag>
- <rdf_Bag>
- <rdf_Bag>
- <rdf_statement>
- <rdf_subject>
- <rdf_Alt>
  <rdf_li>X</rdf_li>
  <rdf_li>Z</rdf_li>
  <rdf_li>Y</rdf_li>
        </rdf_Alt>
        </rdf_subject>
  <rdf_propriete>Type</rdf_propriete>
  <rdf_object>phi</rdf_object>
        </rdf_statement>
        </rdf_Bag>
        </rdf_Bag>
- <rdf_Bag>
- <rdf_Bag>
- <rdf_statement>
- <rdf_subject>
- <rdf_Alt>
  <rdf_li>Y</rdf_li>
  <rdf_li>Z</rdf_li>
        </rdf_Alt>
        </rdf_subject>
  <rdf_propriete>Type</rdf_propriete>
  <rdf_object>phi</rdf_object>
        </rdf_statement>
- <rdf_statement>
  <rdf_subject>X</rdf_subject>
  <rdf_propriete>Birary_Relation</rdf_propriete>
- <rdf_object>
- <rdf_Alt>
  <rdf_li>Y</rdf_li>
  <rdf_li>Z</rdf_li>
        </rdf_Alt>
        </rdf_object>
        </rdf_statement>
- <rdf_statement>
  <rdf_subject>phi</rdf_subject>
  <rdf_propriete>Birary_Relation</rdf_propriete>
- <rdf_object>
- <rdf_Alt>
  <rdf_li>Y</rdf_li>
  <rdf_li>Z</rdf_li>
        </rdf_Alt>
        </rdf_object>
        </rdf_statement>
        </rdf_Bag>
        </rdf_Bag>
        </Concepts_Dérivés>
        </Shema_RDFS>
```

7.3.Conclusion

Nous avons présenté, au cours de ce chapitre, deux tests détaillés en utilisant DLOT pour la l'édition des ontologies et des graphes RDF initiaux et l'apprentissage des nouveaux concepts et hiérarchies suivant des longueurs choisis. Ces deux tests sont : le premier c'est une application dans le domaine de médecine, c'est le même exemple vu dans la partie théorique, et le deuxième test est pour l'ontologie de la géométrie projective et un cas d'apprentissage qui concerne la relation d'appartenance.

Pour chaque test nous avons présenté l'ontologie et le graphe, initiaux, puis pour chaque longueur nous avons montré les concepts de bases, les concepts dérivés, la hiérarchie entre ces concepts et l'ontologie RDFS résultante.

CONCLUSION GENERALE

Le Web sémantique a pour objectif de transformer le World Wide Web actuel, entièrement tourné vers la présentation des documents, vers un Web dont le contenu serait compréhensible aux machines.

La vision s'appuie sur l'utilisation :

- d'ontologies, qui sont des conceptualisations communes et partagées entre différents agents qui présente une vue de monde réel sur un domaine spécifique, L'ontologie permet de modéliser les connaissances partagées de la communauté. Elle représente la structure de connaissances du domaine spécifique de façon compréhensible par machines. Tous les concepts et leurs relations du domaine sont représentés explicitement par les classes et les instances,…etc.
- et d'un langage de description, permettant d'exprimer à la fois les définitions des concepts et des relations dans les ontologies et d'exprimer des annotations utilisant le vocabulaire de ces ontologies.

Ce langage et ces ontologies permettraient aux agents informatiques de comprendre les diverses annotations et de communiquer entre eux, en effectuant des raisonnements sur les concepts.

Pour représenter une ontologie du web sémantique il faut utiliser un langage permettant de décrire l'environnement du web sémantique telles que les ressources web. Le langage que nous parait indispensable est le RDF.

Le RDF est un langage formel qui permet d'affirmer des relations entre des ressources. Ces affirmations sont représentées par des déclarations sur ces ressources, les déclarations sont des triplets de la forme <*sujet, prédicat, objet*>.

Un document RDF sera codé en machine par un document RDF/XML, mais est souvent représenté sous une forme graphique.

Les éléments de ces triplets peuvent être des URIs, des littéraux ou des variables. Un tel ensemble peut être représenté de façon naturelle par un graphe, où les éléments apparaissant comme sujet ou objet sont les sommets, et chaque propriété est représentée par un arc dont l'origine est son sujet et la destination son objet.

RDF aura un système de classe comme dans tout environnement de modélisation orienté objet. Une collection de classes écrite pour un domaine ou un but spécifique est appelée un *schéma*. Les classes sont organisées en hiérarchie. Cette structure (classe et hiérarchie) nous permet de représenter une ontologie.

L'utilisation d'ontologies pour le Web Sémantique entraîne un besoin de techniques de gestion de ces ontologies, notamment pour les mettre à jour et les faire évoluer.

Il serait intéressant d'enrichir des ontologies avec des concepts générés en exploitant les annotations RDF. Il s'agit de l'apprentissage d'ontologies.

L'apprentissage d'ontologies est un processus fondamental au sein du web sémantique. En effet, il facilite, l'enrichissement des ontologies.

Nous avons montré dans ce mémoire une construction automatique de classes au sein du formalisme de description de ressources RDF. Pour cela nous avons défini une méthode pour la construction automatique des classe et leurs hiérarchie qui permet une conceptualisation automatique d'une ontologie.

Cette méthode prend en considération la complexité structurelle des concepts où chaque concept est représenté par un graphe. Ainsi, plutôt que de créer une nouvelle méthode, nous avons fait appel a des techniques connues et validées de construction de classes tout en définissant un nombre d'outils et de mécanismes à son application directe sur les concepts.

Nous avons présenté un algorithme qui permet une conceptualisation automatique d'une ontologie. Cet algorithme repose sur des descriptions des ressources. Il consiste à construire l'espace des généralisations maximalement spécifiques d'ordre 1 des descriptions des objets puis, dans un second temps, à enrichir cet espace par des descriptions plus complexes. Cette deuxième étape se base sur un enrichissement progressif des longueurs des descriptions extraites à partir du graphe RDF initial.

Nous avons proposé un algorithme d'apprentissage de concepts et d'ontologie pour le Web Sémantique. Notre algorithme est incrémental par rapport à longueur des descriptions des ressources. Dans ce travail nous avons développé une méthode d'apprentissage incrémentale

de type Bottom-Up à partir d'exemples qui sont des cas d'apprentissage. L'apprentissage se fait à partir d'une ontologie RDFS et un cas d'apprentissage qui est un graphe RDF définit par l'utilisateur. L'algorithme cherche les nouveaux concepts qui sont plus spécifique au domaine et plus complexes et cherche aussi la hiérarchie entre ces concepts. L'algorithme ayant un processus de spécialisation, qui lui garantit de rester dans l'espace des généralisations maximalement spécifiques entre les concepts. L'algorithme étant incrémental, permettant à chaque étape de trouver un ensemble de concepts qui sont plus spécifique aux concepts trouvés dans l'étape précédente et plus généraux que l'étape qui se suit l'étape courante.

Nous avons montré dans ce mémoire une méthode d'apprentissage des concepts pour la construction des ontologies en utilisant le langage de description des ressources du web RDF, cette méthode permet une conceptualisation automatique d'une ontologie en utilisant l'apprentissage, c'est à partir d'une ontologie initiale représenté par un RDFS et d'un cas d'apprentissage qui est un graphe RDF nous construisons une nouvelle hiérarchie entre de nouveaux concepts, nous avons utilisé le conceptual clustering, tout en définissant un outil pour l'édition des ontologies et pour l'apprentissage.

La méthode proposée est fondée sur le regroupement conceptuel des objets, la construction de la hiérarchie est incrémentale en fonction de la taille des descriptions. Les principaux éléments de notre travail peuvent être résumés en quelques points :

- *Mesure de similarité (subsomption) entre graphes par la définition d'un principe de comparaison entre objets structurés* : lorsque les objets sont représentés par des graphes, qui sont utilisés pour modéliser des objets structurés, ces objets sont modélisés par des graphes étiquetés où les sommets représentent les composants des objets et les arcs représentent les relations entre ces composants. Ce problème se ramène à mesurer la similarité entre graphes. Comparer deux objets revient alors à comparer deux graphes (problème de subsomption) : Cette comparaison peut se faire à travers la recherche de (sous)graphes afin de montrer l'existence d'une relation de généralisation entre les deux graphes.

- *Définition d'une méthode de construction de classes* qui sépare le processus en deux phases : construction des concepts par les regroupements possibles entre les ressources et construction de la hiérarchie entre ces concepts.

- Les limites de l'approche

Si notre approche présente de nombreux intérêts, elle possède également des limites. La limite de l'algorithme est liée à la complexité algorithmique de la méthode d'apprentissage d'ontologie. Cette complexité est due à la structure graphique des objets, et par conséquent le problème du calcul de la subsomption entre deux graphes. En effet, le calcul de la subsomption à un certain longueur demande la vérification de tout les sous graphe partiels contenue dans les graphes initiaux. Il y a donc un compromis entre la complexité structurelle des objets que l'on veut traiter et le coût algorithmique.

- Perspectives **:** les perspectives de ce travail peuvent être résumé en :

❖ L'étape d'opérationnalisation d'ontologie : le travail effectue dans ce dans ce mémoire concerne l'étape de conceptualisation d'une ontologie qu'elle consiste à trouver un modèle hiérarchique entre les concepts, il s'agit d'une conceptualisation automatique en utilisant le langage RDF.

Au cours de processus de développement d'ontologie nous avons vu qu'après la conceptualisation d'une ontologie, il arrive l'étape d'opérationnalisation où il faut utiliser les mécanismes d'inférence et de raisonnement.

L'étape d'opérationnalisation reste comme perspective pour ceux qui veulent opérationnaliser notre ontologie pour un domaine spécifique.

❖ Limites de RDF(S) : nous avons vu que RDF et RDFS permettent de définir, sous forme de graphes de triplets, des données ou des métadonnées. Cependant, de nombreuses limitations bornent la capacité d'expression des connaissances établies à l'aide de RDF/RDFS. On peut citer, par exemple, l'impossibilité de raisonner et de mener des raisonnements automatisés (automated reasoning) sur les modèles de connaissances établis à l'aide de RDF/RDFS. C'est ce manque que se propose de combler OWL.

RDF est

- un modèle de métadonnées puissant,
- largement accepté et utilisé,
- mais trop limité pour :
  - formuler des contraintes sémantiques plus riches et
  - raisonner.

RDFS ne fournit que des mécanismes très primitifs pour spécifier ces classes. OWL fournit un grand nombre de constructeurs permettant d'exprimer de façon très fine les propriétés des classes définies.

RDF Schema définit un minimum de notions et de propriétés nécessaires à la définition d'une ontologie :

- classe, ressource, litéral
- sous-classe, sous-propriété
- OWL "étend RDFS" avec des notions de
    - classe et propriété équivalente,
    - d'identité d'objet,
    - propriétés symétriques, transitives, cardinalité, etc.

OWL a été conçu pour permettre différents types de raisonnements, plus ou moins complexes. Il est considéré par W3C comme une langage d'ontologie standard. Il a non seulement la capacité de décrire les concepts dans un domaine mais aussi d'un ensemble plus riche d'opérateurs, donc ces concepts bien définis et bien décrits. On peut construire des concepts complexes en basant les définitions des concepts plus simples. En outre, on peut vérifier si tous les rapports et les définitions dans l'ontologie sont conformés et identifier quels concepts s'adaptent sous quelles définitions. Donc, on peut maintenir la hiérarchie correctement entre les classes.

OWL est, tout comme RDF, un langage XML profitant de l'universalité syntaxique de XML. Fondé sur la syntaxe de RDF/XML. OWL se différencie du couple RDF/RDFS en ceci que, contrairement à RDF, il est justement un langage d'ontologies. Si RDF et RDFS apportent à l'utilisateur la capacité de décrire des classes et des propriétés, OWL intègre, en plus, des outils de comparaison des propriétés et des classes : identité, équivalence, contraire, cardinalité, symétrie, transitivité, disjonction, etc.

Ainsi, OWL offre aux machines une plus grande capacité d'interprétation du contenu web que RDF et RDFS, grâce à un vocabulaire plu.s large et à une vraie sémantique formelle.

Notre travail est réalisé par le RDF, il reste comme perspective de travailler avec le langage OWL à la place de RDF pour permettre de faire des raisonnements grâce à son large vocabulaire et sa sémantique formelle.

REFERENCES

1. FRÉDÉRIC FÜRST Opérationnalisation d'une ontologie de la géométrie projective, Rapport de stage de recherche, DEA Informatique 2000-2001.

2. T.R. GRUBER. Toward principles for the Design Of Ontologies used for Knowledge Sharing. In « Formal Ontology in Conceptual Analysis and Knowledge Representation » (GUARINO, POLI). Kluwer Academic Publishers.

3. Michel LECLÉRE, Francky TRICHET et Frédéric FÜRST IRIN, Construction d'une ontologie du domaine de la géométrie projective, 2002.

4. La Création d'Ontologies Web Sémantique avec Protégé-2000. http://www.cetic.be/indexEN.php3

5. Jean Charlet, developpements, résultats et perspectives pour la gestion des connaissances medicales , Mémoire d'Habilitation à diriger des recherches, l'Université Pierre et Marie Curie,2002.

6. C. Roche, Condillac Group, 2004.
http://ontology.univsavoie.fr/activites/recherche/ontologie/notreDefinitionEng.asp

7. J.F. Sowa. KR Ontology. http://www.jfsowa.com/ontology/ , 2004.

8. http://www.ontology.org/

9. http://ontoserver.aifb.uni-karlsruhe.de/ontoedit/

10. http://ontology.univ-savoie.fr/okstation/article/okmodel.html

11. http://ontobroker.aifb.uni-karlsruhe.de/

12. http://www.ksl.Stanford.EDU/software/chimaera/

13. http://kmi.open.ac.uk/projects/webonto/

14. Xavier Petard, systemes collectifs et mediation outillage du travail coloboratif par augmentation des communications, LIMSI-CNRS.

15. Valéry Psyché, Olavo Mendes, Jacqueline Bourdeau, Apport de l'ingénierie ontologique aux environnements de formation à distance, 2003.

16. DERIC FURST, L'ingénierie ontologique, IRIN, Universite' de Nantes, 2002.

17. Hector Bustillo et Jérôme Blanc**,** Ontologies, 2003. http://www.communautic.uqam.ca/mba8t1a/pmwiki.php/Groupe3/Ontologie

18. Jean-Paul Sansonnet, Knowledge Representation : langages et Formalismes de représentation des connaissances, LIMSI-CNRS.

19. Alexandre Delteil, Représentation et apprentissage de concepts et d'ontologies pour le Web S´emantique, UNIVERSITE DE NICE-SOPHIA ANTIPOLIS- UFR Sciences ,2002.

20. Jean-François Perrot, Introduction à l'Intelligence Artificielle 4. LISP : Les symboles et leur P-liste, Université Pierre et Marie Curie, Paris.

21. lingway, pas de web sémantique sans traitement automatique des langues, octobre 2004.

22. LUONG Phuc Hiep, Graphes Conceptuels Flous.

23. Alexander Maedche and Steffen Staab, Ontology Learning for the Semantic Web, IEEE Intelligent Systems 16 (2). March 2001.University of Karlsruhe, Germany.

24. Borys Omelayenko,learning of ontologies for the web : analysis of existent approaches, Vrijie universiteit Amsterdam ,2001.

25. Alexander Maedche, Viktor Pekar, and Steffen Staab , Ontology Learning Part One On Discovering Taxonomic Relations from the Web ,University of Karlsruhe Germany , 2003.

26. Asunción Gómez-Pérez, David Manzano-Macho, A survey of ontology learning methods and techniques ,Universidad Politécnica de Madrid ,2003.

27. Xavier Lacot, Introduction à OWL, un langage XML d'ontologies Web, Juin 2005.

28. HUỲNH HỮU HƯNG, Developpement d'un outil efficace pour annoter des documents, 2003.

29. Web sémantique, Laboratoire de Recherche en Informatique - Équipe IASI , CNRS, 2002 .

30. http://www.w3.org/TR/2004/REC-rdf-schema-20040210/ ,2004.

31. http://www.w3.org/TR/2004/REC-rdf-concepts-20040210 , 2004.

32. Spécification du modèle et la syntaxe du cadre de description des ressources (Resource Description Framework ou RDF), La version française de cette traduction est : http://www.la-grange.net/w3c/REC-rdf-syntax/
Version originale : http://www.w3.org/TR/1999/REC-rdf-syntax-19990222

33. http://www.w3.org/TR/rdf-concepts/

34. Raphaél TRONCY, Intégration texte représentation formelle pour la gestion de documents XML, INRIA ,2000.

35. Jean-François Baget, Étienne Canaud, Jérôme Euzenat, Mohand Saïd-Hacid, Les langages du web sémantique, INRIA, LISI, 2003.

36. HerveTigier , Le web sémantique http://websemantique.org/PagePrincipale, 2004.

37. REZE S.PICAROUGNE C, Un outil d'aide à l'annotation de Curriculum Vitae par exploitation d'ontologies de compétences, 2003 http://www.sciences.univ-nantes.fr/irin/commoncv/documents/CVOntoAnnot.doc

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

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

40. Eric SARDET, Intégration des approches modélisation conceptuelle et structuration documentaire pour la saisie, la représentation, l'échange et l'exploitation

d'informations. Application aux catalogues de composants industriels, thèse Pour l'obtention du grade de docteur de l'université de Poitiers ,1999.

41. Vincent QUINT, documents structurés sur le web, structured documents on the web, W3C /INRIA, 2002.

42. Mohamed Mahdi Malik, le rôle de la logique floue dans le web sémantique, disponible sur Internet sur l'adresse : http://www.dil.univ-mrs.fr/dea/dea2002/memoires/malik.pdf,2002.

43. Tim Berners-Lee, James Hendler, Ora Lassila, The Semantic Web, Scientific American, May 2001. disponible sur internt sur l'adresse www.sciam.com/2001/0501issue/0501berners-lee.html

44. Regis Nacfaire. "Xyleme se lance sur le marché de la recherche et de l'intégration de contenus XML". Digital Business Globe, www.xyleme.com/fr/presse/doc1.pdf, mars 2002.

45. Xml vers un format universel ? http://www.dsi.cnrs.fr/bureau_qualite/developpement-web/technologie/ XML/XML2001.pdf , 2001.

46. Alain TESTE, Web Sémantique pour le E-learning dans le cadre d'environnements d'apprentissage coopératif à base de situation-problème , Université de Pau et des Pays de l'Adour , 2004.

47. Ollivier Haemmerlé, Des graphes conceptuels pour la représentation de connaissances en microbiologie, Université Paris Sud – Orsay, 2004.

48. Benjamin Nguyen, Construction de Classes de Documents Web, INRIA FUTURS, France, 2003.

49. David Faurey, Claire Nedellec, Celine Rouveirol, Acquisition de connaissances sémantiques par des méthodes d'apprentissage le système asium, 2002.

50. Paul BALEZ, Mathieu BEAL, Algorithmes de Datamining ,21 février 2002.

51. Yves KODRATOFF, techniques et outils de l'extraction de connaissances à partir des données , Paris, 1998.

52. Gilles Bisson, Evaluation & Catégorisation, Equipe SICLAD CNRS-INPG-UJF.

53. Sébastien Paquet, Apprentissage de la coordination dans un système multiagent Examen de synthèse Par, Université Laval, 2002.

54. BART Pascal et FRIGOT Eric, Algorithmes non supervisé, EPITA, 2002.

55. Laurent CANDILLIER, La classification non supervisé, lille, 2004.

56. Periklis ANDRITSOS , Scalable Clustering of Categorical Data and Applications, University of Toronto , 2004.

57. GLUSEN demiroz, Non incremental classification learning algorithms based on voting features intervals , université de bilkent ,1997.

58. Petko Valtchev, Construction automatique de taxonomies pour l'aide à la représentation de connaissances par objets, Thèse pour l'obtention du Doctorat de l'Université Joseph Fourier - Grenoble .Spécialité informatique ,1999.

59. Concepts formation in structured domains, kevin thompson et pat langley.

60. Darrel Conklin , Structured concepts discovery : theory and methods, queen's university, Canada,1994.

61. Dominique BOUTHINON, Henry SOLDANO et Veronique VENTOS, Aspects théoriques et pratiques de l'apprentissage en cas d'incomplétude des données : Etendre l'apprentissage par satisfiabilité, Atelier de BioInformatique Paris, France, 2000.

62. Philippe 62TIN , THÈSE pour obtenir le titre de DOCTEUR EN SCIENCES , Exploitation de graphes conceptuels et de documents structurés et hypertextes pour l'acquisition de connaissances et la recherche d'informations,université de NICE, Sophia Antipolis , 1996.

63. Frédéric FÜRST, THÈSE pour obtenir le grade de DOCTEUR DE L'UNIVERSITÉ DE NANTES : Contribution à l'ingénierie des ontologies : une méthode et un outil d'opérationalisation ,2004.

64. Marianne HUCHARD, Classification de classes dans les approches à objets : algorithmes et applications, 2003.

65. Serge HADDAD, réseaux sémantique et graphes conceptuels, université paris-dauphine ,2002.

66. D.HILBERT. Les fondements de la géométrie (Translation of « Grundlagen der geometrie »).Editions Jacques Gabay, 1997.

67. http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/#section-Graph-URIref.