

République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études

Pour l'obtention du diplôme de Licence en Informatique

Thème

**La création d'une ontologie
pharmaceutique**

Réalisé par :

Meriem KARAOUZENE

Fatima Zohra BERRADANE

Présenté le 27 Juin 2013 devant la commission d'examination composée de MM.

- BENMAMMAR Badr..... (Examineur)
- BELABED Amine..... (Examineur)
- MERZOUG Mohammed..... (Examineur)

Année universitaire : 2012-2013

Remerciement

Qu'il nous soit permis de présenter ici nos remerciements à tout un petit monde de personnes qui ont rendu possible la présente étude et qui ont contribué à son élaboration sous quelque forme que ce soit.

Nous tenons tout d'abord à dire notre reconnaissance envers Monsieur Belabbed Amine qui a accepté sans réserve, de diriger ce modeste travail. Il s'est grandement impliqué par son savoir, ses directives, ses remarques et suggestions, mais aussi par ses encouragements dans les moments clés de son élaboration. Nous tenons à le remercier aussi pour cette liberté qu'il nous a permise, sans laquelle le chercheur ne saurait affirmer sa manière de penser et de procéder, sa manière d'être, bref toute sa personnalité.

Nos remerciements s'étendent également à nos familles, amis et collègues membres du Groupe de Recherche du laboratoire d'informatique de la Faculté de l'ingénieur, sous la direction de Monsieur Benammar, ils sont également chaleureusement remerciés pour avoir lu et commenté presque la totalité du travail. Leurs remarques étaient d'une aide précieuse.

Nous ne manquerons pas non plus de dire un grand merci aux membres du jury Monsieur Benmammar Badr, Merzoug Mohammed d'avoir accepté, sans réserve aucune, d'évaluer cette thèse à sa juste valeur, et de nous faire part de leur remarques sûrement pertinentes qui, avec un peu de recul, contribueront, sans nul doute, au perfectionnement du présent travail.

Dédicaces

À Nos Très Chers Parents

Tous les mots du monde ne sauraient exprimer l'immense amour que nous vous portons, ni la profonde gratitude que nous vous témoignons pour tous les efforts et les sacrifices que vous n'avez jamais cessé de consentir pour notre instruction et notre bien-être.

C'est à travers vos encouragements que nous avons opté pour cette noble profession, et c'est à travers vos critiques que nous nous sommes réalisés.

Nous espérons avoir répondu aux espoirs que vous avez fondés en nous.

Nous vous rendons hommage par ce modeste travail en guise de notre reconnaissance éternelle et de notre infini amour.

Vous résumez si bien le mot parents qu'il serait superflu d'y ajouter quelque chose.

Que Dieu tout puissant vous garde et vous procure santé, bonheur et longue vie pour que vous demeuriez le flambeau illuminant le chemin de vos enfants.

À Tous ceux qui nous ont aidés de près ou de loin ;

Finalement à tous ceux qui nous portent dans leurs cœurs.

TABLE DES MATIERES

Introduction générale :	1
Chapitre 1 : LES ONTOLOGIES	2
I. INTRODUCTION	2
II. HISTORIQUE :	2
III. DEFINITION	3
III.1 Approche opérationnelle	4
IV. COMPOSANTES D'UNE ONTOLOGIE	5
V. LES ETAPES DE CONSTRUCTION D'ONTOLOGIES :.....	7
V.1 Objectif et critères d'évaluation d'une ontologie :.....	7
VI. AVANTAGES D'UNE ONTOLOGIE :.....	9
VII. DIFFERENTES SORTES D'ONTOLOGIES :.....	9
VII.1 Objet de conceptualisation :	9
VII.2 Niveau de formalisme de représentation :.....	10
VIII. OUTILS DE MANIPULATION DES ONTOLOGIES.....	11
CONCLUSION :	12
CHAPITRE 2 : LES LANGAGES DES ONTOLOGIES	13
I. LANGAGES POUR LES ONTOLOGIES	13
II. LES STANDARDS : XML RDF, ET RDFS	13
II.1 XML:	13
II.2 RDF:	14
II.3 RDF Schéma:	15
III OWL:	16
III.1 Une extension de RDF et RDFS	17
III.2 Les niveaux d'OWL.....	17
III.3 Spécifications OWL existants	18
Conclusion :	18
CHAPITRE 3 : CONCEPTION ET IMPLEMENTATION	19
I. INTRODUCTION.....	19
II. CONCEPTION ET IMPLEMENTATION DES ONTOLOGIES :.....	19
II.1 Présentation de l'ontologie :.....	19
II.2 Conception de l'application :	21
II.3 Outils d'implantation:	22
II.4.1 le langage JAVA :.....	22
II.4.2 NetBeans.....	22
II.4.3 Jena :	22
II.4 Interfaces de l'application	22
CONCLUSION	25
Conclusion générale	26
Reference bibliographique	27

Tables des figures

FIGURE I.1 - EXEMPLE D'ONTOLOGIE [3] -----	3
FIGURE I. 2: EXEMPLE DES RELATIONS ET CONCEPTS [7] -----	6
FIGURE I.3 : ONTOLOGIE D'UNE COMPOSITION D'UNE VOITURE -----	8
FIGURE II.1 SCHEMATIQUE COURANTE DU TRIPLET RDF [16]-----	14
FIGURE II.2 EXEMPLE DE TRIPLET RDF[16]-----	14
FIGURE II.3 LE GRAPHE RDF ET RDFS -----	16
FIGURE II.1 UNE PARTIE D'ONTOLOGIE -----	20
FIGURE II.2 DIAGRAMME DE CAS D'UTILISATION. -----	21
FIGURE II.3 DIAGRAMME DE CLASSE. -----	21
FIGURE II.3: INTERFACE PRINCIPALE.-----	23
FIGURE III.3: L'AFFICHAGE DE L'ONTOLOGIE EN OWL.-----	23
FIGURE III.4: LE GRAPHE D'AFFICHAGE DE ONTOLOGIE -----	24
FIGURE III.5: AJOUT D'UN MEDICAMENT-----	24

Résumé

Dans ce mémoire nous avons présenté les phases d'implémentation d'une ontologie qui offre des services aux utilisateurs dans le domaine pharmaceutique, etc.... pour ce faire nous avons développé un logiciel pour gérer cette ontologie... Notre application fournit les fonctionnalités d'ajout des concepts et des relations ainsi que l'affichage de l'ontologie. Pour le développement de cette application nous avons utilisé le langage Java et la bibliothèque Jena.

Mots-clés ontologie, OWL.

Abstract

In this paper we presented the implementation phases of an ontology which provides services to users in the pharmaceutical industry, etc. to do this we have developed a software to manage the ontology ... Our application provides functionality for adding concepts and relationships as well as the display of the ontology. For the development of this application we used the Java language and the Jena library.

Keywords: OWL, ontology.

ملخص

في هذا العمل قمنا بتبيان مراحل التنفيذ الأنطولوجيا التي توفر للمستخدمين في مجال الصيدلة... الخ للقيام بذلك قمنا بتطوير برنامج لإدارة الأنطولوجيا ويوفر التطبيق لدينا وظائف لإضافة مفاهيم وعلاقات وكذلك عرض الأنطولوجيا لتطوير هذا التطبيق استخدمنا لغة جافا والمكتبة جينا

كلمات البحث. الأنطولوجيا OWL

Introduction générale :

Le concept d'ontologie existe depuis très longtemps, notamment en philosophie. en informatique ce concept est associé à la définition d'un vocabulaire compréhensible par des machines.

Une ontologie définit les termes utilisés pour décrire et représenter un champ d'expertise. Les ontologies sont utilisées par les personnes, les bases de données, et les applications qui ont besoin de partager des informations relatives à un domaine bien spécifique, comme la médecine, la fabrication d'outils, l'immobilier, la réparation d'automobiles, la gestion de finances, etc. Les ontologies associent les concepts de base d'un domaine précis et les relations entre ces concepts, tout cela d'une manière compréhensible par les machines. Elles encodent la connaissance d'un domaine particulier ainsi que les connaissances qui recouvrent d'autres domaines, ce qui permet de rendre les connaissances réutilisables.

Dans ce travail nous avons réalisé une application qui gère une ontologie pharmaceutique en créant des classes et sous classe avec des relations

Ce document est organisé comme suit :

Le chapitre 1 : Ce chapitre présente une vue générale sur les ontologies ainsi que quelques outils qui les mettent en œuvre.

Le chapitre 2 : Ce chapitre présente quelques langages d'ontologies XML, RDF, RDFS, OWL

Le chapitre 3 : Ce chapitre est consacré pour la présentation de l'application réalisée.

Et enfin une conclusion générale qui résume notre travail et présente quelques perspectives.

Chapitre 1

Les ontologies

Premier chapitre est consacré aux ontologies, leurs définitions, historiques cycles de vie etc...

I. Introduction

Le terme « ontologie », construit à partir des racines grecques ontos (ce que existe, l'existant) et logos (le discours, l'étude), est un mot que l'informatique a emprunté à la philosophie au début des années 1990. En philosophie, l'Ontologie est une branche fondamentale de la Métaphysique, qui s'intéresse à la notion d'existence, aux catégories fondamentales de l'existant et étudie les propriétés les plus générales de l'être.

Par analogie, le terme est repris en informatique et en science de l'information, où une ontologie est l'ensemble structuré des termes et concepts représentant le sens d'un champ d'informations, que ce soit par les métadonnées d'un espace de noms, ou les éléments d'un domaine de connaissances.

II. Historique :

La branche de l'intelligence artificielle est qualifiée de symbolique parce qu'elle repose sur des représentations formelles des connaissances, sous la forme de symboles que le système peut stocker et manipuler (par exemple, langages et opération logiques, structures et opérations de graphes). Contrairement à d'autres approches, ces représentations sont à la fois compréhensibles par les humains et manipulables par les systèmes, en appliquant des règles de manipulation définies sur les symboles de ces représentations et dont l'interprétation simule, par exemple, un raisonnement.

Ainsi, dès les années 1970, la notion d'ontologie existait, sans être nommée et de façon transversale, dans les différents systèmes de représentation de connaissances : c'est la TBox des logiques de description, où l'on décrit les types de termes qui existent dans la représentation et leurs caractéristiques ; c'est le support des graphes conceptuels, où l'on décrit des hiérarchies de

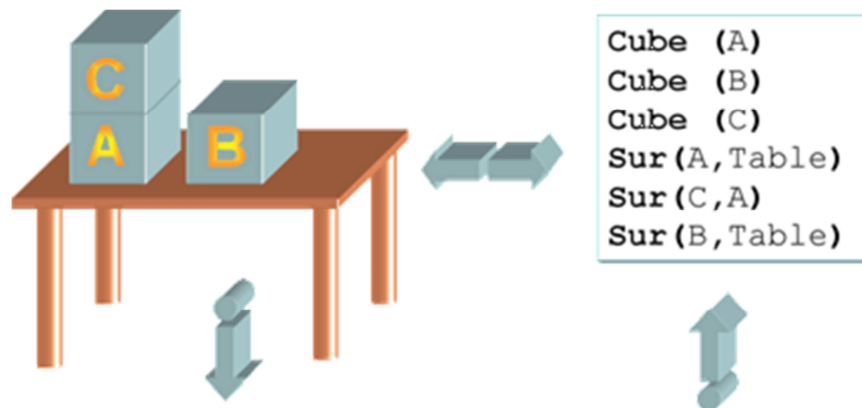
multi-héritage entre des types de concepts ou des types de relations ; ce sont enfin les schémas des « Frames » et les classes des langages de représentation par objets.

Il aura pourtant fallu attendre les années 1990 pour que le mot « ontologie » soit adopté par toute la communauté, et sa définition fait encore couler de l'encre (électronique). [1]

III. Définition

Une ontologie informatique est une représentation de propriétés générales de ce qui existe dans un formalisme supportant un traitement rationnel. C'est le résultat d'une formulation exhaustive et rigoureuse de la conceptualisation d'un domaine. Cette conceptualisation est souvent qualifiée de partielle car, en l'état de l'art, il est illusoire de croire pouvoir capturer dans un formalisme toute la complexité d'un domaine. Notons aussi que le degré de formalisation d'une ontologie varie avec l'usage qui en est envisagé.

C'est pour cet aspect de description de l'existant et de ses catégories que les ontologies informatiques ont emprunté leur nom à l'Ontologie philosophique. De ce rapprochement vient aussi la possibilité d'adapter des méthodes de la philosophie pour proposer des méthodes d'ingénierie d'ontologies.[2].



Cube (x) : L'entité X est parallélépipède à angle droits avec toutes ses cotes de même longueur.

Table : Un objet global qui est un meuble composé d'une planche horizontale posée sur un ou plusieurs pieds.

Sur (cube : X, Cube : Y/ Table) : une relation dénotant qu'un cube Y ou sur la table.

Figure I.1 - Exemple d'Ontologie [3]

Pour faire simple, considérons l'exemple de la figure ci-dessus. On y voit une certaine « scène du monde ». La description de cette scène demande deux choses : un vocabulaire non ambigu, aussi appelé vocabulaire conceptuel ou ontologie ; une énonciation des faits de la scène, reposant sur l'utilisation du vocabulaire de l'ontologie.

D'un point de vue pratique, une ontologie informatique permet, en particulier grâce aux travaux de l'intelligence artificielle symbolique sur les systèmes à base de connaissances et les moteurs d'inférence, d'implanter des mécanismes de raisonnement déductif, de classification automatique, de recherche d'information, et d'assurer l'interopérabilité entre plusieurs systèmes de ce type.

III.1 Approche opérationnelle

Parallèlement à cette définition de ce que représente une ontologie, une autre définition, plus opérationnelle, peut être formulée ainsi :

Une ontologie est un réseau sémantique qui regroupe un ensemble de concept décrivant complètement un domaine. Ces concepts sont liés les uns aux autres par des relations taxinomiques (hiérarchisation des concepts) d'une part, et sémantiques d'autre part.

Cette définition rend possible l'écriture de langages destinés à implémenter des ontologies. Pour construire une ontologie, on dispose d'au moins trois de ces notions :

- détermination des agents passifs ou actifs.
- leurs conditions fonctionnelles et contextuelles.
- leurs transformations possibles vers des objectifs limités.

Pour modéliser une ontologie, on utilisera ces outils (mécanismes) :

- Raffiner les vocabulaires et notions adjacentes.
- Décomposer en catégories et autres sujets.
- Prédiquer afin de connaître les transformations adjacentes et d'orienter vers les objectifs internes.
- Relativiser afin d'englober des concepts.
- Similiser afin de réduire à des bases totalement distinctes.
- Instancier afin de reproduire l'ensemble d'une « branche » vers une autre ontologie. [4]

IV. Composantes d'une ontologie

Comme tout formalisme de représentation, les ontologies sont basées sur l'utilisation d'un certain nombre de composantes (dites aussi briques ou constituants) de base, véhiculant avec eux les connaissances traduites par ces dernières et qui sont principalement : Concept, Relation, Fonction, Axiomes, Instance.

a. Les concepts :

Aussi appelés termes ou classes de l'ontologie, constituent les objets de base manipulés par les ontologies. Ils correspondent aux abstractions pertinentes du domaine du problème, retenues en fonction des objectifs qu'on se donne et de l'application envisagée pour l'ontologie, par exemple, la description d'un ensemble d'objets, d'une tâche, d'une fonction, d'une stratégie, d'un processus de raisonnement, etc. [5]

b. Les relations :

Traduisent les interactions existant entre les concepts présents dans le domaine ciblé. Ces relations sont formellement définies comme tout sous ensemble d'un produit cartésien de n ensembles, c'est à dire $R : C_1 \times C_2 \times \dots \times C_n$

- la relation de spécialisation (subsomption),
- la relation de composition (méronymie),
- la relation d'instanciation, etc. Ces relations nous permettent de capturer, la structuration ainsi que l'interaction entre les concepts, ce qui permet de représenter une grande partie de la sémantique de l'ontologie. [5]

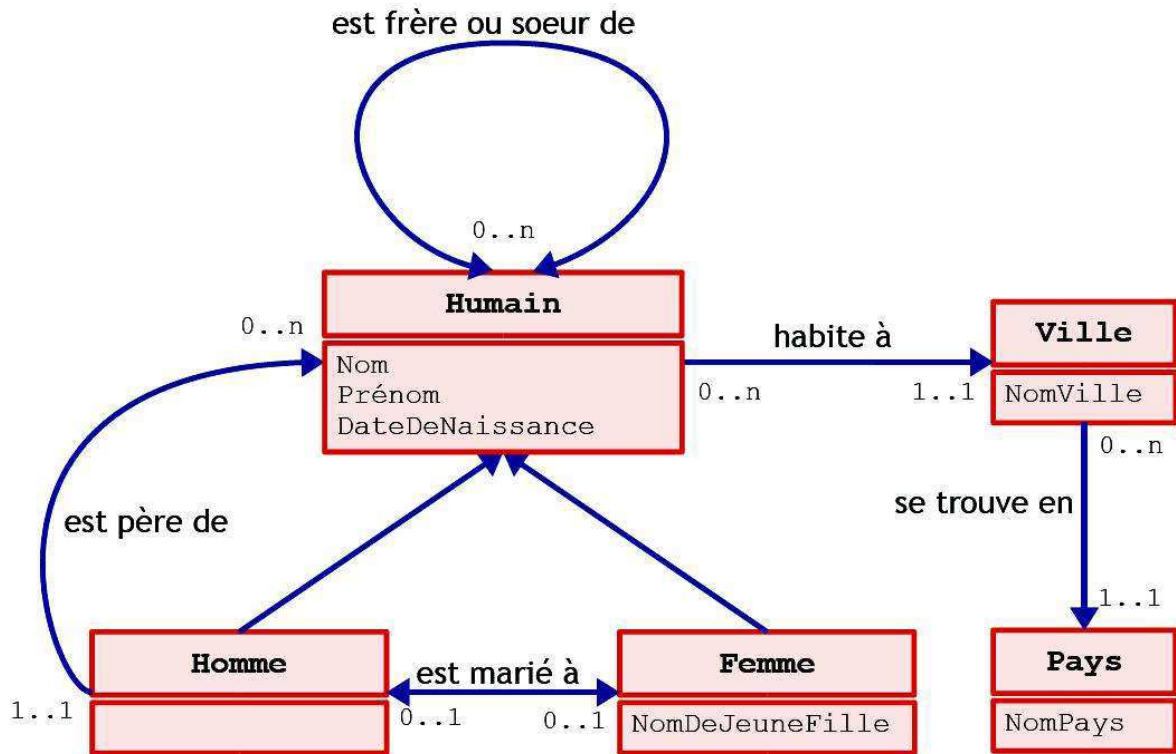


Figure I. 2: Exemple des relations et concepts [7]

c. Les fonctions :

Sont des cas particuliers de relations dans lesquelles le nième élément (extrant) de la relation est défini de manière unique à partir des n-1 éléments précédents (intrants). Formellement, les fonctions sont définies ainsi : $F : C_1 \times C_2 \dots \times C_{n-1} \rightarrow C_n$. Comme exemple de fonctions binaires, nous pouvons citer la fonction « mère-de » ; tandis que, « *prix-de voiture-usagée* » qui calcule le prix d'une voiture de seconde main en fonction du modèle de voiture, de la date de fabrication et du nombre de kilomètres est une fonction ternaire.

d. Les axiomes :

Permettent de modéliser des assertions toujours vraies, à propos des abstractions du domaine traduites par l'ontologie. Ils permettent de combiner des concepts, des relations et des fonctions pour définir des règles d'inférences et qui peuvent intervenir, par exemple, dans la déduction, la définition des concepts et des relations, ou alors pour restreindre les valeurs des propriétés ou les arguments d'une relation.

e. Les entités :

Où individus constituent la définition extensionnelle de l'ontologie. Ils représentent des éléments singuliers véhiculant les connaissances (statiques, factuelles) à propos du domaine du problème.

V. Les étapes de construction d'ontologies :**V.1 Objectif et critères d'évaluation d'une ontologie :**

L'objectif premier d'une ontologie est de modéliser un ensemble de connaissances dans un domaine donné, qui peut être réel ou imaginaire.

Les ontologies sont employées dans l'intelligence artificielle, le Web sémantique, le génie logiciel, l'informatique biomédicale ou encore l'architecture de l'information comme une forme de représentation de la connaissance au sujet d'un monde ou d'une certaine partie de ce monde.

D'après Gruber, cinq critères permettent de mettre en évidence des aspects importants d'une ontologie :

- **La clarté :**

La définition d'un concept doit faire passer le sens voulu du terme, de manière aussi objective que possible (indépendante du contexte). Une définition doit de plus être complète (c'est-à-dire définie par des conditions à la fois nécessaires et suffisantes) et documentée en langage naturel.

- **La cohérence :**

Rien qui ne puisse être inféré de l'ontologie ne doit entrer en contradiction avec les définitions des concepts (y compris celles qui sont exprimées en langage naturel).

- **L'extensibilité :**

Les extensions qui pourront être ajoutées à l'ontologie doivent être anticipées. Il doit être possible d'ajouter de nouveaux concepts sans avoir à toucher aux fondations de l'ontologie.

- **une déformation d'encodage minimale :**

Une déformation d'encodage a lieu lorsque la spécification influe la conceptualisation (un concept donné peut être plus simple à définir d'une certaine façon pour un langage d'ontologie donné, bien que cette définition ne corresponde pas exactement au sens initial). Ces déformations doivent être évitées autant que possible.

- **un engagement ontologique minimal :**

Le but d'une ontologie est de définir un vocabulaire pour décrire un domaine, si possible de manière complète ; ni plus, ni moins. Contrairement aux bases de connaissances par exemple, on n'attend pas d'une ontologie qu'elle soit en mesure de fournir systématiquement une réponse à une question arbitraire sur le domaine. Toujours selon Gruber, l'engagement ontologique peut être minimisé en spécifiant la théorie la plus faible (celle permettant le plus de modèles) couvrant un domaine ; elle ne définit que les termes nécessaires pour partager les connaissances consistantes avec cette théorie.[6]

Exemple d'ontologie:

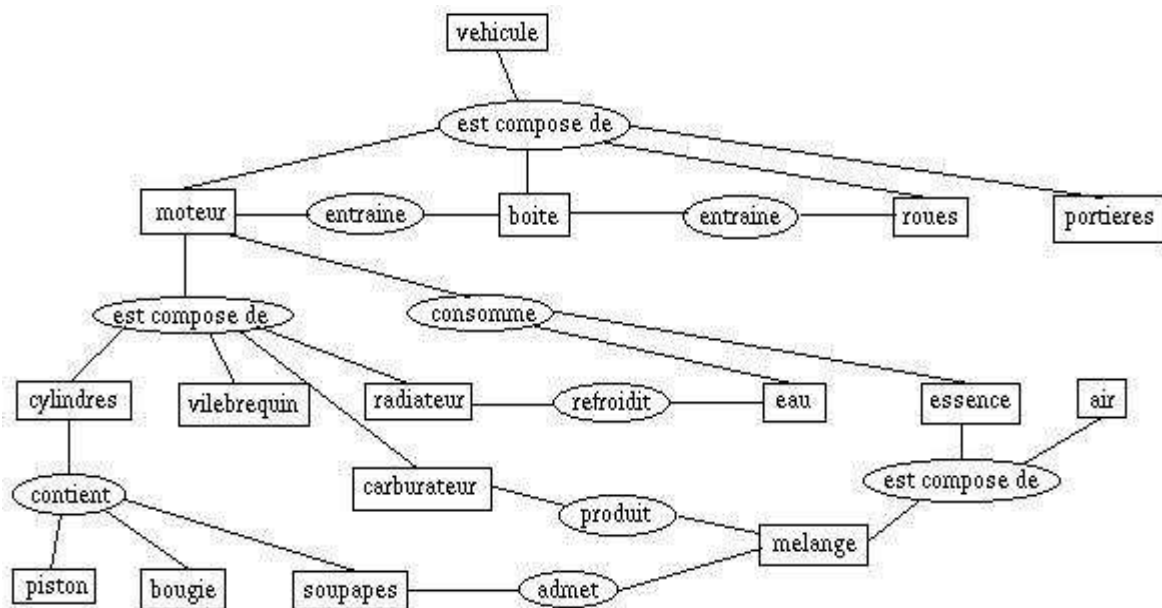


Figure I.3 : Ontologie d'une composition d'une voiture

VI. Avantages d'une ontologie :

Parmi les avantages de la méthodes des ontologies la compréhension commune de la structure de l'information entre les personnes ou les fabricants de logiciels, assurer l'interopérabilité entre systèmes, permettre l'échange de connaissances entre Systèmes, Permettre la réutilisation du savoir sur un domaine: créer et conserver des bases de connaissances réutilisables, Expliciter ce qui est considéré comme implicite sur un domaine, Distinguer le savoir sur un domaine du savoir opérationnel et Analyser le savoir sur un domaine. [7]

VII. Différentes sortes d'ontologies :

Cette section présente les types les plus généralement utilisés d'ontologies. On peut avoir une idée de la connaissance qui est incluse dans chaque type d'ontologie. Les ontologies peuvent être classifiées selon plusieurs dimensions. Parmi celles-ci, nous en examinerons deux.

VII.1 Objet de conceptualisation :

Les ontologies classifiées selon leur objet de conceptualisation c'est –à-dire selon le but leur utilisateur [8].

a) Ontologie de haut niveau :

Décrit des concepts très généraux comme l'espace, le temps, la matière, les objets, les événements, les actions, etc. Ces concepts ne dépendent pas d'un problème ou d'un domaine particulier, et doivent être, du moins en théorie, consensuels à de grandes communautés d'utilisateurs [9]. Des exemples d'ontologies de haut niveau sont Dolce ou Sumo.

b) Ontologie de domaine :

Contrairement aux ontologies de haut niveau, les ontologies de domaine sont plus spécifiques. Elles synthétisent les connaissances spécifiques à un domaine particulier. Elles décrivent le vocabulaire ayant trait à un domaine générique (ex. : l'enseignement, la médecine...), notamment en spécialisant les concepts d'une ontologie de haut niveau [9].

c) Ontologie de tâches :

Ce type d'ontologies est utilisé pour conceptualiser des tâches spécifiques dans les systèmes, telles que les tâches de diagnostic, de planification, de conception, de configuration, de

tutorat. Soit tout ce qui concerne la résolution de problèmes. Ce type d'ontologies décrit le vocabulaire concernant une tâche générique (ex. : enseigner, diagnostiquer...), notamment en spécialisant les concepts d'une ontologie de haut niveau. Certains auteurs emploient le nom « ontologie du domaine de la tâche » pour faire référence à ce type d'ontologie [10].

d) Ontologie d'application:

Cette ontologie est la plus spécifique, elle contient des concepts dépendants d'un domaine et d'une tâche particuliers, qui sont généralement subsumés par des concepts de ces deux ontologies. Ces concepts correspondent souvent aux rôles joués, Ontologie de haut niveau, Ontologie d'application, Ontologie de Tâche, Ontologie de Domaine L'INGENIERIE ONTOLOGIQUE par les entités du domaine lors de l'exécution d'une certaine activité [9]. Il s'agit donc ici de mettre en relation les concepts d'un domaine et les concepts liés à une tâche particulière, de manière à en décrire l'exécution (ex. : apprendre les statistiques, effectuer des recherches dans le domaine de l'astronomie, etc.).

VII.2 Niveau de formalisme de représentation :

D'autre part, selon le niveau du formalisme de représentation du langage utilisé pour décrire l'ontologie, [11] proposent une classification comprenant quatre catégories:

- **Informelles :**

Ontologies opérationnelles dans un langage naturel (sémantique ouverte).

- **Semi-informelles :**

Utilisation d'un langage naturel structuré et limité.

- **Semi-formelles :**

Langage artificiel défini formellement.

- **Formelles :**

Utilisation d'un langage artificiel contenant une sémantique formelle, ainsi que des théorèmes et des preuves de propriétés telles la robustesse et l'exhaustivité [8].

VIII. Outils de manipulation des ontologies

Il existe plusieurs outils de manipulation des ontologies :

- Protégé est le plus connu et le plus utilisé des éditeurs d'ontologie. Open-source, développé par l'université Stanford, il a évolué depuis ses premières versions (Protégé-2000) pour intégrer à partir de 2003 les standards du Web sémantique et notamment OWL. Il offre de nombreux composants optionnels : raisonneurs, interfaces graphiques.
- SWOOP est un éditeur d'ontologie développé par l'Université du Maryland dans le cadre du projet MINDSWAP. Contrairement à Protégé, il a été développé de façon native sur les standards RDF et OWL, qu'il prend en charge dans leurs différentes syntaxes (pas seulement XML). C'est une application plus légère que Protégé, moins évoluée en termes d'interface, mais qui intègre aussi des outils de raisonnement.
- KMgen est un éditeur d'ontologie pour le langage KM (KM: The Knowledge Machine).
- Avec l'émergence du marché des technologies du Web sémantique, on peut noter l'apparition depuis 2005 d'outils logiciels proposés par des éditeurs commerciaux. On peut citer :
 - SemanticWorks qui fait partie de la suite d'outils XML développée par Altova, supportant le langage OWL à travers sa syntaxe XML,
 - TopBraid Composer développé par TopQuadrant, son interface et ses fonctionnalités ressemblant beaucoup à celles de Protégé (le développeur principal de TopBraid étant l'ancien développeur des extensions OWL de Protégé),
 - OntologyCraftWorkbench développé par l'équipe Condillac "Ingénierie des Connaissances" de l'Université de Savoie, les ontologies étant disponibles aux formats XML et OWL. OCW et utilisées par la société Ontologia,
 - KAD-Office développé par Iknova Consulting et spécialisé sur la représentation graphique des connaissances industrielles d'ingénierie à l'aide de système d'exploitation. [12]

Il existe d'autre part des outils informatiques permettant de construire une ontologie à partir d'un corpus de textes. Ces outils parcourent le texte à la recherche de termes récurrents ou définis par l'utilisateur, puis analysent la manière dont ces termes sont mis en relation dans le texte (par la grammaire, et par les concepts qu'ils recouvrent et dont une définition peut être trouvée dans un lexique fourni par l'utilisateur). Le résultat est une ontologie qui représente la connaissance globale que contient le corpus de texte sur le domaine d'application qu'il couvre.

Conclusion :

Au long de ce chapitre, nous avons essayé d'éclaircir la notion d'ontologie en présentant certaines définitions. Nous avons montré aussi leurs avantages, leurs domaines d'application et leurs principaux types.

Chapitre 2

Les langages des ontologies.

I. Langages pour les ontologies

La plupart de ces langages se basent ou sont proches sur la logique du premier ordre, et représentent donc les connaissances sous forme d'assertion (sujet, prédicat, objet). Ces langages sont typiquement conçus pour s'abstraire des structures de données et se concentrer sur la sémantique. Parmi les formalismes les plus employés se basant sur la logique des prédicats, on retrouve des langages comme N3 ou N-Triple. On peut aussi évoquer le langage DEF.

Par ailleurs, dans le cadre de ses travaux sur le Web sémantique, le W3C a mis en place en 2002 un groupe de travail dédié au développement de langages standards pour modéliser des ontologies utilisables et échangeables sur le Web. S'inspirant de langages précédents comme DAML+OIL et des fondements théoriques des logiques de description, ce groupe a publié en 2004 une recommandation définissant le langage OWL (Web Ontology Language), fondé sur le standard RDF et en spécifiant une syntaxe XML. Plus expressif que son prédécesseur RDFS, OWL a rapidement pris une place prépondérante dans le paysage des ontologies et est désormais, de facto, le standard le plus utilisé. [13]

II. Les standards : XML RDF, et RDFS

II.1 XML:

Le XML (eXtensible Markup Language) est une syntaxe de description, langage informatique conçu pour structurer et partager les données. Le XML répond à certains principes rigoureux dont la validation nécessite le plus souvent une DTD ou autre schéma .syntaxique, mais également à une logique informationnelle, une rigueur dans la formulation (ce qui lui donne ce degré sémantique).[14]

II.2 RDF:

RDF, abréviation de *Ressource description Framework*, créé en 1999, il est conçu comme un modèle de représentation des connaissances. Il est à la base de la logique de représentation et de structuration des savoirs. Il permet d'écrire des métadonnées (Métadonnée : information permettant d'en décrire une autre), rendre l'information traitable par la machine, favoriser l'interopérabilité des savoirs (la combinaison de plusieurs savoirs).[15]

Ces derniers s'organisent autour de plusieurs graphes, collections de triplets dans lesquels chaque élément de connaissance se présente sous la forme : sujet, prédicat, objet.

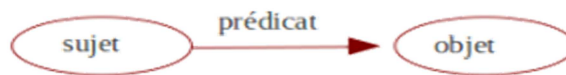


Figure II.1 Schématique courante du triplet RDF [16]

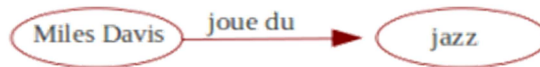


Figure II.2 Exemple de triplet RDF[16]

Un ensemble de triplets est nommé un graphe RDF :

- Un diagramme composé de nœuds
- Des liens entre deux nœuds.

Le modèle RDF définit trois types d'objets:

- **Les ressources :**

Les ressources sont tous les objets décrits par RDF. Généralement, ces ressources peuvent être aussi bien des pages Web que tout objet ou personne du monde réel. Les ressources sont alors identifiées par leur URI (Uniform Resource Identifier).

- Les propriétés :

Une propriété est un attribut, un aspect, une caractéristique qui s'applique à une ressource. Il peut également s'agir d'une mise en relation avec une autre ressource.

- Les valeurs:

Les valeurs en question sont les valeurs particulières que prennent les propriétés. Ces trois types d'objets peuvent être mis en relation par des assertions, c'est à dire des triplets.

II.3 RDF Schéma:

RDF Schéma est le vocabulaire de structuration proposé pour la description du savoir. Répandu aussi sous l'acronyme RDFS, il répond aux principes du RDF (collection de triplets organisés en graphes) et propose des noms de balises et attributs XML standardisés (tant dans leur forme que dans leur organisation) pour l'échange et le partage des données décrivant les savoirs.[17]

Le RDF Schéma permet de :

- Définir un « schéma » de méta donné
- Qualifier les relations décrites en RDF :
 - Type des ressources manipulées (littéral, etc.)
 - Hiérarchie de ces ressources
 - Quelles propriétés s'appliquent à une ressource
 - Domaine de valeur de ces propriétés

Exemple: sur RDF et RDFS

En clair

- Un chien est un animal.
- le chat1 est un chat.
- Les chats sont des animaux.
- les zoos hébergent uniquement des animaux.
- Le zoo1 héberge le chat2.

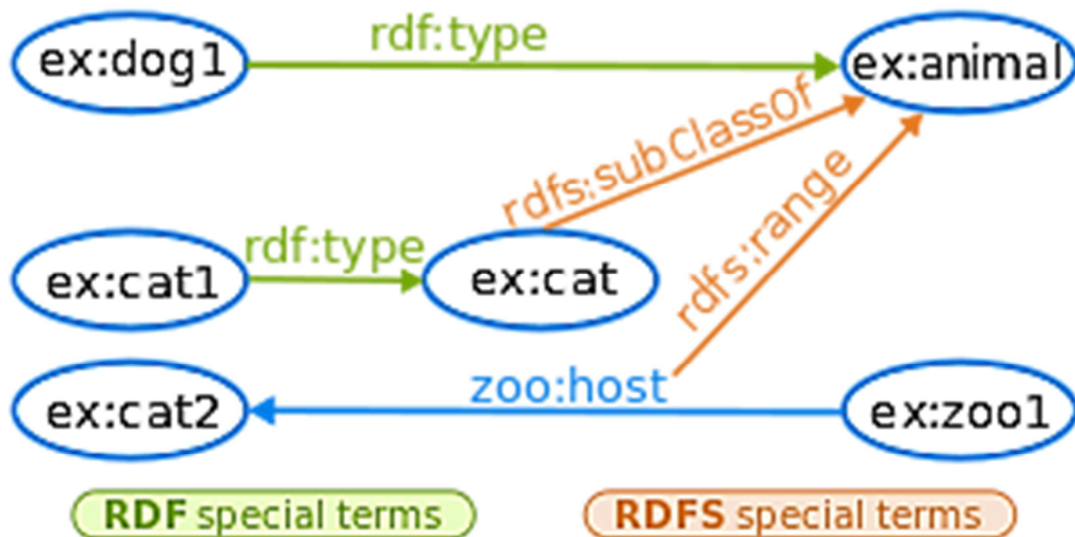


Figure II.3 Le graphe RDF et RDFS

En RDF/turtle[18]

```

ex: dog1 rdf: type ex: animal.
ex: cat1 rdf: type ex: cat.
ex: cat rdfs: sub Class Of ex: animal.
zoo: host rdfs: range ex: animal.
ex:zoo1 zoo: host ex: cat2.

```

III OWL:

OWL est un langage d'ontologies Web. Les premiers langages utilisés pour le développement d'outils et d'ontologies pour des communautés d'utilisateurs spécifiques n'ont pas été définis pour être compatibles avec l'architecture du Web en général, et du Web Sémantique en particulier. OWL répare ce manque en utilisant à la fois les URIs pour nommer, et la fonctionnalité fournie par RDF pour créer des liens. Ainsi, les ontologies Web possèdent les avantages suivants: [19]

- Capacité d'être distribué au travers de nombreux systèmes
- Mise à échelle pour les besoins du Web
- Compatible avec les standards Web pour l'accessibilité et l'internationalisation
- Ouvert et extensible

III.1 Une extension de RDF et RDFS

En pratique, le langage OWL est conçu comme une extension de Resource Description Framework (RDF) et RDF Schéma (RDFS) ; OWL est destiné à la description de classes (par des constructeurs) et de types de propriétés. De ce fait, il est plus expressif que RDF et RDFS, auxquels certains reprochent une insuffisance d'expressivité due à la seule définition des relations entre objets par des assertions. OWL apporte aussi une meilleure intégration, une évolution, un partage et une inférence plus facile des ontologies.

Aux concepts de classe, de ressource, de littéral et de propriétés des sous-classes, de sous-propriétés, de champs de valeurs et de domaines d'application déjà présents dans RDFS, OWL ajoute les concepts de classes équivalentes, de propriétés équivalentes, d'égalité de deux ressources, de leurs différences, du contraire, de symétrie et de cardinalité Par exemple :

- la manière de limiter les propriétés de la classe au regard du nombre et du type.
- les moyens d'induire que ces termes assortis de propriétés diverses soient des membres d'une classe particulière ou non.
- les moyens de déterminer si tous les membres d'une classe auront une propriété particulière, ou seulement quelques-uns d'entre eux.
- les moyens de séparer des relations de types un-à-un de relations de type plusieurs-à-un ou un-à-plusieurs, permettant ainsi de représenter des "clés étrangères" d'une base de données dans une ontologie.
- les moyens d'exprimer des relations entre des classes définies dans différent documents sur le Web.
- les moyens de construire de nouvelles classes en dehors de toutes unions, intersections et compléments avec d'autres classes.
- les moyens de contraindre un domaine à des combinaisons classe/propriété spécifiques.[20]

III.2 Les niveaux d'OWL

OWL permet, grâce à sa sémantique formelle basée sur une fondation logique largement étudiée, de définir des associations plus complexes des ressources ainsi que les propriétés de leurs classes respectives. OWL définit trois sous-langages, du moins expressif au plus expressif : OWL-Lite, OWL-DL et OWL-Full. Des algorithmes décidables existent pour la totalité de OWL-Lite. Quoique les problèmes d'inférence de OWL-DL puissent être résolus en temps exponentiel de

façon générale, le comportement est souvent satisfaisant. Il n'existe aucun algorithme d'inférence décidable pour OWL-Full.

OWL-Lite : est la version la plus simple du langage OWL. Il correspond à la logique de descriptions.

OWL-DL : (sigle pour Ontology Web Language Description Logics)¹ est une version décidable du langage informatique OWL. Ce langage correspond à la logique de descriptions.

OWL-Full est une version indécidable du langage informatique OWL.[21]

III.3 Spécifications OWL existants

Le Groupe de Travail Web Ontology du W3C a produit six documents OWL. Chacun d'entre eux adresse un segment précis relatif à la spécification, ce qui permet un apprentissage, une compréhension et une utilisation plus simples du langage OWL. Les six documents sont les suivants : une représentation des cas d'utilisation et conditions préalables qui ont motivé les travaux sur OWL - un document schématique qui explique brièvement les caractéristiques d'OWL et comment elles peuvent être utilisées.

- Un guide compréhensif illustré d'exemples d'utilisation des caractéristiques d'OWL
- Un document de référence qui fournit les détails de chaque caractéristique d'OWL
- Des exemples de tests et une collection de tests, fournissant plus d'une centaine de tests permettant de s'assurer que les implémentations OWL sont cohérentes avec la conception du langage
- Un document présentant la sémantique d'OWL et les détails du mapping d'OWL à RDF.[22]

Conclusion :

Dans ce chapitre nous avons découvert les outils nécessaires de développement des ontologies à savoir les langages de représentation.

Chapitre 3

Conception et implémentation

I. Introduction

Le travail réalisé dans le cadre de notre projet vise à développer une application qui sert à renforcer la collaboration entre les différents services de notre établissement E.P.H. « Ghazaouet », spécialement avec le service de la pharmacie centrale en faisant implémenter une ontologie qui permet aux différents services de partager un vocabulaire commun.

II. Conception et implémentation des ontologies :

Dans cette partie, nous allons présenter notre application d'un point de vue pratique.

II.1 Présentation de l'ontologie :

Notre but d'application est de mettre à la disposition des utilisateurs une ontologie qui répond au besoin du domaine pharmaceutique. Comme il est présenté sur la **figure III.1**, l'ontologie est conçue pour présenter les informations et l'hierarchique (classe et sous classe), ainsi que les relations entre propriétés.

Notre ontologie contient trois classes principales :

La Classe « thérapeutique » : c'est la classe globale, qui englobe les différents « sous-classe -thérapeutique »,

La classe « Sous-classe-thérapeutique » comprenant la classe « Médicament ».

Après la création des classes nous avons conçu les relations suivantes :

- La relation : « Appartient à » :

Le domaine est la classe médicament et le range, la classe thérapeutique.

- La relation : « Forme » :

Le domaine est la classe médicament, le range est de type string

- La relation : « Date-premp » :

Le domaine est la classe médicament et le range est de type date

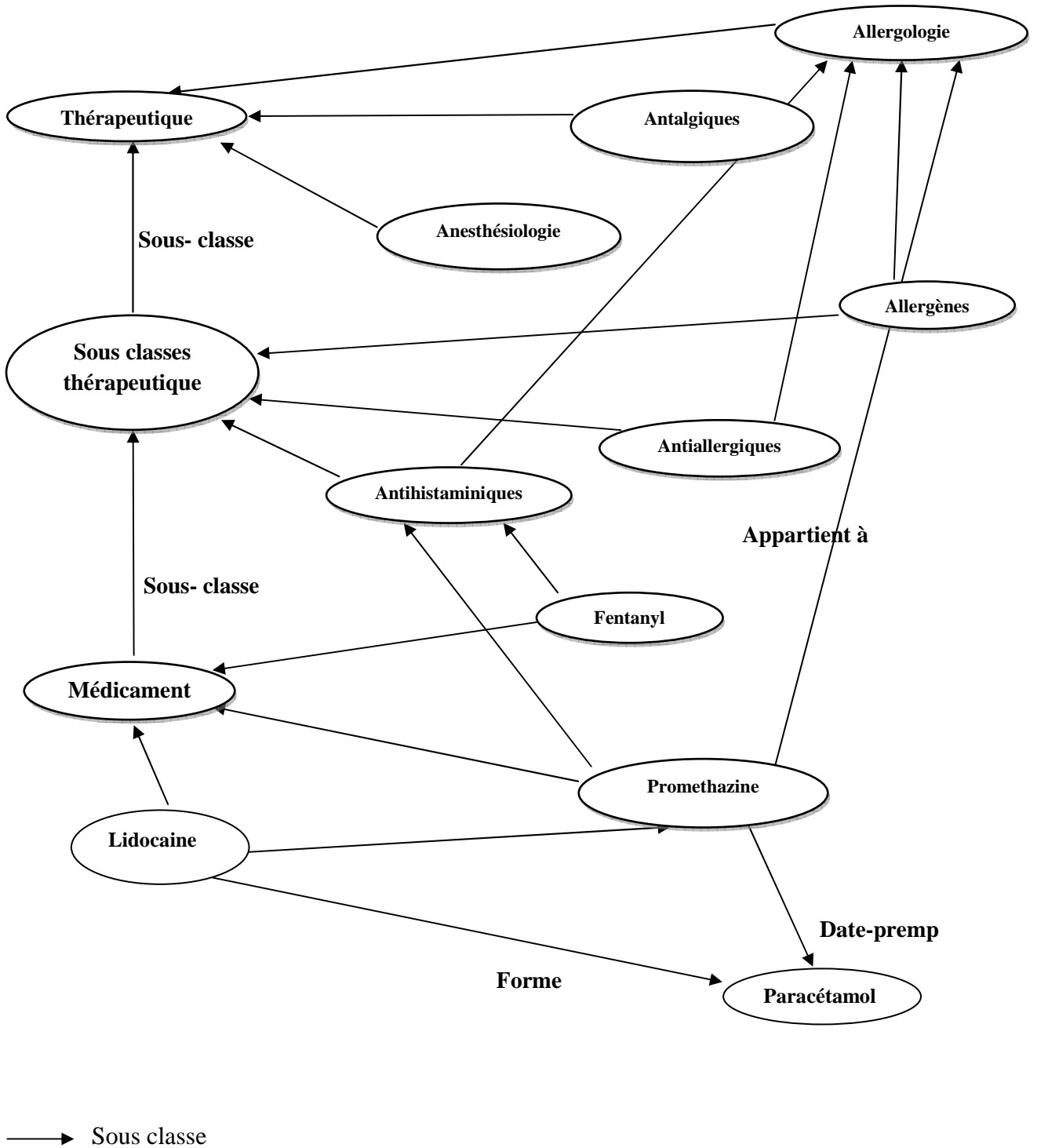


Figure II.1 Une partie d'ontologie

II.2 Conception de l'application :

La **figure II.2** présente le diagramme de cas d'utilisation de notre application.

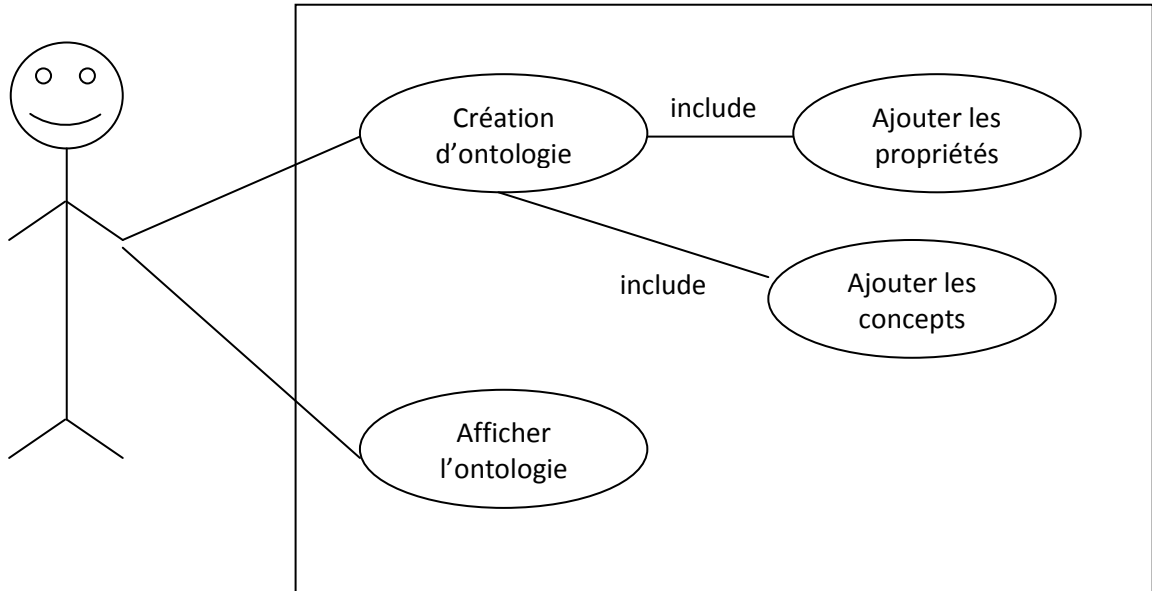


Figure II.2 Diagramme de cas d'utilisation.

La figure suivante montre le diagramme des classes de cette application.

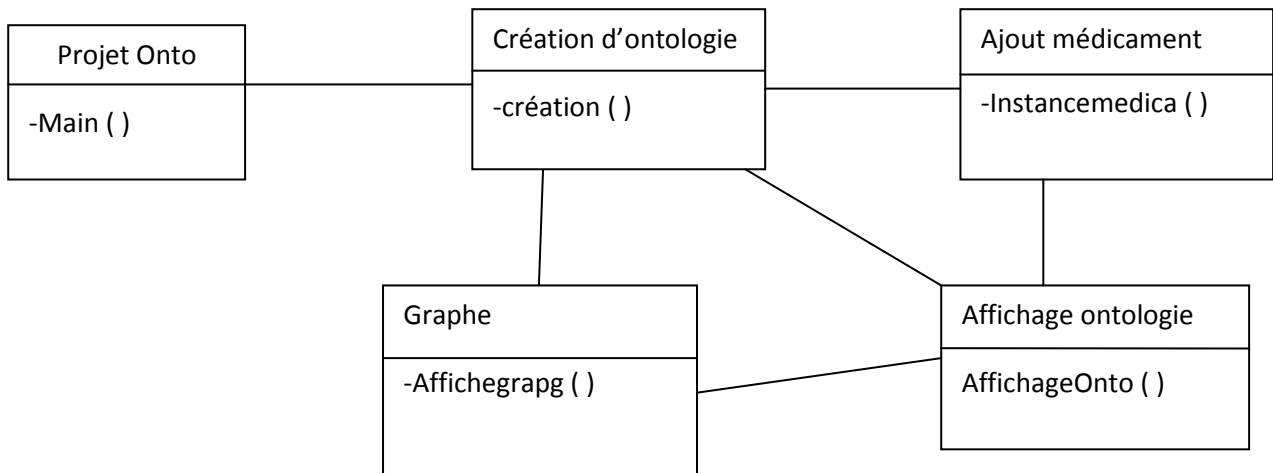


Figure II.3 Diagramme de classe.

II.3 Outils d'implantation:

Avant de commencer l'implémentation, nous allons tout d'abord spécifier les outils utilisés dans notre implémentation.

II.4.1 le langage JAVA :

Le langage Java est un langage de programmation informatique orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy .Il existe plusieurs IDE (Integrated Development Environment) pour le langage JAVA par exemple Eclips, JBuilder et NetBeans que nous avons utilisé.

II.4.2 NetBeans

NetBeans est un environnement de développement intégré (EDI), placé en open source en juin 2000 sous licence CDDL (Common Développement and Distribution License). En plus de Java, NetBeans permet également de supporter différents autres langages, comme C, C++, JavaScript, XML, et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, éditeur graphique d'interfaces et de pages Web).

Conçu en Java, NetBeans est disponible sous Windows, Linux, ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java).

NetBeans constitue par ailleurs une plate-forme qui permet le développement d'applications spécifiques (bibliothèque Swing (Java)). L'IDE NetBeans s'appuie sur cette plate-forme, il s'enrichit à l'aide de plugins.

II.4.3 Jena :

Jena est un API java open source développé par le laboratoire de Hewlett-Packard permettant la lecture et la manipulation des ontologies décrites en RDFS ou en OWL.

II.4 Interfaces de l'application :

Dans cette partie, nous allons présenter les différentes formes de notre application.

L'écran ci-dessous présent l'interface principale de notre application qui fournit les fonctionnalités de la création et l'affichage de l'ontologie.

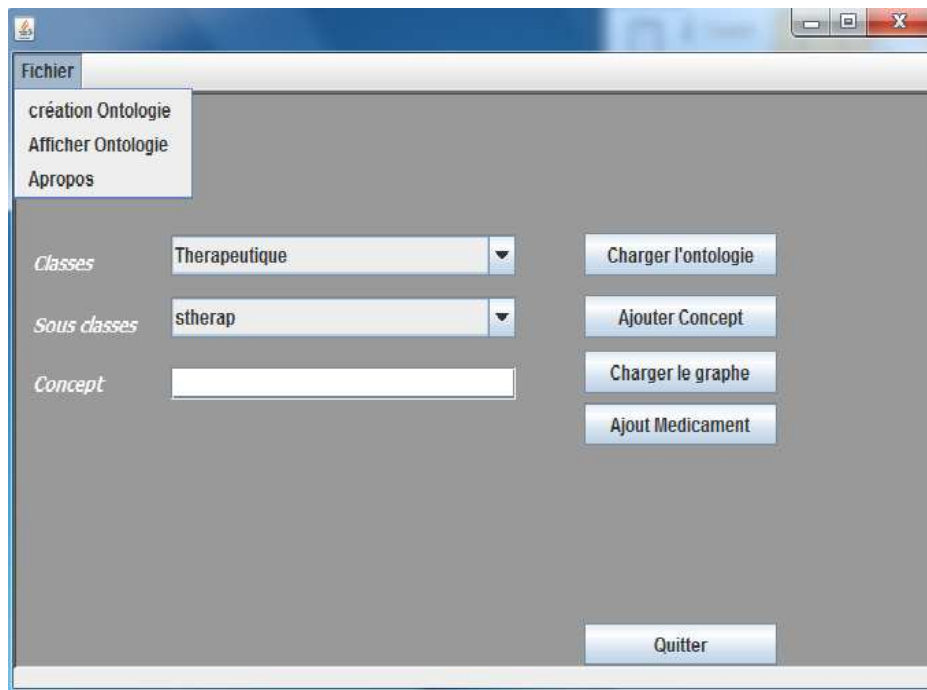


Figure II.3: Interface principale.

L'écran suivant (Figure III.3) permet d'afficher l'ontologie sous forme OWL.

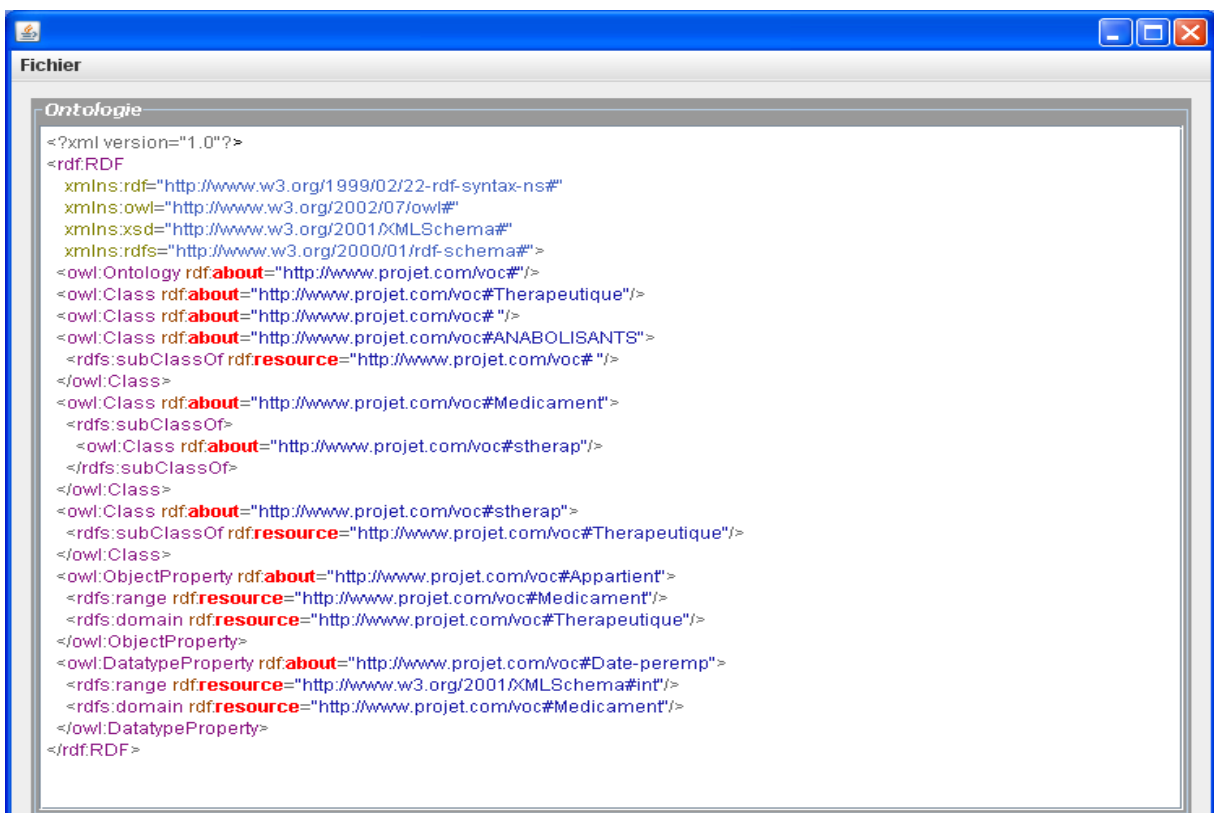


Figure III.3: l'Affichage de l'ontologie en OWL.

Le graphe si dessous représente l'ontologie ouvert. Dans notre cas nous avons choisi d'ouvrir une ontologie qui contient des classes thérapeutiques et sous classes thérapeutiques.

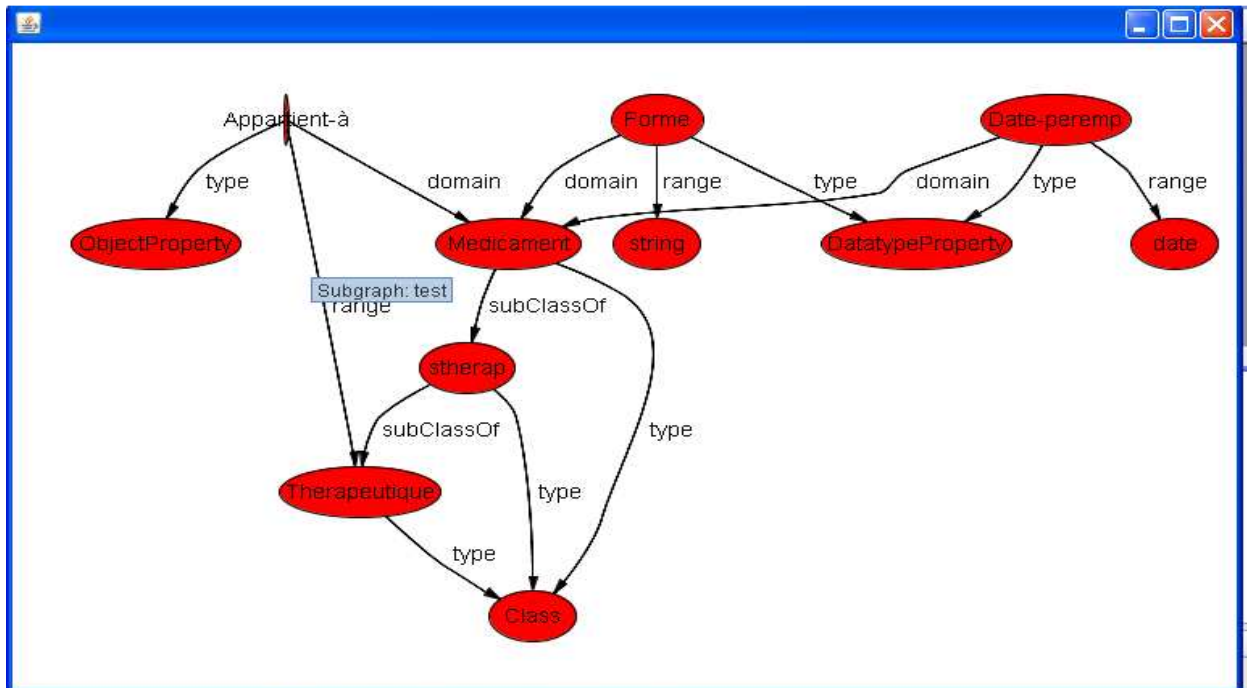


Figure III.4: le graphe d'affichage d'ontologie

L'écran suivant (Figure III.5) représente l'ajout de médicament

The screenshot shows a window titled "Ajout d'un médicament" with the following components:

- Classe:** A dropdown menu.
- Sous classe:** A dropdown menu.
- Nom Medicament:** A text input field.
- Date:** A text input field.
- Forme:** A text input field.
- Buttons:** "Load Ontologie", "Ajouter médicament", "Afficher graphe", and "Quitter".

Figure III.5: Ajout d'un médicament

Conclusion

Dans ce chapitre, nous avons présenté les aspects techniques utilisés dans notre travail,

Nous avons commencé par présenter les aspects d'implémentation qui sont utilisés dans notre travail. L'implantation repose essentiellement sur le langage JAVA avec NetBeans comme environnement de développement et le langage OWL pour la représentation des ontologies.

Conclusion générale

Le travail présenté dans ce mémoire rentre dans le cadre du développement et la création d'une ontologie pharmaceutique qui regroupe de nombreuses concepts ainsi que les liens entre ces concepts (classes et sous classes), des relations entre les concepts, des propriétés et des instances

Dans notre travail, nous avons développé un logiciel pour gérer notre ontologie, qui structure sous forme de hiérarchie les différentes relations, entre les classes et les sous classes, les médicaments, etc...., qui servira aux utilisateurs comme outil de recherche et de traitement de l'information.

Le choix du langage d'implémentation joue un rôle important dans la construction d'une application donnée. Une liste d'outils de construction d'ontologie est présentée.

L'état de l'art que nous avons présenté concerne les notions liées aux ontologies ainsi que les langages et techniques qui permettent de représenter les ontologies utilisés dans le cadre d'un system d'information. En utilisant le langage JAVA et la bibliothèque Jena, nous avons développé une application de gestion d'ontologie pharmaceutique.

Comme perspectives nous envisageons de pouvoir faire des requêtes sur cette ontologie, d'intégrer d'autre ontologies, aussi enrichir notre ontologie par d'autres concepts et de relations.

Référence Bibliographique

[1]: Henri Briand, Fabien Gandon et Fabien Picarougne. Atelier « Modélisation des connaissances », conférence Extraction et Gestion des connaissances, 8èmes Journées Francophones, Sophia Antipolis, 29 janvier 2008.

[2]: Elena Simperl, Reusing ontologies on the Semantic Web: A feasibility study, Data et Knowledge Engineering, Volume 68, Issue 10, Octobre 2009, Pages 905-925.

[3]: Guarino N. (1997b). Understanding, building and using ontologies. International J. Human-Computer Studies.

[4]: Thomas R. Gruber, Towards Principles for the Design of Ontologies Used for Knowledge Sharing in Formal Ontology in Conceptual Analysis and Knowledge Representation, Kluwer Academic Publishers, 1993,

[5] Djida BAHLOUL (Une approche hybride de gestion des connaissances basée sur les ontologies: *application aux incidents informatiques*) thèse de doctorat L'Institut National des Sciences Appliquées de Lyon.

[6]: Tom Gruber, article « Ontology » dans l'Encyclopedia of Data Base Systems, Springer-Verlag, 2009.

[7]: HDR Jean Charlet : <http://www-test.biomath.jussieu.fr/~jc/>.

[8]: A. G. Pérez, V. R. Benjamins. "Overview of Knowledge Sharing and Reuse Components : Ontologies and problem-Solving Methods".Proceeding the IJCAI-99 workshop on Ontologies and problem-Solving Methods (KRR5), Stockholm (Suède), Pp. 1.1-1.15, 1999.

[9]: N. Guarino. « Formal Ontology and Information Systems », Formal Ontology in Information Systems. IOS Press, 1998.

[10]: N. Hernandez. Ontologies de domaine pour la modélisation du contexte en recherche d'information . Thèse de doctorat, Université de Toulouse, 2005.

[11]: M. Uschold , M. Grüninger, "ONTOLOGIES: Principles, Methods and Applications".

Knowledge Engineering Review. 1996.

- [12]: Mathieu d'Aquin, Natalya F.Noy, Web Semantics : Sciences and Agents on the World Wide Web, Volume 11, March 2012, Pages 96-111.
- [13]: Raul Palma, Oscar Corcho, Asuncion Gomez-Pèrez, Peter Haase, A holistic approach to collaborative ontology development based on change management, WebSemantic : Science, Services and Agents on the World Wide Web, Volume 9, Issue 3, September 2011, Pages 299-314.
- [14]: WorldWideWeb: Proposal for a HyperText Project , T. Berners-Lee/CN, R.Cailliau/ECP, 12 novembre 1990
- [15] James Gillies, Robert Cailliau, How the Web was born: the story of the World Wide Web, Oxford, Oxford University Press, 2000,
- [16] Chris Welty, IBM Research, chris.welty@us.ibm.com, mardi 28 mai 2013.
- [17] Michael K.Smith, Electronic Data Systems, michael.smith@eds.com, mardi 28 mai 2013.
- [18] www.w3.org/1999/02/22-rdf-syntax-ns, mardi 28 mai 2013.
- [19] Frank Manola, fmanola@acm.org Eric Miller, W3C, em@w3.org, <http://www.w3.org/TR/2003/PR-rdf-primer-20031215/>
- [20] Peter F.Patel-Schneider, Patrick Hayes et Ian Horrocks, rédacteurs, Recommandation du W3C du 10 février 2004.
- [21] Marek Obitko, www.obitko.com/tutorials/ontologies-semantic-web-owl-dl-semantic
- [23] Jeremy J. Carroll, HP <jjc@hpl.hp.com> ; Jos De Roo, AGFA, <jos.deroo@agfa.com>