REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE ABOU BEKR BELKAID –TLEMCEN-
FACULTE DES SCIENCES
DEPARTEMENT D'INFORMATIQUE

MEMOIRE DE FIN D'ETUDE

POUR L'OBTENTION DU DIPLOME D'INGENIEUR D'ETAT
EN INFORMATIQUE
OPTION : SYSTEME D'INFORMATION AVANCE

THEME

Conception et implémentation d'une ontologie médicale dans un système question réponse

Présenté par :

- **LAÂMECHE Anouar**

Devant le jury :

- **Président : BENMAMMAR Badreddine**
- **Encadreur : ILES Nawel**
- **Examinateur : DIDI Fedoua**

Année Universitaire 2010 - 2011

Résumé

Le domaine médical dispose aujourd'hui d'un très grand volume de documents électroniques. Cependant, l'exploitation de cette grande quantité de données rend la recherche d'une information précise, complexe et coûteuse en termes de temps. Cette difficulté a motivé le développement de nouveaux outils de recherche adaptés, comme les systèmes de question-réponse. Ce type de système permet à un utilisateur de poser une question en langage naturel et de retourner une réponse précise à sa requête au lieu d'un ensemble de documents jugés pertinents, comme c'est le cas des moteurs de recherche.

Dans ce travail, je fais une étude théorique des systèmes de question réponse dans le domaine médical et à cause de l'utilité d'ontologie de structurer des bases de connaissances, j'implémente une ontologie dans le domaine médical par l'éditeur Protégé 4.1_rc4, le code retourné par cet éditeur je l'interroge par le langage SPARQL, pour cela j'utilise le langage java.

Mots clés : systèmes de question-réponse, moteurs de recherche, ontologie, bases de connaissances.

Abstract

The medical field now has a very large volume of electronic documents. However, the exploitation of this wealth of data makes the search for accurate, complex and costly in terms of time. This difficulty has motivated the development of new research tools adapted, such as question-answering systems. This type of system allows a user to ask a question in natural language and return a specific answer to the query instead of a set of documents deemed relevant, as is the case of search engines.

In this work, I am a theoretical question and answer systems in the medical field and because of the usefulness of ontology to structure knowledge bases, I implement an ontology in the medical field by the editor Protégé 4.1 _rc4, the code returned by this editor I question the SPARQL language, why I use the Java language.

Keywords: question answering systems, search engines, ontology, knowledge bases.

ملخص

الآن المجال الطبي لديه كميات كبيرة جدا من الوثائق الإلكترونية ومع ذلك، استغلال هذه الثروة من البيانات يجعل من عملية البحث عن دقيقة ومعقدة ومكلفة من حيث الوقت ، حفزت هذه الصعوبة في تطوير أدوات بحثية جديدة مكيفة، مثل نظم الإجابة على السؤال. هذا النوع من النظم يسمح للمستخدم أن يسأل سؤالا باللغة الطبيعية، والتحصل على إجابة محددة الاستعلام بدلا من مجموعة من الوثائق التي تعتبر ذات الصلة ، كما هو الحال في محركات البحث.

في هذا العمل، قمت بدراسة نظرية للنظم الإجابة على السؤال في المجال الطبي، ونظرا لفائدة الأنطولوجيا لهيكلة قواعد المعرفة ، قمت بتحضير انطولوجيا في المجال الطبي من قبل المحرر Protégé rc4_ 4.1 ، الرمز المحصل عليه من قبل هذا المحرر قمت بالعمل عليه باستخدام لغة SPARQL، و هذا بالاستعانة بلغة الجافا.

الكلمات الرئيسية : نظم الإجابة على السؤال، محركات البحث ، أنطولوجيا، وقواعد المعرفة.

Sommaire

Chapitre 3: Ontologies

Chapitre 4: Conception d'une ontologie médicale

Chapitre 5: Implémentation de l'ontologie médicale

Liste des figures

Liste des tableaux

Introduction générale

Introduction générale :

L'internet, a rendu l'accès à l'information plus aisée et rapide. De nos jours, rechercher une information ou un document sur le Web est devenu une activité quotidienne et prépondérante pour les internautes. Cette explosion du nombre de documents s'accompagne d'un accroissement du nombre d'utilisateurs interrogeant les différents moteurs de recherche tels que Google (http://www.google.com) et Yahoo! Search (http://www.yahoo.com).

Cependant, cette masse documentaire est devenue de plus en plus difficile à exploiter et à gérer. L'exploitation de cette grande quantité de données a rendu la recherche complexe et coûteuse en termes de temps. Désormais, l'utilisateur éprouve beaucoup de difficultés à trouver l'information correspondant à son besoin. Deux facteurs sont essentiellement responsables : le nombre de documents retournés par les moteurs de recherche d'une part ; l'hétérogénéité des informations disponibles sur le Web d'autre part. De plus, parmi tous les documents retournés par les moteurs, la plupart d'entre eux ne sont pas pertinents. De ce fait, un nouveau besoin a émergé : les futurs systèmes de recherche d'information doivent pouvoir répondre, en un minimum de temps, à des besoins plus précis que les systèmes actuels pour mieux satisfaire les utilisateurs.

Les systèmes de Question/Réponse sont une extension des systèmes de recherche documentaire allant dans ce sens. Ce type de système permet à un utilisateur de poser une question en langage naturel et de retourner une réponse à cette question au lieu d'un ensemble de documents jugés pertinents, comme c'est le cas des moteurs de recherche.

Répondre à des questions précises requiert une analyse plus en profondeur des documents sélectionnés afin d'en extraire l'information recherchée.

De ce fait, l'utilisation des systèmes question réponse est intéressant dans le domaine de recherche d'information.

L'objectif de ce travail est de comprendre le principe de fonctionnement d'un système de question-réponse dans le domaine médical, pour cela on entame trois chapitres théoriques, un chapitre de conception et un dernier chapitre de réalisation :

- Le premier chapitre est considéré un état de l'art des systèmes de question réponse.
- Le deuxième chapitre est considéré sur les ressources terminologiques et sémantiques du domaine médical.

- Le troisième chapitre concerne les ontologies pour modéliser les connaissances d'un domaine particulier.
- Dans le quatrième chapitre on exprime l'étude de cas d'une ontologie médicale.
- Dans le dernier chapitre on implémente l'ontologie modélisée.

Chapitre I

Etat de l'art des systèmes question réponse

I. Introduction [1]:

La quantité de documents électroniques mise à disposition, notamment grâce aux réseaux informatiques, a largement modifié la notion de recherche d'information. Les utilisateurs ont en effet un accès de plus en plus direct à l'information. Cependant, pour accéder plus facilement à une information pertinente, des systèmes de recherche d'information se révèlent incontournables. Bien que les moteurs de recherche constituent une solution efficace pour trouver des documents correspondant à une requête utilisateur, ils s'avèrent moins performants concernant la recherche d'une donnée précise. De ce fait, il est primordial de faire appel à des systèmes plus élaborés capables de retourner une information fiable à un besoin d'information précis. C'est l'ambition des systèmes de question-réponse.

II. Définition d'un système question réponse [1]:

Les systèmes de question-réponse peuvent se définir comme étant des systèmes de recherche d'information évolués qui permettent de retourner une réponse précise, ou un passage contenant la réponse, à une requête utilisateur, au contraire d'un moteur de recherche qui renvoie un ensemble de documents jugés pertinents. Ils offrent la possibilité aux utilisateurs de poser une question en langage naturel sans aucune restriction sur le vocabulaire. La question est analysée et traitée afin d'extraire automatiquement, à partir d'une base documentaire, une réponse directe à la question posée. Cette extraction, à la différence des moteurs de recherche, ne nécessite pas d'intervention manuelle.

La majorité des systèmes de question-réponse actuels affichent une certaine pertinence sur les questions factuelles, c'est-à-dire les questions dont la réponse attendue est une entité nommée. Toutefois, de nos jours, les systèmes ont tendance à se focaliser sur le traitement d'autres types de questions plus complexes, à savoir, les questions non factuelles, dont les réponses ne sont généralement pas aussi évidentes à trouver dans les corpus. Ce type de questions nécessite une analyse en profondeur de la question afin d'en extraire tous les éléments indispensables pouvant intervenir dans le processus de recherche. Pour ce faire, les systèmes de question-réponse utilisent différentes techniques pour améliorer l'analyse des questions comme les outils issus du traitement automatique des langues.

Pour cela il faut déterminer non seulement le type de la réponse recherchée, mais aussi les entités nommées présentes et l'objet sur lequel porte la question. Par ailleurs, pour

étendre leurs performances, les systèmes ont recourt à des ressources sémantiques, éventuellement extraites du Web. Cette utilisation de bases de connaissances existantes telles que le réseau lexico-sémantique de WordNet[1] ou encore les ontologies d'un domaine précis dans le cas d'un système de question-réponse en domaine restreint, permet aux systèmes d'augmenter la précision des réponses proposées.

III. Architecture d'un système question-réponse [1] :

Bien que les techniques diffèrent d'un système à l'autre, la plupart des systèmes de question réponse reposent sur une architecture classiquement fondée sur quatre modules complémentaires (voir **Figure 1.1**).

1. Le premier de ces quatre modules concerne l'analyse de la question. Il vise plus précisément à extraire d'une question les informations permettant de repérer la réponse dans les documents comme le type de la question posée, l'objet sur lequel porte cette question, appelé aussi «focus», le type de la réponse attendue et les mots importants de la question.
2. Le deuxième module a quant à lui pour objectif de sélectionner un ensemble de documents ou d'extraits de documents facilitant ainsi les traitements de la suite de la chaîne.
3. Le troisième module se charge d'analyser les documents sélectionnés et d'en extraire les passages candidats susceptibles de contenir la réponse.
4. Le quatrième et dernier module permet de rechercher dans les passages sélectionnés la réponse qui, selon la question et la particularité des systèmes, se présente sous la forme d'une entité nommée ou d'un passage contenant la réponse.

Ces quatre modules s'appuient principalement sur des techniques de traitement automatique de la langue et de recherche d'information. Les outils de recherche d'information servent plus particulièrement à la recherche des documents et des passages les plus pertinents, tandis que les techniques de traitement de la langue permettent d'améliorer les procédures d'extraction d'information en offrant la possibilité d'effectuer une analyse plus en profondeur de la question et des documents.

[1] http://wordnet.princeton.edu/

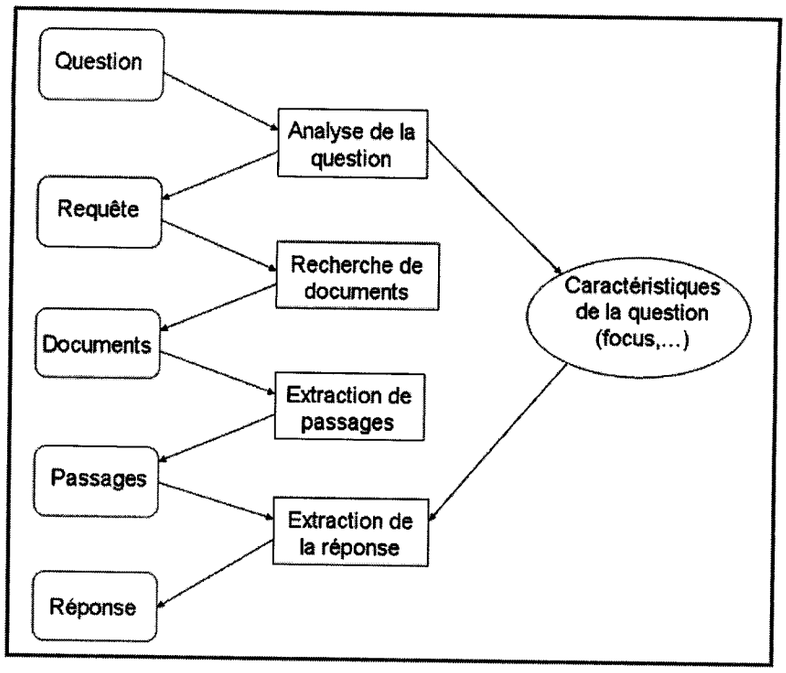


Figure 1.1: Architecture d'un système de question-réponse [1]

III.1 Analyse des questions :

L'analyse de la question est une étape importante dans la chaîne de traitement d'un système de question-réponse, outre le fait qu'elle est la première. En effet, il est primordial pour un système d'analyser une question aussi soigneusement que possible car cette analyse conditionne la stratégie de recherche à appliquer.

L'objectif principal de l'analyse de la question est à la fois de déterminer ce que le système doit chercher et de mettre en évidence les éléments informatifs permettant de sélectionner une réponse. Ainsi, l'analyse de la question doit déterminer :

- ***Le typage de la question:*** Il permet d'attribuer à la question une catégorie selon une classification prédéfinie (Définition, Factuelle, Booléenne). Par exemple la question suivante : « *Quelle est la définition du paludisme ?* » est une question définitoire, tandis que la question « *Citer sept pays membres de l'Union européenne ?* » se verra attribuer la catégorie factuelle de type liste ;
- ***Les entités nommées de la question :*** Il s'agit de repérer toutes les entités nommées présentes dans la question. Cela revient à repérer par exemple l'entité personne « *Pablo Picasso* » dans la question « *Dans quelle ville est né Pablo Picasso ?* » ;

- ***Le type de la réponse attendue :*** Ce type est généralement formalisé sous la forme d'un type d'entité nommée (personne, date, lieu) ou d'un type d'entité plus élargi (par exemple maladie, traitement). Ainsi, pour la question « *Qui a écrit Harry Potter ?* », le type de la réponse attendue est une entité nommée PERSONNE ; pour la question « Quel est le traitement de la cirrhose ? », le type attendu est l'entité TRAITEMENT. Ce type de questions est souvent plus facile à traiter que les questions portant sur des définitions ou des explications où le type sémantique de la réponse est plus complexe et moins facilement identifiable.
- ***Le focus de la question :*** Il s'agit d'extraire l'objet sur lequel porte la question, c'est-à-dire un élément susceptible d'être présent dans le passage réponse. Pour la question :

« En quelle année est né Alexandre Pouchkine ? », le focus est ainsi Alexandre Pouchkine.

Parallèlement, les mots-clés présents dans la question sont extraits pour composer une requête d'interrogation permettant à un système de recherche documentaire de retourner un ensemble de documents jugés pertinents. Ces mots sont considérés comme des éléments importants ayant un rapport direct avec la réponse permettant ainsi de restreindre le contexte de la question. Par exemple, pour la question :
« Combien d'oscars a reçu le film Titanic ? », les mots-clés à extraire sont : « oscars, film, Titanic » et la réponse à rechercher est une entité numérique de type quantité (en oscars).

III.2 Recherche des documents :

Dans un système de question-réponse, la recherche des documents se fait par l'interrogation d'un système de recherche d'information. Cette étape se révèle particulièrement capitale et complémentaire à l'analyse de la question pour la recherche de la bonne réponse car les systèmes de question-réponse ne peuvent trouver une réponse à une question que si elle est présente dans les documents sélectionnés. Cette tâche consiste donc à interroger un moteur de recherche classique pour récupérer une sélection de documents ou de passages restreints potentiellement porteurs de la réponse. Pour ce faire, les systèmes de question-réponse se reposent sur l'analyse de la question qui permet de générer une requête, souvent de nature booléenne, dédiée à l'interrogation d'une base textuelle.

Dans un contexte des systèmes de question-réponse en domaine restreint, la recherche documentaire se fait sur un ensemble généralement limité de documents alors que pour les systèmes en domaine ouvert, la recherche d'information s'effectue sur une grande collection de textes couvrant presque tous les domaines tels que les sources de données existantes sur le Web.

La requête d'interrogation est constituée principalement des termes importants de la question tels que les noms, verbes et adjectifs. Elle permet à la fois de restreindre le contexte de la recherche d'information et d'identifier les documents jugés pertinents par le moteur de recherche pour l'extraction de la réponse.

III.3 Analyse des documents candidats :

L'analyse des documents candidats a pour objectif principal de parcourir les documents sélectionnés pour rechercher les meilleurs passages de textes ou les phrases correspondant à la réponse recherchée en s'appuyant principalement sur les éléments issus de l'analyse de la question. La stratégie pour ce faire consiste le plus souvent à extraire des documents les passages ou les phrases comportant au moins un mot de la question ou une entité du même type sémantique que la réponse attendue. De même que pour la sélection des documents candidats, ces passages ou ces phrases sont hiérarchisés par ordre de pertinence. Leur choix est réalisé par des approches différentes spécifiques à chaque système :

- La méthode la plus utilisée consiste à repérer les mots de la question dans les documents pour n'extraire que les passages ou les phrases ayant le plus de mots en commun avec la question.
- Un certain nombre de systèmes adoptent une stratégie plus avancée fondée sur le calcul d'une mesure de proximité entre les mots de la question dans les passages, c'est-à-dire qu'ils font l'hypothèse que dans les documents censés contenir une réponse, les termes de la question et le type de la réponse attendue sont proches.
- D'autres approches, améliorant la performance des systèmes de question-réponse dans la sélection des passages pertinents ont été proposés et appliquées comme celle de (Gillard et ses collègues 2006) qui repose sur la densité des mots de la question dans les passages.

  Le calcul de cette densité est tout d'abord déterminé par l'extraction des objets de la question : les lemmes des mots, les types d'entités nommées présentes et le type de la réponse à rechercher. Ensuite, pour chaque élément, une distance

moyenne est calculée entre l'objet courant et les autres objets de la question. Cette distance est utilisée par la suite pour le calcul du score de densité afin d'identifier le passage le plus en relation avec la question, c'est-à-dire le passage censé contenir la réponse souhaitée.

Pour réduire la perte d'information, le passage candidat est composé d'un bloc de trois phrases regroupant la phrase réponse complétée par la phrase précédente et la phrase suivante.

III.4 Extraction des réponses :

Le module d'extraction de réponses constitue le dernier maillon de la chaîne de traitement d'un système de question-réponse. Cette fonction symbolise la différence majeure d'un tel type de systèmes par rapport aux systèmes de recherche d'information traditionnels.

Rechercher une réponse à une question revient à fouiller les passages candidats sélectionnés par l'analyse des documents choisis afin d'identifier et extraire le passage réponse correspondant à la question formulée. Cette notion de « passage réponse », qui caractérise la réponse supposée correcte retournée par le système, peut être présentée sous différentes formes suivant le système. Dans la majorité des systèmes de question-réponse, la réponse retournée est une liste de réponses organisée selon un indice de confiance ou bien leur fréquence d'apparition dans les documents candidats tandis que pour certains, la réponse retournée est une réponse unique courte ou un extrait d'un document contenant la bonne réponse avec son contexte.

La façon d'extraire les réponses est dépendante du type de la réponse attendue :

- Lorsqu'il s'agit d'une entité nommée, une approche commune est de repérer les entités correspondant au type sémantique de la réponse désirée dans les passages pertinents puis de les classer selon leur fréquence d'apparition. Cette fréquence est généralement calculée sur l'ensemble des documents renvoyés par le moteur de recherche, ou parfois pour certains systèmes, elle peut même être étendue sur une grande quantité de documents comme le Web pour profiter de la redondance de l'information.
- Dans le cas où la question n'attend pas une entité nommée en réponse, les systèmes font appel à des motifs d'extraction prédéfinis, appelés aussi patrons d'extraction. Ces patrons linguistiques exprimés sous la forme d'expressions régulières sont habituellement écrits manuellement mais sont parfois appris automatiquement a priori à partir de corpus de textes.

IV. Système de question réponse dans le domaine médical [2]:

IV.1 Principe de fonctionnement :

L'idée est d'associer des graphes sémantiques à la question d'une part et aux phrases du corpus d'autre part puis de rechercher les appariements pour trouver l'extrait de document qui répond à la question posée.

Le prototype que nous proposons, décrit dans la **figure 1.2**, suit une architecture de base semblable à celle des systèmes de question réponse généralistes (voir la **figure 1.1**). Comme point de départ, nous analysons le corpus médical pour extraire les index et construire les graphes sémantiques correspondant aux différentes unités textuelles des documents du corpus.

Nous nous intéressons ensuite à la question, que nous analysons pour déterminer les mots clés (termes médicaux et verbes de la question) et construire le graphe sémantique correspondant.

Le module « Recherche de documents » utilise ensuite les index des documents et les mots clés de la question pour extraire un premier sous-ensemble de documents avec des techniques de recherche d'information classiques. Cette première recherche réduit la taille des documents à fouiller sémantiquement. Enfin, le module « Extraction de la réponse » recherche des appariements entre le graphe sémantique de la question et les graphes correspondant aux documents extraits pour récupérer les unités textuelles pertinentes et donc extraire la ou les réponse(s).

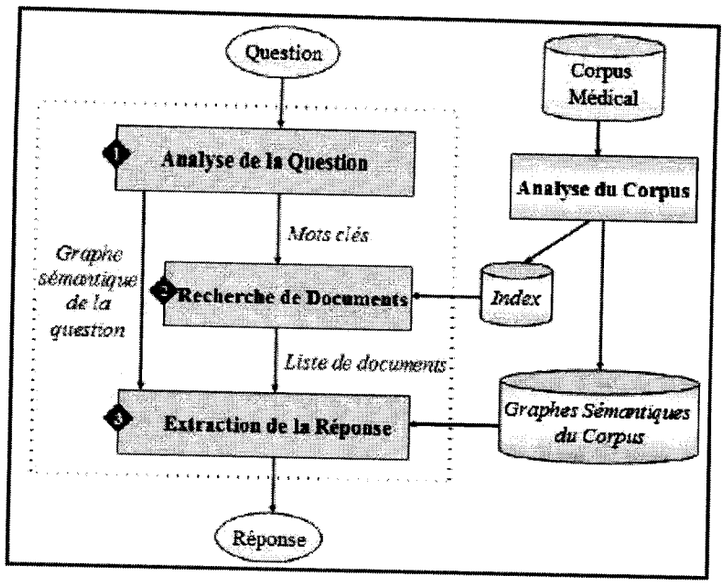


Figure 1.2 : Architecture d'un système de question-réponse pour un domaine médical [2]

IV.2 Méthode pour la construction automatique des graphes sémantiques : [2]

Pour construire automatiquement un graphe sémantique à partir d'une phrase, nous procédons en trois étapes : le repérage des entités nommées médicales présentes dans la phrase, l'étiquetage de ces entités nommées par leurs types et l'identification des relations sémantiques les reliant.

IV.2.a Repérage et étiquetages des entités nommées médicales :

Les entités nommées médicales correspondent aux instances des concepts génériques du domaine médical (par exemple maladie, médicament). Afin de faciliter leur reconnaissance, il est possible d'utiliser des ressources terminologiques spécialisées.

L'analyseur MetaMap permet la reconnaissance des entités nommées médicales. Il s'agit d'un outil qui permet de détecter le vocabulaire médical à partir de documents en anglais et de déterminer les concepts du méta thésaurus[2] de la ressource terminologique la plus développée en domaine médical UMLS correspondant aux termes repérés ainsi que leurs types sémantiques.

La **figure 1.3** présente un extrait du résultat de MetaMap associé à la question : « What is the drug of choice for enterococcus infections ? »

[2] Le méta thésaurus intègre une centaine de thésaurus et de terminologies biomédicaux

L'objet de la question est l'entité nommée 'enterococcus infections' qui désigne une maladie.

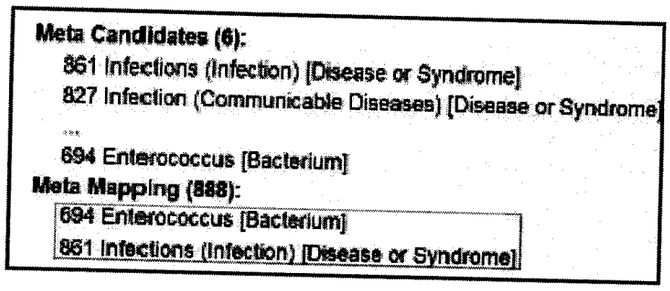


Figure 1.3 : Extraits de résultats de MetaMap [2]

Malgré ses performances, MetaMap ne fournit pas toujours les bonnes entités nommées médicales et les bons concepts.

IV.2.b Identification des relations sémantiques entre les termes médicaux :

Après avoir repéré les entités nommées médicales d'une phrase et les avoir étiquetées par leurs types, l'utilisation du réseau sémantique[3] de l'UMLS sert pour déterminer la ou les relation(s) sémantique(s) liant ces entités nommées. Si plusieurs relations existent entre deux entités nommées, il faut déterminer la bonne relation sémantique en se basant sur les informations syntaxiques provenant principalement des verbes de la phrase.

L'application de l'analyseur syntaxique XIP[4] permet d'identifier les syntagmes verbaux. Puis l'utilisation des patrons lexico-syntaxiques, sert pour déduire la bonne relation sémantique parmi les relations possibles. Par exemple, pour les phrases de la forme « X [Therapeutic or Preventive Procedure] used to treat Y [Disease or Syndrome] », on peut déduire que la relation sémantique entre X et Y est treats (traite). Si ces patrons ne permettent pas de sélectionner une relation unique, nous gardons toutes les relations possibles.

IV.2.c Construction d'un graphe sémantique associé à la phrase analysée :

Les nœuds de ce graphe correspondent aux entités nommées identifiées et les arcs correspondent aux relations sémantiques entre ces termes. Par exemple on a la phrase (1) suivante : "Nasogastric intubation is used to treat gastric atony".

[3] Le réseau sémantique organise les concepts du Méta thésaurus avec des types sémantiques (135 types)

[4] XIP : Xerox Incremental Parsing

Le graphe sémantique associé à cette phrase est présenté dans la **figure1.4**. Les entités nommées « nasogastric intubation » et « gastric atony » forment les nœuds du graphe. L'arc les reliant correspond à la relation sémantique « treats ».

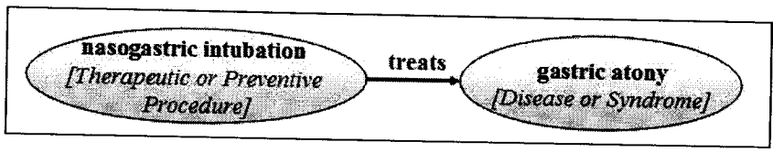


Figure 1.4 : Graphe sémantique correspondant à la phrase (1) [2]

Pour construire le graphe sémantique d'une question, nous avons besoin, en plus des entités nommées médicales, des informations suivantes : le focus de la question et le concept générique correspondant et le type de la réponse attendue désigné par un concept générique (par exemple traitement, maladie).

Nous différencions également deux types de relations sémantiques entre les entités nommées d'une question : relations sémantiques entre le concept générique correspondant au focus et le type de la réponse attendue et autres relations sémantiques possibles entre les entités nommées de la question. Le premier type de relation est plus prioritaire lors de la recherche de la réponse car il correspond au cœur de la question alors que le deuxième type correspond au contexte sémantique de la question.

Exemple : Voila la question (Q)

« 45-year-old woman with dysfunctional uterine bleeding. What is the treatment ?»

Dans le graphe sémantique associé à cette question présenté dans la **figure 1.5**, la relation « treats » lie la réponse attendue et le focus de la question (dysfunctional uterine bleeding).

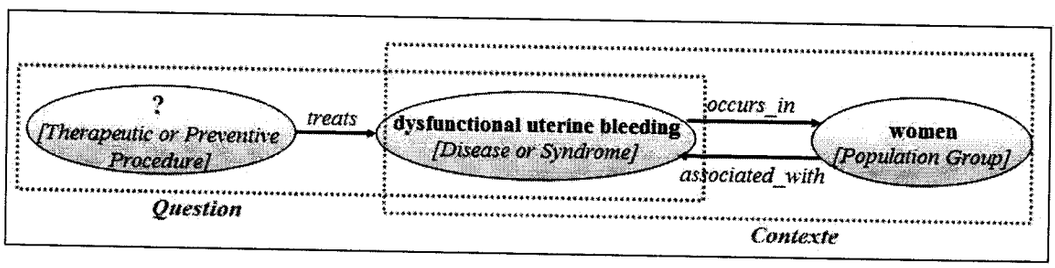


Figure 1.5 : Graphe sémantique correspondant à la question (Q) [2]

Conclusion :

Dans ce chapitre, on illustre la notion de question-réponse ainsi que l'intérêt de la recherche d'information précise. Cette information précise peut être extraite au moyen de systèmes de recherche d'information automatisés, plus précisément des systèmes capables de satisfaire des requêtes d'interrogation formulées par les utilisateurs en renvoyant uniquement une réponse précise en un minimum de temps. Ces systèmes sont appelés « systèmes de question-réponse ».

Les systèmes de question-réponse existants montrent que l'architecture classique d'un tel système repose sur trois modules. Le premier porte sur l'analyse de la question, le deuxième sur la recherche et la sélection de documents pertinents tandis que le dernier module se concentre sur l'extraction de la réponse recherchée.

L'ambition commune de ces systèmes est d'exploiter en premier lieu la question afin d'en extraire tous les traits syntaxiques et sémantiques qu'elle contient. C'est une étape cruciale qui joue un rôle prépondérant sur la performance du système de question-réponse.

Chapitre II

Ressources terminologiques et sémantiques du domaine médical

I. Introduction [1]:

Le domaine médical constitue l'un des domaines de spécialité les plus importants et les plus traités depuis l'essor de l'informatique. Il se caractérise par une terminologie riche et complexe qui ne cesse en outre de croître du fait des évolutions rapides des recherches qui y sont menées.

La richesse et la complexité du vocabulaire médical ont conduit depuis de nombreuses années au développement d'un ensemble important de ressources terminologiques et sémantiques telles que le MeSH[5] ou l'UMLS[6] par exemple. Ces ressources ont été constituées dans le but d'une part, de normaliser la terminologie médicale et d'autre part, de faciliter l'accès à l'information médicale. L'effort qui sous-tend leur réalisation a permis à la fois une modélisation de la connaissance médicale et une meilleure structuration des données. L'utilisation de ces ressources permet d'identifier plus facilement dans les textes les termes médicaux et sont très utiles pour de nombreuses applications comme l'indexation des documents médicaux, la recherche d'information par exemple son intégration dans les systèmes de question-réponse adaptés à la médecine.

II. Représentation des connaissances [12]:

Il existe tout un panneau de systèmes de structuration pour représenter les connaissances: classification, thésaurus, terminologie, ontologie.

- Une classification est la répartition systématique en classes, en catégories d'êtres, de choses ou de notions ayant des caractères communs notamment afin d'en faciliter l'étude. Notons que les classes peuvent être organisées entre elles hiérarchiquement selon un principe générique-spécifique. Un exemple de classification est la CIM-10[7].
- Un thésaurus est un ensemble structuré de termes nécessaires à son utilisation au sein d'une hiérarchie de concepts liés par des relations sémantiques par exemple MeSH.

[5] http://www.nlm.nih.gov/mesh/

[6] http://nlm.nih.gov/research/umls/

[7] http://www.icd10.ch/index.asp

- ❖ Une terminologie est une liste de termes d'un domaine ou sujet donné représentant les concepts ou notions les plus fréquemment utilisés ou les plus caractéristiques. Contrairement à un thésaurus, dans une terminologie, l'accent est mis sur l'exhaustivité des termes (synonymes, abréviations). Un exemple de terminologie est la SNOMED.
- ❖ Une ontologie est un ensemble de concepts et de relations pour une utilisation particulière d'un domaine déterminé ; cette structure repose sur une formalisation avouée afin d'effectuer des inférences dans un système informatique. Un exemple d'ontologie est la SNOMED-CT.

III. Ressources terminologiques et sémantiques du domaine médical :

On présente dans ce qui suit quelques ressources terminologiques et ontologiques existantes explicitement conçues pour le domaine médical. Ces ressources ont été construites pour répondre à des besoins précis et divers :

- ❖ Le thésaurus MeSH est utilisé pour indexer des documents médicaux dans des bases documentaires,
- ❖ L'UMLS a comme objectif de faciliter le développement de systèmes informatisés afin d'améliorer l'accès à l'information médicale,
- ❖ La CIM permet le codage des dossiers patients à des fins statistiques,
- ❖ Et enfin la SNOMED est une nomenclature utilisée pour le codage des dossiers électroniques des patients. [4]

III.1 MeSH : [1]

Le MeSH (Medical Subject Heading) est un thésaurus numérisé. Il a été développé par la National Library of Medecine (NLM), principalement pour indexer la base bibliographique MEDLINE[8]. Il est traduit en français par l'INSERM[9]. De nos jours, ce thésaurus est également utilisé pour l'indexation de nombreuses sources de données médicales. Le MeSH est une liste structurée de termes médicaux organisés en une arborescence. Au fur et à mesure que l'on descend dans la hiérarchie, les termes sont de plus en plus spécifiques. Ces termes sont appelés « descripteurs » car ils expriment de manière précise et spécifique le contenu d'un document. Les descripteurs, au nombre de 23 000 (en 2005), sont regroupés en 15 branches majeures. Par exemple la branche « A

[8] Medical Literature Analysis and Retrieval System Online

[9] INSERM est l'Institut National de la Santé Et de la Recherche Médicale

» correspond à l'anatomie (Anatomy), la branche « B » aux organismes (Organisms), la branche « C » aux noms de maladies (Diseases). Chacune de ces branches contient plusieurs sous branches qui constituent les différents niveaux de la hiérarchie. Par exemple « C01 » pour la catégorie « Infections bactériennes et mycoses » (Bacterial Infections and Mycoses), « C02 » pour « Maladies virales » (Virus Diseases) ou encore « C03 » pour « Maladies parasitaires » (Parasitic Diseases).

Par ailleurs, chaque terme du thésaurus MeSH est associé à sa définition, ses synonymes et sa position dans l'arborescence (identifiant hiérarchique). Cependant, certains descripteurs peuvent apparaître dans plusieurs branches de l'arborescence, c'est-à-dire qu'un même terme peut appartenir à plusieurs catégories du MeSH et par conséquent, il peut donc avoir plusieurs identifiants. Un identifiant est composé d'un numéro alphanumérique : une lettre qui précise la catégorie (comme C = Maladies) et une série de nombres qui indiquent la position du terme dans la hiérarchie. Par exemple, l'identifiant attribué au descripteur « Hépatite C » est « C02.440.440 », ce qui signifie : « C » pour Maladie, « C02 » pour la catégorie « Maladies virales », « C02.440 » pour « Hépatites virales humaines » et ainsi de suite.

III.2 SNOMED : [1]

La SNOMED (Systematized Nomenclature of Medicine) (http://www.snomed.org/) est une nomenclature de type classification multiaxiale. La version SNOMED 3.5 développer en 1998 comprend plus de 200 000 termes médicaux couvrant plusieurs domaines de la médecine.

SNOMED renferme des concepts de base qui peuvent être associés pour décrire des diagnostics ou des actes professionnels. Son vocabulaire est organisé selon onze axes de classification définis par une lettre (par exemple, T pour topographie, M pour morphologie). Les éléments à l'intérieur de chaque axe sont organisés suivant une structure hiérarchique. La classification d'un terme repose sur une décomposition de celui-ci en combinaison de termes appartenant à différents axes. Par exemple, la juxtaposition :

M4405 (granulome éosinophile), F0300 (fièvre), E2001 (tuberculose) et T2800 (poumon) correspond à la phrase « tuberculose pulmonaire ». Cette possibilité de combiner des termes appartenant à des classes différentes avec des qualificatifs et des termes relationnels permettant ainsi de composer des expressions fait de la SNOMED une terminologie très importante dans le domaine médical, notamment pour l'indexation des dossiers médicaux.

III.3 CIM-10 : [1,4]

La Classification Internationale des Maladies (CIM-10) (en anglais ICD pour International Classification of Diseases) publiée par l'Organisation Mondiale de la Santé (OMS), est apparue en 1993.

La CIM bénéficie d'une remise à niveau régulière, le chiffre 10 correspond à la dernière version exploitable de la classification (1993). Une nouvelle révision de la CIM est en cours de lancement dans le cadre du projet (CIM-11) administré par l'OMS.

La classification dans CIM-10 est mono axiale comprenant 21 chapitres principaux dont 17 concernent des maladies et 4 concernent les signes, les causes et les facteurs de recours aux soins. Les maladies sont classées selon plusieurs catégories telles que : les maladies du système nerveux (G), les maladies de l'appareil circulatoire (I). Elles sont répertoriées suivant leur degré de gravité. Par exemple, le chapitre des maladies infectieuses recense le plus grand nombre d'entrées car ces maladies sont la première cause de morbidité et de mortalité dans le monde. Chaque entrée est identifiée dans la CIM par un code. Ce dernier est composé de quatre caractères : une lettre correspondant au chapitre suivie de trois chiffres pour spécifier les maladies définies à un niveau général. Par exemple, le code A15.9 indique une tuberculose de l'appareil respiratoire.

III.4 UMLS : [1,4]

L'UMLS (Unified Medical Language System) (pour Système d'unification de la langue médicale) est actuellement la ressource terminologique de référence pour le domaine biomédical. Cette ressource, développée et maintenue par la NLM depuis 1986, est le résultat de la compilation d'une centaine de thésaurus de langues et structures différentes dont le MeSH et la SNOMED pour les plus connus d'entre eux, ce qui lui confère le statut de métathésaurus multilingue. Ce métathésaurus comporte donc la terminologie résultant de l'union des vocabulaires de ces différentes sources médicales tout en préservant les relations intervenant entre les termes.

L'UMLS est constitué de plus d'un million de concepts (version 2006) et indique les relations existant entre les concepts. Ces derniers, au nombre graduellement croissant. Les relations sémantiques présentes dans l'UMLS sont principalement des relations de nature paradigmatique telles que les relations de synonymie ou d'hyperonymie ainsi que d'autres relations plus spécifiques comme la relation « affecte ». Par ailleurs, l'UMLS dispose d'un vaste réseau sémantique comportant 134 types hiérarchisés par le lien « is-a ». Ce réseau fait de l'UMLS la ressource terminologique du domaine médical la plus largement exploitée. Elle s'avère très appropriée pour le traitement de l'information

biomédicale et par conséquent, elle constitue un outil précieux pour les systèmes de recherche documentaire, notamment pour repérer dans les documents médicaux les concepts spécifiques au domaine biomédical comme les gènes, les maladies ou encore les médicaments.

Cependant, l'utilisation de l'UMLS et de son réseau sémantique se révèle difficile pour la langue française puisque la majorité des termes intégrés dans le méta thésaurus UMLS sont en langue anglaise. En fait, la terminologie en français ne couvre que 2% des concepts présents dans l'UMLS.

III.5 SNOMED-CT : [13]

La SNOMED-CT (Systematized Nomenclature of MEDicine-Clinical Terms) est une ontologie multilingue de la santé clinique. Il s'agit d'une structure hiérarchique de concepts désignés par des descriptions (termes) sur plus de 31 niveaux de subsomption. Elle contient plus de 311 000 concepts, près de 800 000 termes et 1 360 000 relations en janvier 2008.

Cette terminologie a pour vocation d'être utilisée pour tous documents cliniques par exemple des dossiers patients électroniques et des systèmes informatiques des hôpitaux.

Conclusion :

Dans ce chapitre, on présente quelques ressources terminologiques du domaine médical accessibles sur le Web. Les terminologies disponibles en langue française sont plus limitées par rapport à la terminologie en anglais, ces ressources sémantiques contiennent majoritairement des relations paradigmatiques de type synonymie ou hyperonymie d'où les termes sont reliées par des relations hiérarchiques et sont beaucoup moins riches en relations syntagmatiques comme celles spécifiant qu'une maladie « M » peut être traité par le traitement « T » ou encore que l'examen « E » permet de détecter la maladie « M ».

Chapitre III

Ontologies

I. Introduction [4] :

Durant cette dernière décennie, Nous avons remarqué qu'une attention croissante a été concentrée sur l'ingénierie ontologique où l'ontologie est l'objet fondamental sur lequel il faut se penser.

Les ontologies sont largement utilisées et ont prouvé leurs utilités dans de nombreux domaines tels que : l'ingénierie de connaissances, l'intelligence artificielle, l'intégration des sources de données, la recherche d'information, la commerce électronique et sont au cœur du Web Sémantique. Cette utilité est motivée par le fait que les ontologies sont un moyen efficace pour la gestion et le partage des connaissances d'un domaine particulier entre personnes et systèmes.

II. Définition d'une ontologie [4]:

Historiquement, l'ontologie est un terme philosophique qui signifie la science et la théorie de l'être. L'origine de ce terme est grec, c'est la composition de deux autres termes, *On* signifie être et *Logos* signifie science.

En informatique, la littérature fournit un tas de définitions du mot ontologie. Ces définitions, dans leur diversité, offrent des points de vues à la fois différents et complémentaires :

- Neches et ses collègues en 1991 ont été les premiers à en proposer une définition :

[Une ontologie définit les termes et les relations de base du vocabulaire d'un domaine ainsi que les règles qui indiquent comment combiner les termes et les relations de façon à pouvoir étendre le vocabulaire]. Cette définition descriptive donne un premier aperçu sur la manière de construire une ontologie, à savoir l'identification des termes de bases d'un domaine et les relations entre ces termes ainsi que les règles pouvant s'appliquer sur ces derniers.

- En 1997 Swartout et ses collègues au sein du projet SENSUS :

[Une ontologie est un ensemble de termes hiérarchiquement structurés, conçu afin de décrire un domaine qui peut être utilisé comme un squelette de base pour les bases de connaissances]. Selon cette définition, une ontologie peut servir à construire plusieurs bases de connaissances qui peuvent partagés la même taxonomie.

- En 1998, Studer et ses collègues propose la définition suivante :

[Une ontologie est une spécification formelle et explicite d'une conceptualisation partagée]. Ils l'expliquent comme suit :

- Spécification explicite signifie que les concepts, les propriétés, les relations, les fonctions, les axiomes de l'ontologie sont définis de façon déclarative ;
- Formelle réfère au fait qu'une ontologie doit être traduite dans un langage interprétable par une machine;
- Conceptualisation réfère à un modèle abstrait d'un phénomène du monde en identifiant les concepts appropriés à ce domaine ;
- Partagé réfère au fait qu'une ontologie capture la connaissance consensuelle c'est-à-dire non réservée à quelque individus, mais partagée par un groupe ou une communauté.

III. Composantes d'une ontologie [4,7]:

Les ontologies rassemblent les connaissances propres à un domaine particulier. Ces connaissances sont formalisées en mettant en jeu les composants suivants: concepts ; relations ; axiomes et instances.

III.1 Concepts :

Un concept peut représenter un objet, une idée, ou bien une notion abstraite. Ils sont appelés aussi classes de l'ontologie dans certain travaux. Un concept peut être divisé en trois parties : un terme (ou plusieurs), une notion et un ensemble d'objets.

- Le terme (ou bien label) d'un concept est l'expression linguistique utilisée couramment pour y faire référence.
- La notion désigne ce qui est appelé, au sens de la représentation des connaissances, l'intension du concept. Elle contient sa sémantique qui est définie à l'aide de propriétés (relations et attributs), de règles et de contraintes.
- L'ensemble d'objets définis par le concept forme ce qui est appelé l'extension du concept. Il s'agit des objets auxquels le concept fait référence, autrement dit, de ses instances.

III.2 Relations :

Les ontologies généralement contiennent que des relations binaires. Le premier argument d'une relation binaire est dit domaine, alors que le deuxième argument est dit Co-domaine. Cela permet de designer la façon dont la relation doit être lue.

On distingue alors, les relations taxonomiques (dite aussi de subsomption) et les relations associatives.

III.2.a Relation de subsomption :

La relation de subsomption « est-un » (is-a) a un statut particulier car elle structure la hiérarchie ontologique. Un concept C1 (concept père) subsume un concept C2 (concept fils) si toute propriété sémantique de C1 est également une propriété sémantique de C2. La relation de subsomption n'est pas la seule relation qui permette de structurer la hiérarchie ontologique, la relation de méronymie, « partie de » (part-of) est souvent utilisée.

L'héritage multiple est une propriété qui peut être définie sur la relation de subsomption : un concept d'une ontologie peut avoir plusieurs pères par la relation de subsomption. L'héritage multiple implique que le concept hérite des propriétés de tous ses pères.

III.2.b Relation associative :

Les relations « associatives » sont des relations d'interaction entre deux concepts qui ne sont pas la relation de subsomption. La désignation « relation associative » est empruntée aux domaines de la bioinformatique

III.3 Fonctions :

La fonction est un cas particulier de relations dont un élément d'une relation est unique par rapport aux éléments qui le précèdent. Un exemple d'une fonction binaire est « *Mère-de* » qui donne la mère d'un individu. Ce dernier, doit avoir une seule mère.

III.4 Axiomes :

Les axiomes modélisent les connaissances considérées comme vrais dans le domaine traité. Leur inclusion dans une ontologie peut avoir plusieurs objectifs: interviennent dans la définition des significations des composants d'ontologie, les contraintes sur les valeurs des attributs.

III.5 Instances (ou individus):

Elles constituent la définition extensionnelle de l'ontologie. Ils représentent les éléments singuliers véhiculant les connaissances à propos du domaine.

IV. Différents langages d'ontologies :

Plusieurs langages, dont la syntaxe est basée sur le langage XML, ont été conçus pour une utilisation des ontologies. Parmi eux, les plus importants sont RDF/RDF(S), qui est la recommandation par l'organisme de normalisation du Web W3C[10] pour représenter les métadonnées, DAML+OIL, OWL qui est la recommandation du W3C

[10] W3C : World Wide Web Consortium

pour représenter des ontologies. La figure 3.1 faire une revue des langages présentés. [9]

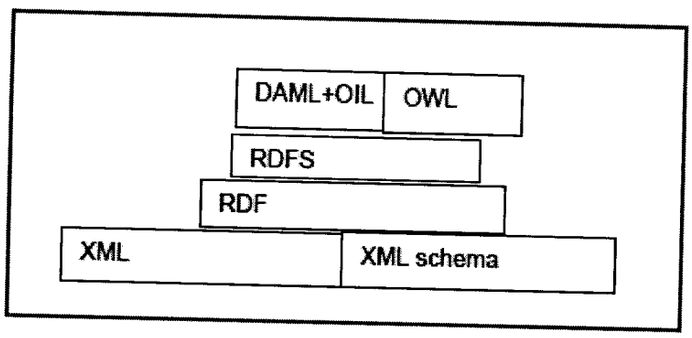


Figure 3.1 : Langages d'ontologies [9]

IV.1 XML, XML schéma : [6,8]

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 (par exemple : entier, réel, décimal) qui peuvent être utilisés dans des documents XML. Chaque élément d'information dans un document XML doit se trouver entre deux balises (la deuxième balise ayant le caractère « / » au début).

Exemple :

```
<employé>
  <numéro> 1234 </numéro>
  <nom> Dupont </nom>
</employé>
```

Tout ce qui concerne l'employé est entre les balises <employé> ..</employé>. À l'intérieur des balises <numéro>…</numéro> se trouve l'élément numéro et ce qui se trouve entre les balises du nom forme le nom de l'employé. L'imbrication des balises définit une arborescence.

XML permet de séparer le contenu de la présentation de l'information ce qui facilite les changements des formats. Une déclaration de type de document DTD[11] 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.

[11] DTD : Document Type Declaration

IV.2 RDF (S): [6]

RDF[12] (Resource Description Framework) est un langage pour représenter de l'information concernant des ressources sur le Web. RDF est conçu pour représenter des métadonnées au sujet des ressources. RDF définit un modèle de donnée pour assigner une sémantique aux données en utilisant la syntaxe de XML. Le modèle de donnée consiste en trois types d'objet :

A. Ressource : Une ressource peut être une page HTML, une partie de page, un ensemble de pages ou un objet qui n'est pas accessible sur internet comme un livre ou une personne. Les ressources sont représentées par un URI (Uniform Resource Indentifier).

L'URI permet d'assigner un nom unique à un objet.

B. Propriété : Une propriété est une caractéristique, attribut ou relation qui décrit une ressource.

C. Triplet (Statement) : Un triplet désigne une ressource avec une propriété ainsi que la valeur de cette propriété pour la ressource. Les trois éléments d'un triplet sont nommés : sujet, prédicat et objet pour la ressource la propriété et la valeur de la propriété.

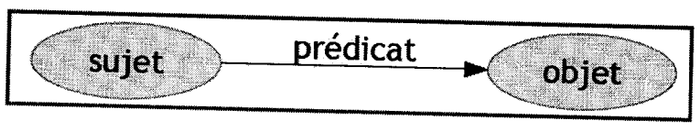
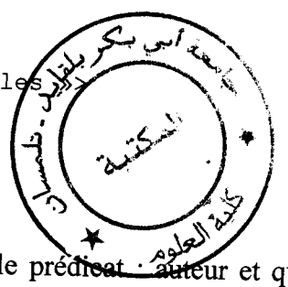


Figure 3.2 : Triplets de RDF [6]

Les ressources sont décrites en termes de triplet par exemple si nous voulons signifier que le livre « Les Misérables » est écrit par Victor Hugo, le triplet suivant pourrait être utilisé.

```
<rdf : RDF>
  <rdf :Description about= « http://…/Les Misérables »
     <auteur>Victor Hugo</auteur>
  </rdf :Description>
</rdf :RDF>
```



Ce triplet décrit que le sujet : http://…/Les Misérable » a le prédicat : auteur et que l'objet est Victor Hugo. Comme le sujet et l'objet d'un triplet peuvent être des

[12] http://www.w3.org/RDF/

ressources, il est possible de lier en chaîne des triplets. Par exemple les deux triplets suivants illustrent cette idée :

```
<rdf :RDF>
  <rdf :Description about= « http://…/Jean Dupont »>
    <est employé de> « http://www.compagnie.com/ABC.inc »</est employé
de>
  </rdf :Description>
</rdf :RDF>

<rdf :RDF>
  <rdf :Description about= « http://www.compagnie.com/ABC.inc »>
    <farbrique> savon</fabrique>
  </rdf :Description>
</rdf :RDF>
```

Dans le premier triplet l'objet « http//www.compagnie.com/ABC.inc » est une ressource qui est le sujet du deuxième triplet. Cela signifie que Jean Dupont est employé d'ABC.inc et cette compagnie fabrique du savon.

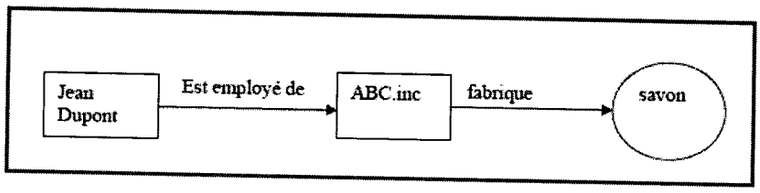


Figure 3.3: Représentation graphique de triplets chaînés [6]

Les spécifications de RDF Schema viennent enrichir les primitives de RDF en assignant une sémantique supplémentaire aux ressources. La combinaison de RDF et de RDF Schema est connue sous le nom RDF(S). RDF Schema 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. Une classe y est définie par les balises <rdfs : Class>… </rdfs :Class>.

  La classe d'une instance est définie par la balise <rdf : type>.

  Il est également possible de représenter une relation entre les classes par le concept de sous classes :<rdfs :subClassOf>. Par exemple « auto » est une sous classe de la classe « véhicule ». La propriété de sous-classes est transitive ainsi si C est une sous-classe de B et B est une sous-classe de A donc C est également une sous-classe de A.

- Hiérarchiser les prédicats ou propriétés par le concept de sous propriété : <rdfs:subPropertyOf>. Par exemple la propriété « est père de » est une sous propriété de « est parent de ». Une propriété peut avoir des contraintes sur le sujet et l'objet auxquels elle s'applique par les balises « domain » et « range ». Ces contraintes ont été ajoutées pour permettre de valider les données RDF. La balise <rdfs : domain> spécifie les types de classe du sujet où s'applique la propriété tandis que la balise <rdfs : range> définit le type de classe de l'objet. Par exemple pour une propriété « est père » peut avoir comme <rdfs : domain> les classes de type masculin et un <rdfs : range> sur les classes de type enfant.
- RDF Schema ajoute des fonctionnalités pour ajouter des commentaires « rdfs : comment » et des noms d'étiquettes « rdfs : label ».

IV.3 DAML-OIL : [10,11]

Pour ajouter une plus grande expressivité au langage RDF(S), DAML+OIL utilise les primitives de la logique de description. DAML+OIL est issu de l'union de deux langages: DAML-ONT (DARPA Agent Markup Langage) et OIL (Ontology Inference Layer) qui provient d'Europe. À l'origine DAML-ONT était un programme du gouvernement américain en vue de développer un langage pour le Web sémantique. OIL provient du projet européen « On-To-Knowledge », ce langage ajoute les primitives de langage de frame à RDF(S).

IV.3.a Constructeurs de classe DAML+OIL :

Tout comme RDF schéma, DAML+OIL supporte les notions de classes et de propriétés, mais DAML+OIL vient enrichir l'expressivité de cette notion par la logique de description (LD). Dans DAML+OIL, tout comme en logique de description, les classes peuvent être des noms (un URI) ou des expressions. Des constructeurs sont utilisés pour créer des expressions de classe. La table 3.1 illustre quelques constructeurs de classe du DAML+OIL avec leurs correspondances en logique de description.

Constructeurs	Syntaxe en logique de description	Exemple
IntersectionOf	$C_1 \wedge .. \wedge C_n$	Humains ∧ Mâle
UnionOf	$C_1 \vee .. \vee C_n$	Comptable ∨ Pompier
ComplementOf	$\neg C$	¬ Mâle
OneOf	$\{x_{1..}, x_n\}$	{Canada, Mexique}

Table 3.1 : Constructeurs de classe DAML+OIL [11]

Par exemple pour illustrer le constructeur d'intersection d'Humain et de Mâle, DAML+OIL utilise la syntaxe RDF suivante :

```
<daml : Class>
  <daml :intersectionOf rdf :parseType= « daml :collection»>
    <daml:Class rdf:about="#Humain"/>
    <daml:Class rdf:about="#Mâle"/>
  </daml :intersectionOf>
</daml:Class>
```

Les trois premiers constructeurs du tableau sont les opérateurs booléens standards. Ainsi l'exemple du constructeur « intersectionOf » indique qu'une nouvelle classe est créée à partir des classes Humain et Mâle pour former la classe des mâles qui sont des humains. Le constructeur « unionOf » représente une nouvelle classe qui est soit les comptables ou les pompiers qui sont issue des classes Comptable et Pompier. Le constructeur « complementOf » est la négation, ainsi la classe de tout ce qui n'est pas un mâle est faite à partir de la classe Mâle.

Le constructeur « oneOf » permet de définir une classe en énumérant ses éléments. Dans l'exemple la classe des pays d'Amérique du Nord pourrait être la liste Canada et Mexique.

IV.4 Langage OWL (Web Ontology Language):

Au cours de juillet 2002 une proposition du groupe de travail sur les ontologies du Web à été faite pour modifier DAML+OIL en un nouveau langage OWL qui est une version expurgée des éléments redondants et moins utiles de DAML+OIL.

OWL est, tout comme RDF, un langage XML profitant de l'universalité syntaxique de XML[13]. OWL offre un moyen d'écrire des ontologies web. OWL se différencie du couple RDF/RDFS en ceci que, contrairement à RDF, il est justement un langage

[13] Extensible Markup Langage

d'ontologies. SI RDF et RDFS apportent à l'utilisateur la capacité de décrire des classes (avec des constructeurs) et des propriétés, OWL intègre, en plus, des outils de comparaison des propriétés et des classes : identité, équivalence, contraire, cardinalité, symétrie, transitivité, disjonction. [11]

IV.4.a Différentes déclinaisons d'OWL : [8]

Plus un outil est complet, plus il est, en général, complexe. C'est cet accueil qu'a voulu éviter le groupe de travail WebOnt du W3C en dotant OWL de trois sous-langages offrant des capacités d'expression croissantes et, naturellement, destinés à des communautés différentes d'utilisateurs :

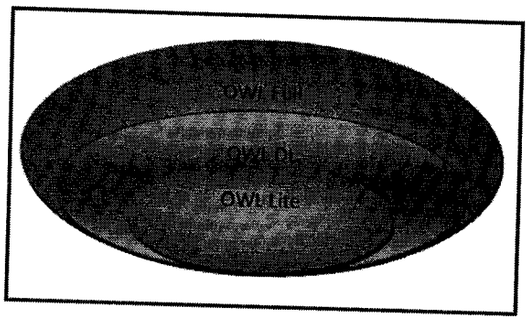


Figure 3.4 : Les 3 niveaux d'OWL [8]

- **OWL Lite** est le sous langage d'OWL le plus simple. Il a la complexité formelle la plus basse et l'expressivité minimale dans la famille OWL. Il est suffisant pour représenter des thésaurus et d'autres taxonomies ou des hiérarchies de classification avec des contraintes simples.
- **OWL DL** est plus complexe qu'OWL Lite, permettant une expressivité plus importante. OWL DL est fondé sur la logique descriptive (d'où son nom, OWL Description Logics) et contient l'ensemble des constructeurs, mais avec des contraintes particulières sur leur utilisation qui assurent la décidabilité de la comparaison de types.
- **OWL Full** est la version la plus complexe d'OWL, mais également celle qui permet le plus haut niveau d'expressivité. OWL Full offre cependant des mécanismes intéressants, comme par exemple la possibilité d'étende le vocabulaire par défaut d'OWL.

Il existe entre ces trois sous langage une dépendance de nature hiérarchique : toute ontologie OWL Lite valide est également une ontologie OWL DL valide, et toute ontologie OWL DL valide est également une ontologie OWL Full valide.

Conclusion :

La majorité des approches de recherche d'information visant à intégrer une ontologie dans leur procédé reposent sur des ontologies existantes. Généralement, l'unique caractéristique prise en compte dans le choix de l'ontologie est le domaine de connaissance représentée dans l'ontologie qui doit couvrir le domaine traité dans le corpus.

L'ontologie permet de représenter des connaissances propres à un domaine c'est-à-dire consiste à décrire et coder les éléments de ce domaine par un langage de spécification par exemple « OWL » pour qu'une machine puisse les manipuler afin de raisonner.

Chapitre IV

Conception d'une ontologie médicale

I. Introduction :

L'ontologie qu'on va étudier a été définie par la sollicitation directe des médecins et par l'analyse des questions typiquement posées par des médecins généralistes.
Dans notre analyse on a réalisé une étude de cas pour une ontologie médicale.

II. Ontologie du domaine médical [1]:

La figure 4.1 représente une ontologie du domaine médical.

Dans l'ontologie, les concepts médicaux sont représentés par des rectangles tels que le concept « Maladie », qui centralise toutes les expressions désignant des noms de maladies ou le concept « Symptôme », qui regroupe toutes les manifestations cliniques révélant la présence potentielle d'une maladie. Les relations, qui expriment le type d'interaction entre deux concepts, sont quant à elles représentées par des flèches permettant ainsi de déterminer le sens de lecture d'une relation telle que la relation « Traite » entre les deux concepts « Traitement » et « Maladie ». De plus, entre deux mêmes concepts, plusieurs relations peuvent intervenir, comme c'est le cas des deux relations « Contre-indication » et « Soigne » entre « Médicament » et « Maladie ».
Il est à noter que le concept « Phénomènes » représente ici toutes les manifestations pathologiques. Elles peuvent être de nature psychologique (stress, troubles psychiques) ou physiologique (formation de vaisseaux collatéraux, eczéma). Des propriétés liées aux médicaments (classe, posologie, forme). Ces informations sont parfois très utiles pour un médecin lors des prescriptions des ordonnances, ainsi, pour distinguer les différents types de traitements, trois concepts ont été définis : traitements physiques (massages, exercices physiques,), traitements médicamenteux et traitements annexes (conseils).

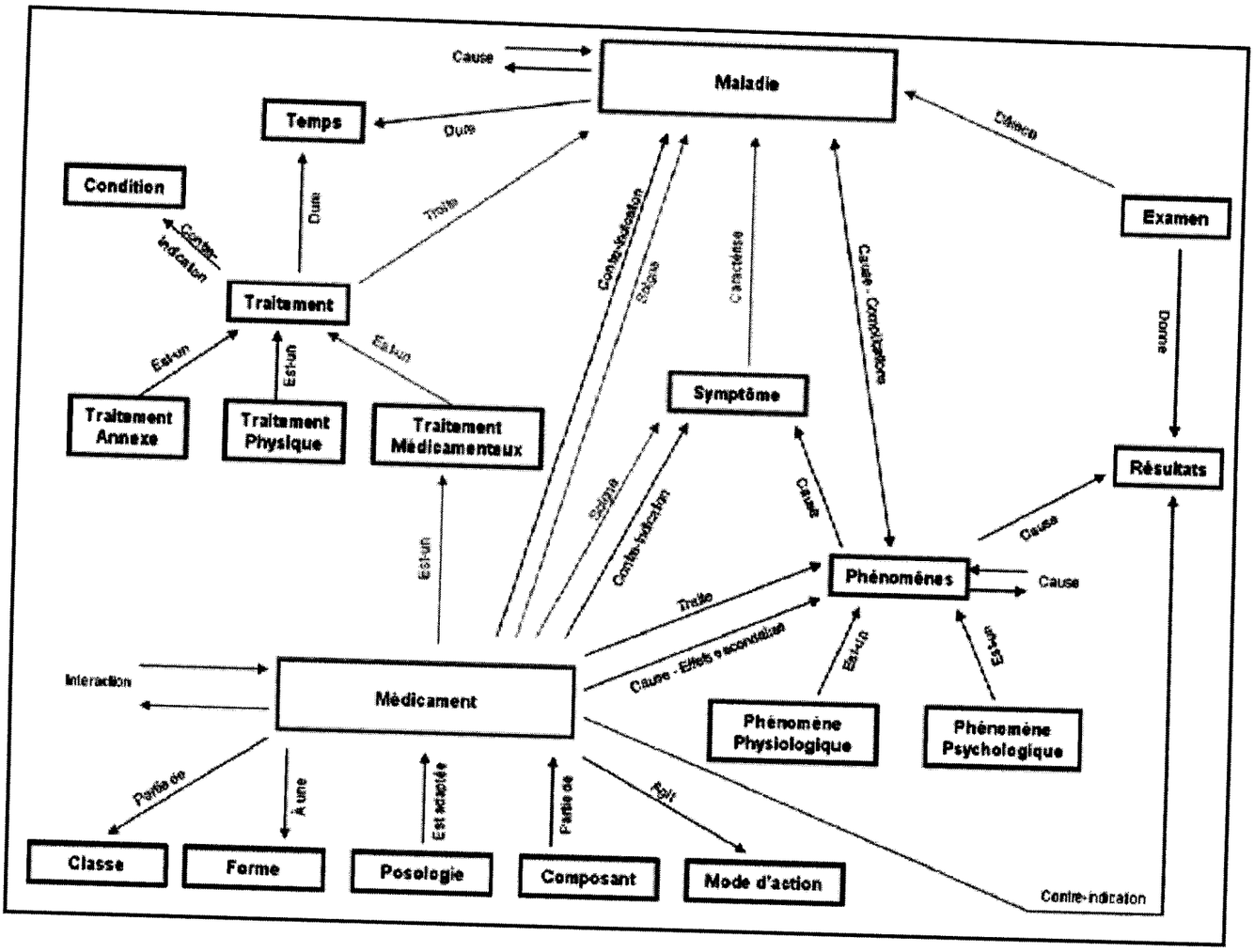


Figure 4.1: Ontologie du domaine médical [1]

III. Conception d'une ontologie médicale :

On présente les différentes étapes pour la construction de l'ontologie médicale:

III.1 Ontologie étudiée :

La figure 4.2 est une représentation du domaine de la médecine général sous la forme d'une ontologie.

Les concepts sont organisés en une arborescence où ils sont reliés entre eux par des relations sémantiques spécifiques (par exemple « consulte » entre Médecin et Patient) et relation hiérarchique (par exemple « est un » entre Médicament et Traitement Médicamentaux).

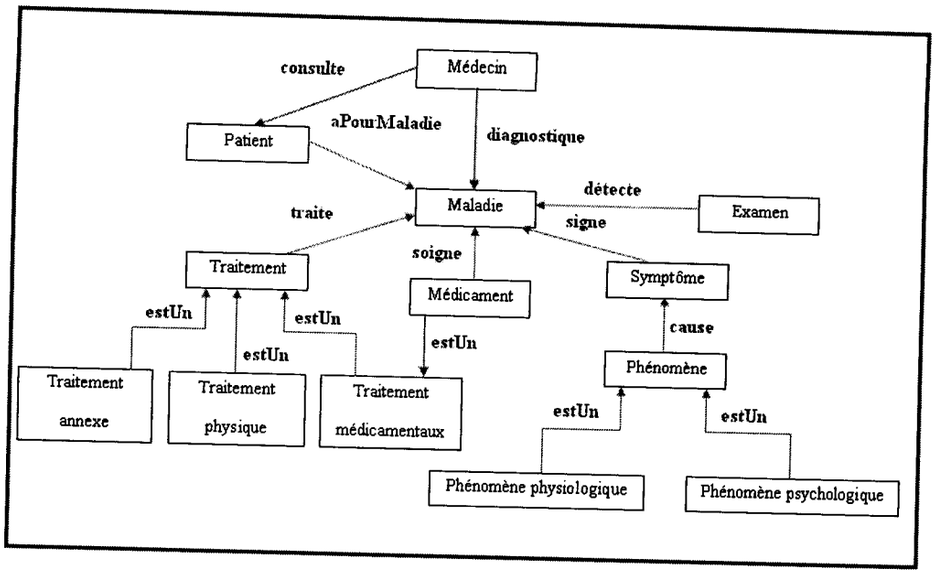


Figure 4.2: Représentation graphique de l'ontologie étudiée

III.2 Liste des concepts et des attributs :

Nous représentons les concepts de notre ontologie, et pour chaque concept son concept père et ses attributs avec leurs types (Voir Tableau 4.1)

Concepts	Concept père	Attributs	Types
Examen	Thing	Code_examen	String
		Type_examen	String
		Nom_examen	String
Maladie	Thing	Code_maladie	String
		Nom_maladie	String
		Type_maladie	String
		Degré_gravité	String
Patient	Thing	Code_patient	String
		Nom_patient	String
		Prénom_patient	String
		adresse	String

Médecin	Thing	code_médecin	String
		nom_médecin	
		Prenom_medecin	
		adresse	
Phénomène	Thing	Code_phénomène	String
		Nom_phénomène	String
Phénomène physiologique	Phénomène	Héritent les mêmes attributs du classe Phénomène	
Phénomène psychologique	Phénomène		
Symptôme	Thing	Code_symptôme	String
		Nom_symptôme	String
Traitement	Thing	Code_traitement	String
		Nom_traitement	String
		Type_traitement	String
Traitement annexe	Traitement	Héritent les mêmes attributs du classe Traitement	
Traitement médicamentaux	Traitement		
Traitement physique	Traitement		

Table 4.1 : Liste des concepts et ses attributs avec leurs types

III.3 Liste des relations entre les différents concepts :

On décrit les relations qui existent entre les différents concepts de notre ontologie (Voir Tableau 4.2).

Relations	Prédécesseurs	Successeurs	Relations inverses
détecte	Examen	Maladie	estDétectéPar
signe	Symptôme	Maladie	aPourSigne
soigne	Medicament	Maladie	estSoignéPar
traite	Traitement	Maladie	estTraitéPar
cause	Phénomène	Symptôme	aPourCause
aPourMaladie	Patient	Maladie	estAssociéAu
consulte	Médecin	Patient	estConsultePar
diagnostique	Médecin	Maladie	estDiagnostiquePar
est un	Phénomène physiologique	Phénomène	
est un	Phénomène psychologique	Phénomène	
est un	Traitement annexe	Traitement	
est un	Traitement médicamenteux	Traitement	
est un	Medicament	Traitement médicamenteux	
est un	Traitement physique	Traitement	

Table 4.2 : Liste des relations entre les différents concepts

La relation « détecte » est une propriété permet de relier le concept « examen » au concept « maladie » ce que signifie qu'un examen détecte une maladie.
La relation inverse de cette propriété « estDétectéPar », ce que signifie de dire une maladie est détectée par un examen.

III.4 Diagramme de classes de l'ontologie médicale:

La figure 4.3 représente le diagramme de classe de l'ontologie médicale.

Les concepts de l'ontologie médicale sont considérés comme des classes publiques[14] qui ont précédé par le signe «+».

Chaque classe à sa propres attributs et le signe «-» avant chaque attribut signifie que ce dernier est privé[15].

[14] Classe publique signifie que leurs attributs et méthodes sont accessible par d'autres classes

[15] Attributs privés signifie que ce dernier n'est pas accessible par d'autres classes

Ce diagramme représente les différentes classes de l'ontologie médicale avec leurs relations.

On a deux types de relations :

Relation d'héritage : c'est une relation binaire de type « est un », par exemple un médicament est un traitement médicamentaux.

Relation associative : c'est une relation binaire entre deux classes, en précisons le nom d'association entre eux et leurs cardinalités, par exemple : un examen peut détecte au moins une maladie ; ainsi qu'une maladie est détecté par plusieurs examens.

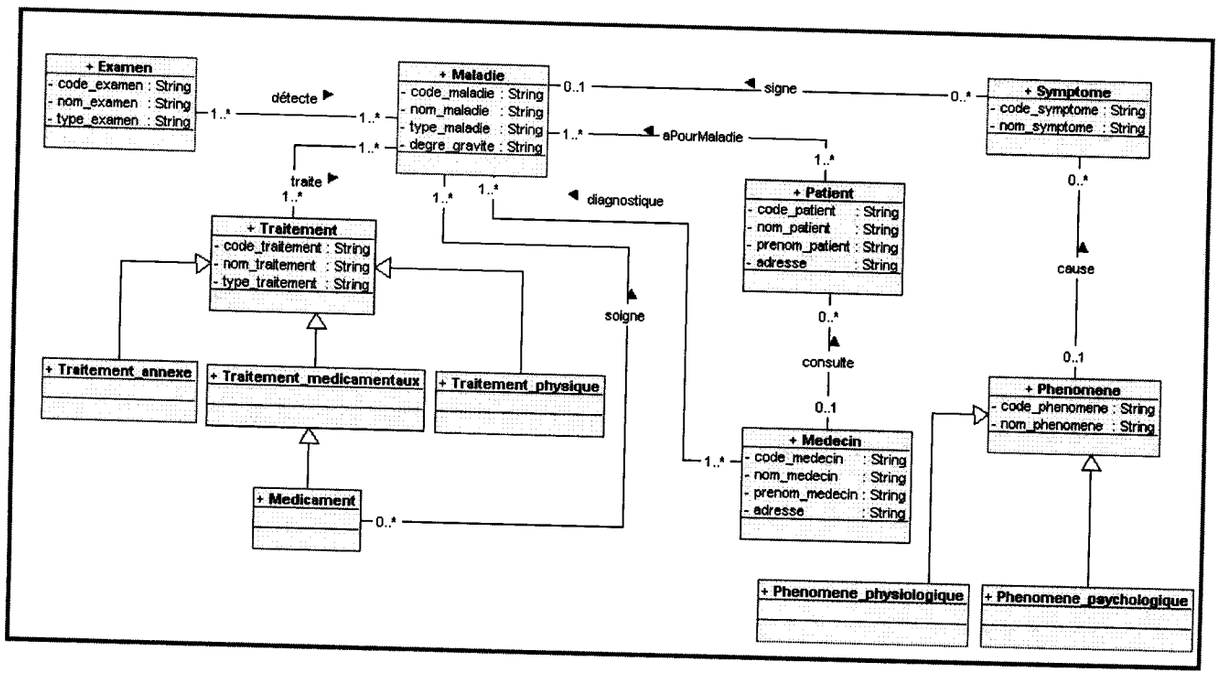


Figure 4.3 : Diagramme de classe de l'ontologie médicale

IV. Diagrammes UML de l'application :

Parmi les méthodes d'analyse et de conception objet, il y a UML[16]. UML constitue une étape importante dans la convergence des notations utilisées dans le domaine d'analyse et la conception objet.

On propose de modéliser les acteurs (Patient et Médecin) et les concepts Maladie, Examen, Médicament, Symptôme, Phénomène et Traitement de l'application.

[16] UML : Unified Modeling Language

IV.1 Patient : On définit également dans ce qui suit les différentes tâches que cet acteur peut participer avec quelques diagrammes.

IV.1.a Diagramme de cas d'utilisation:

Le diagramme « de cas d'utilisation » permet de délimiter le système. Le patient faire la recherche sur une maladie, ses examens, médicaments, symptômes, phénomènes et traitements.

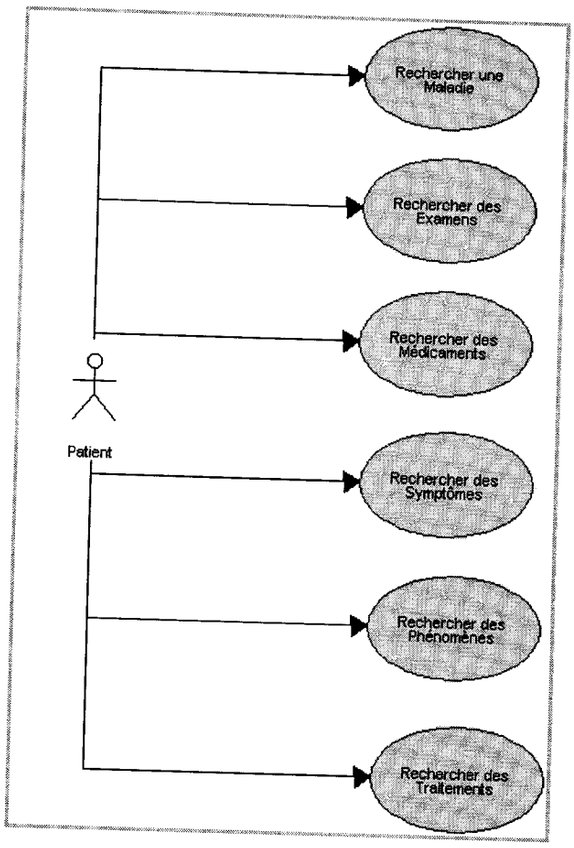


Figure 4.4 : Diagramme de cas d'utilisation pour le Patient

IV.1.b Diagramme de séquences :

Le diagramme de séquence pour le Patient est représenté ci-dessous :

Recherche d'une maladie : Le patient fait une demande de recherche sur une maladie. Le système lui affiche un formulaire, ensuite le patient remplit certains champs et il lance sa recherche sur une maladie, ses examens, médicaments, symptômes, phénomènes et ses traitements. Pour chaque recherche le résultat sera affiché (Voir la figure 4.5)

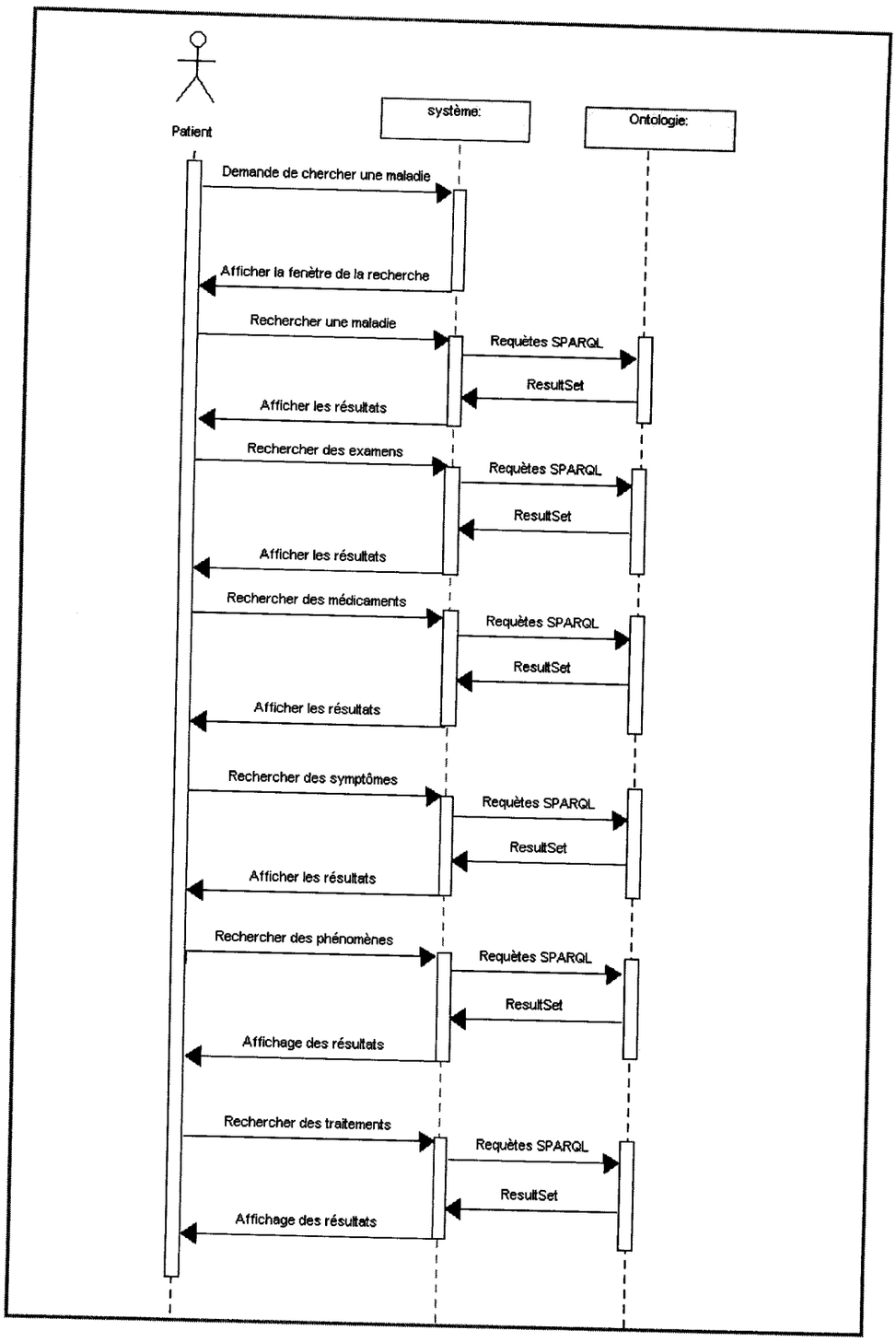


Figure 4.5 : Diagramme de séquence pour la recherche d'une maladie

IV.2 Médecin :

On défini également dans ce qui suit les différentes tâches que cet acteur peut participer avec quelques diagrammes.

IV.2.a Diagramme de cas d'utilisation:

Notre application fournit au médecin les fonctionnalités suivantes :

Ajouter une maladie : Cette fonctionnalité permet au médecin d'ajouter une maladie dans l'ontologie médicale.

Modifier une maladie : Cette fonctionnalité permet au médecin de modifier une maladie existe déjà dans l'ontologie médicale.

Supprimer une maladie : Cette fonctionnalité permet au médecin de supprimer une maladie existe déjà dans l'ontologie médicale.

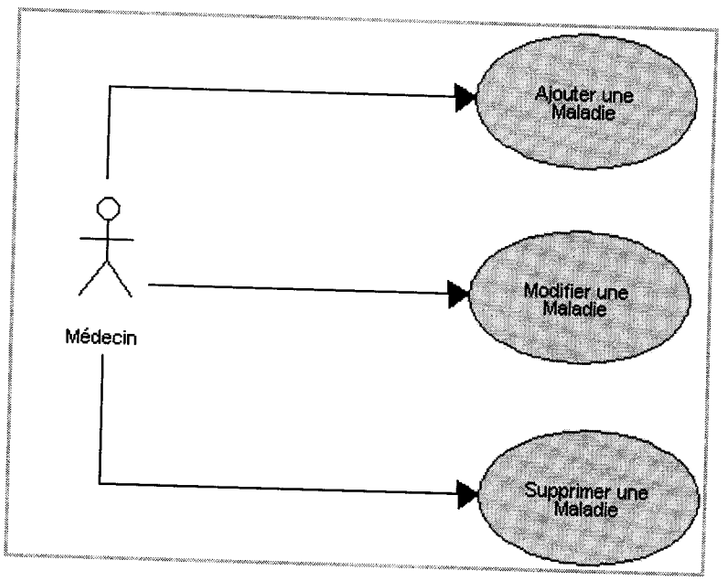


Figure 4.6 : Diagramme de cas d'utilisation pour le Médecin

IV.2.b Diagrammes de séquences :

Les diagrammes de séquence pour l'acteur Médecin sont les suivants :

Ajouter une maladie : Le médecin fait une demande d'ajouter une maladie. Le système lui affiche un formulaire. Le médecin remplit alors certains champs de la fenêtre d'ajout affiché et le valide (voir la figure 4.7)

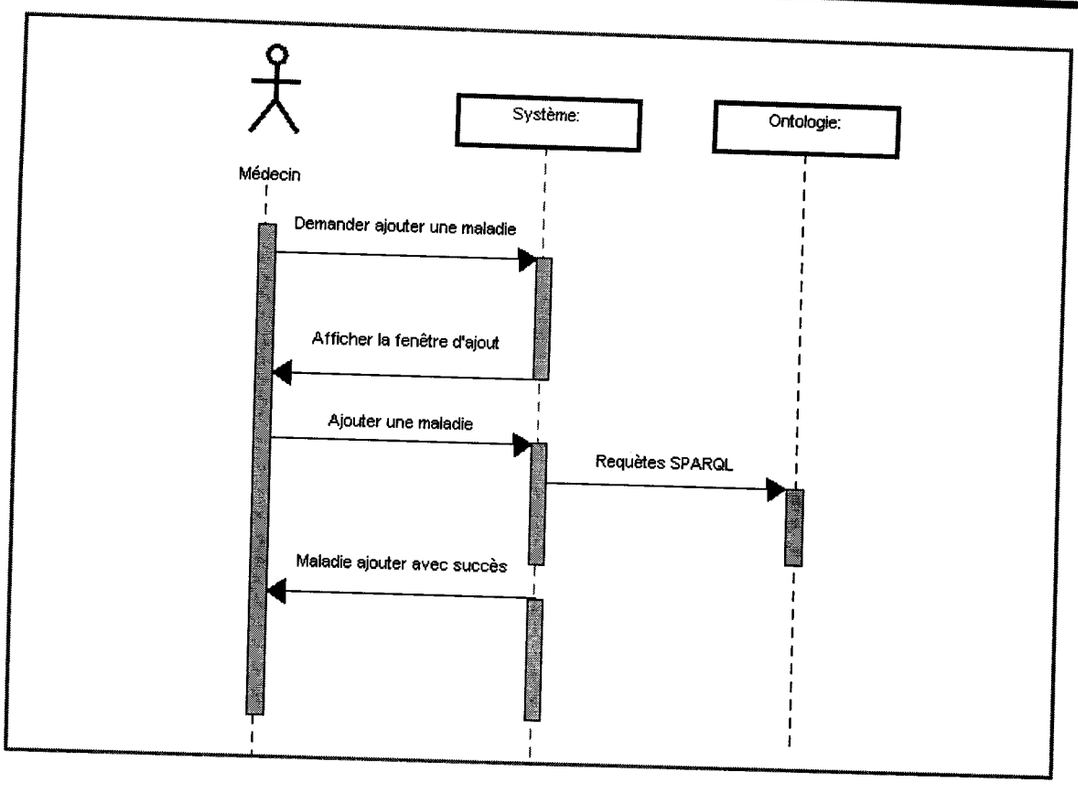


Figure 4.7 : Diagramme de séquence pour ajouter une maladie

Modifier une maladie :

Le médecin fait une demande de mise à jour d'une maladie existante. Le système lui affiche un formulaire. Le médecin choisir une maladie et remplir les nouvelles entrées dans les champs et le valide, une fenêtre de confirmation est affichée, si le médecin clique sur « oui » la maladie est modifiée, sinon la maladie n'est pas modifiée (Voir la figure 4.8).

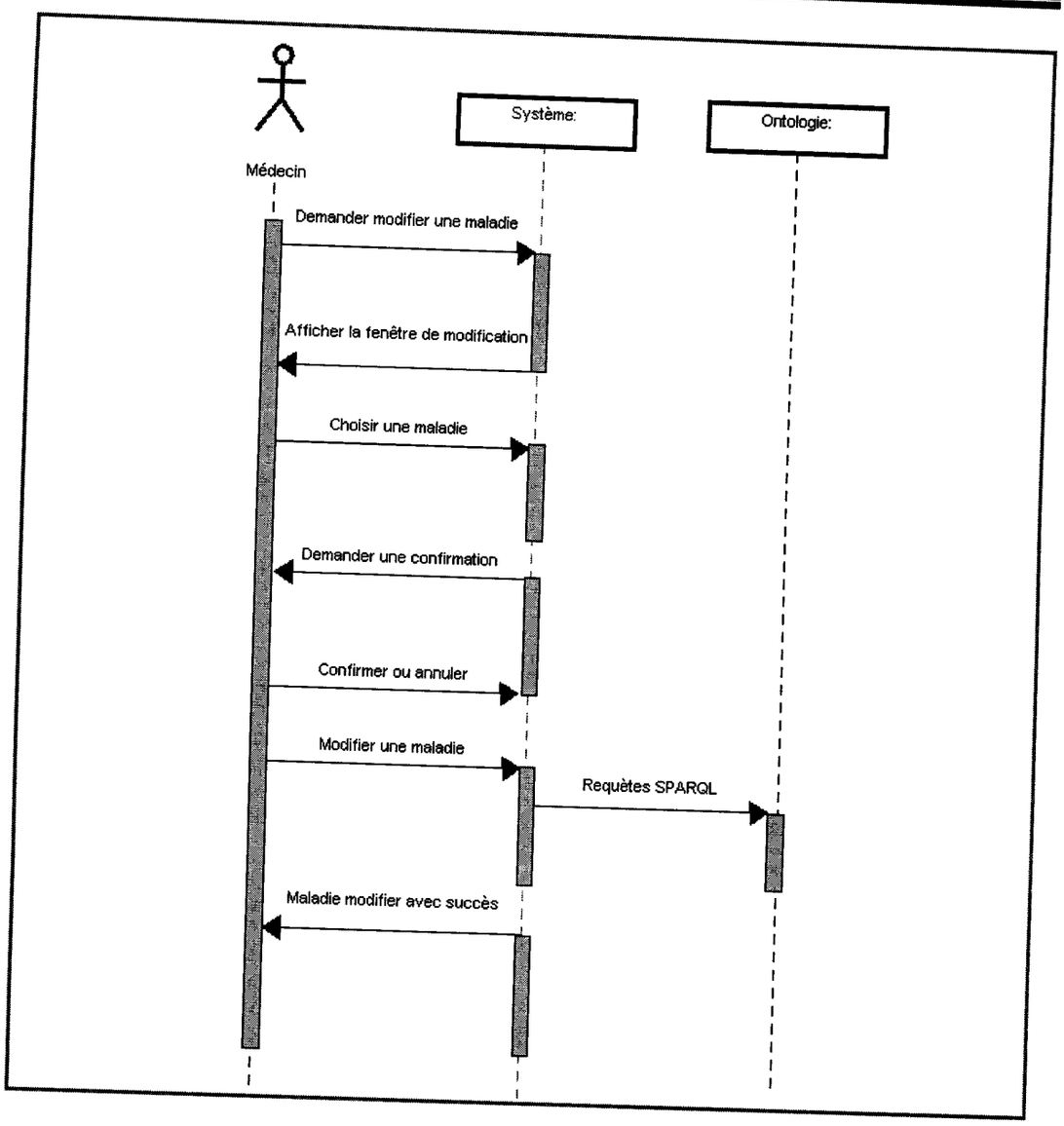


Figure 4.8 : Diagramme de séquence pour la modification d'une maladie

Supprimer une maladie :

Le médecin fait une demande de suppression d'une maladie. Le système lui affiche un formulaire. Le médecin sélectionne une maladie dans la fenêtre affiché, puis il valide, une fenêtre de confirmation est affichée, si le médecin clique sur « oui » la maladie est supprimer, sinon la maladie n'est pas supprimer (Voir la figure 4.9).

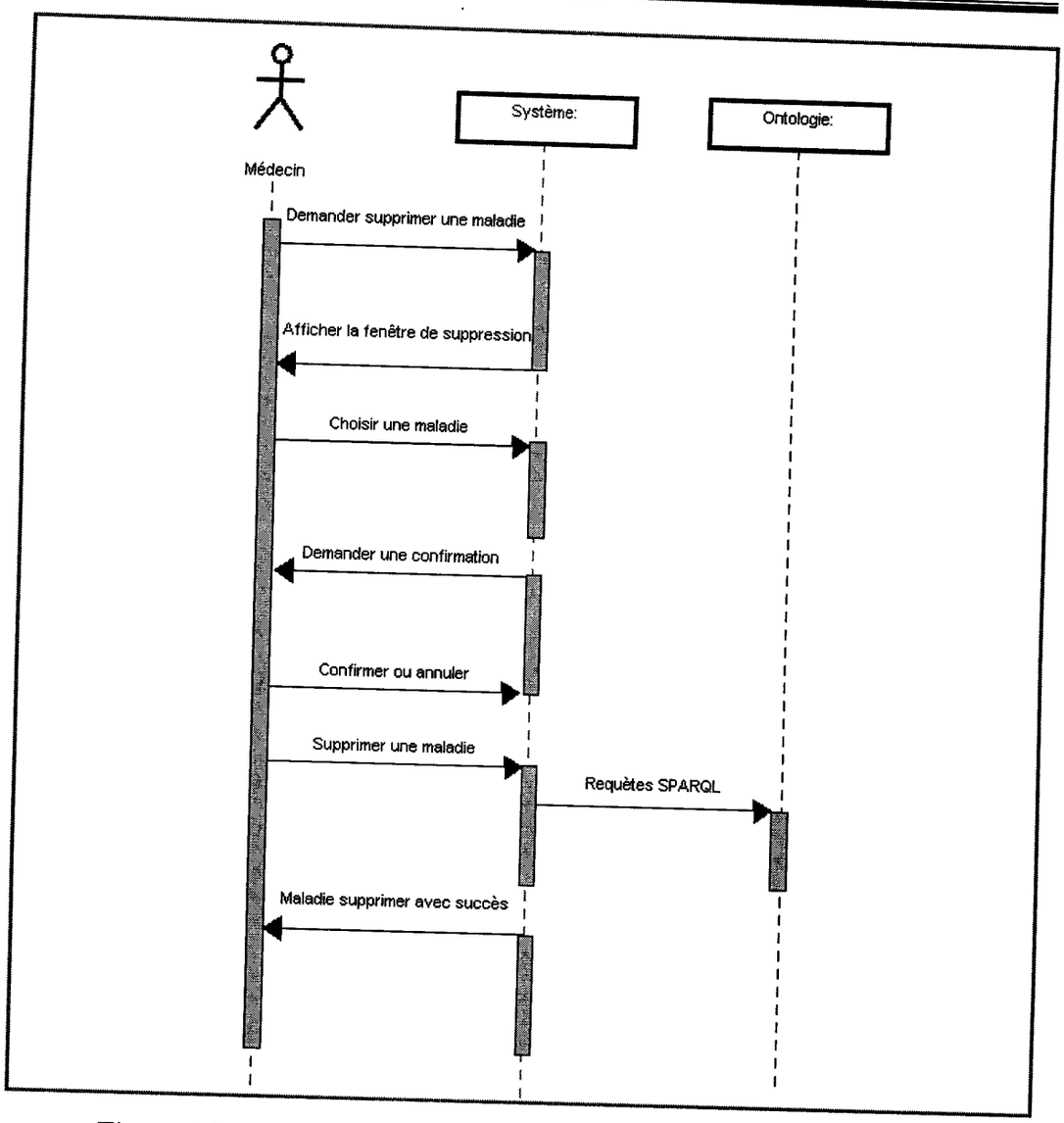


Figure 4.9 : Diagramme de séquence pour la suppression d'une maladie

IV.3 Diagramme de classe de l'application :

La figure 4.10 représente le diagramme de classes de l'application

Ce diagramme représente les différentes tâches que fait la classe Patient et Médecin.

Méthodes du patient : Le patient fait la recherche sur une maladie, ses examens, médicaments, symptômes, phénomènes et ses traitements.

On affecte pour la classe Patient la cardinalité « 0,*» et les classes Maladie, Examen, Médicaments, Symptômes, Phénomènes et Traitements ainsi la cardinalité « 0,*», signifie que plusieurs patients fait la recherche sur plusieurs maladies par exemple.

Méthodes de Médecin : Le médecin fait l'ajout, modification et suppression pour une maladie. On affecte pour la classe Médecin la cardinalité « 0,1» et pour la classe Maladie la cardinalité « 0,*», cela signifie qu'un médecin va ajouter, modifier ou supprimer plusieurs maladies.

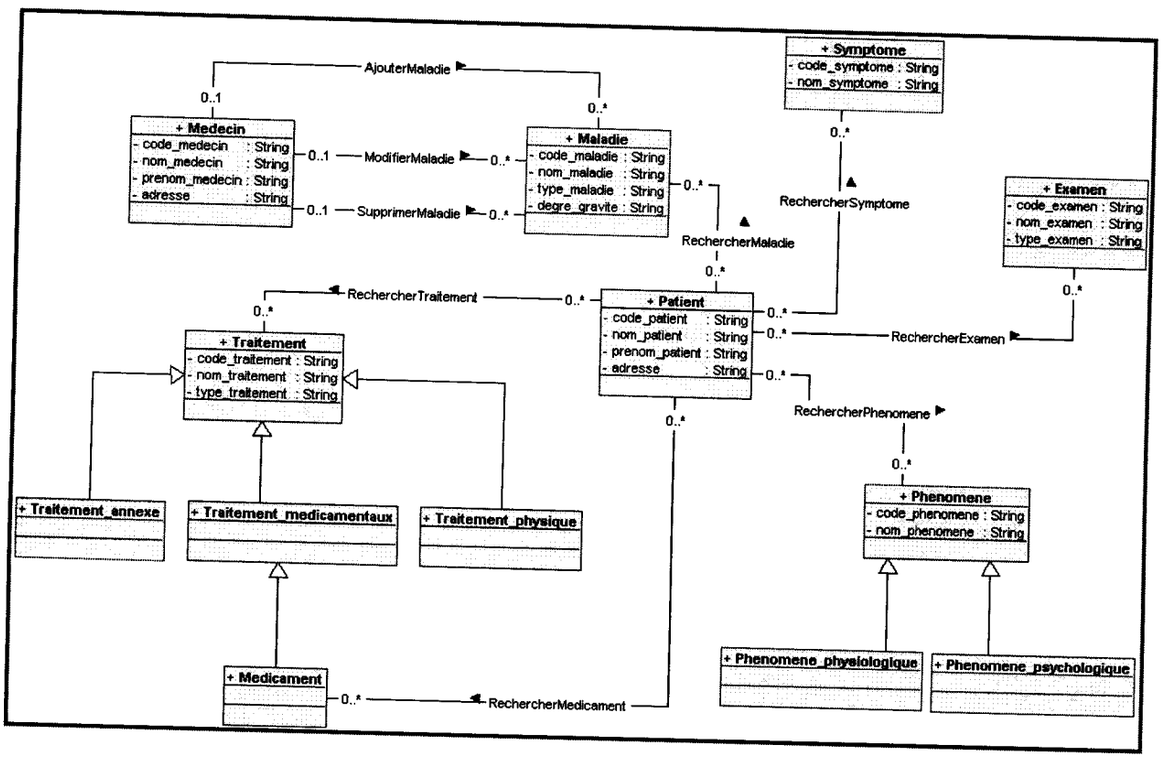


Figure 4.10 : Diagramme de classes de l'application

Conclusion :

Dans ce chapitre on a présenté les différentes phases de conception de l'ontologie médicale de ce projet.

On a présenté au niveau de la conception une liste de concepts, une liste de relations et une autre des attributs, et pour présenter les fonctionnalités des acteurs patients et médecins de notre application on utilise des diagrammes de cas d'utilisation et de séquences.

La prochaine étape consiste à rendre cette ontologie opérationnelle c'est-à-dire exploitable par un ordinateur, c'est ce qu'on présente dans le prochain chapitre.

Chapitre V

Implémentation de l'ontologie médicale

I. Introduction :

Ce chapitre d'implémentation est consacré pour notre *ontologie* ainsi que *l'application*, premièrement on va construire l'ontologie de notre application avec le langage OWL, en suite l'application qui va permettre au patient de faire la recherche d'une maladie, ces examens, médicaments, symptômes, phénomènes et traitements.

II. Outils et langages utilisés :

Parmi les outils que nous avons utilisés : *Protégé, NetBeans IDE, Jena,* et parmi les langages on utilise : *OWL, JAVA, SPARQL.*

II.1 Protégé :

Éditeur d'ontologie open source disponible à l'adresse http://protege.standford.edu , développé au département d'*Informatique Médicale* de l'*Université* de *Standford.*

Nous avons choisi cet éditeur dans sa version récent 4.1_rc4 pour éditer l'ontologie de notre application, et générer son code OWL, pour qu'on puisse l'interroger avec des requêtes SPARQL.

II.2 Java :

Jena c'est un Framework JAVA, c'est pour cette raison on choisi le langage JAVA. Il existe plusieurs IDE (*Integrated Development Environment*) pour le langage JAVA, à savoir Eclipse, JBuilder (de Borland) et NetBeans (de Sun Microsystems) qu'on a utilisé.

II.3 Jena :

Jena est un API java open source développé par un laboratoire de Hewlett-Packard permettant de lire et de manipuler des ontologies décrites en RDFS ou en OWL. Jena est disponible à http://jena.sourceforge.net/ (Voir Figure 5.1).
Pour notre projet on utilise la version 2.6.2

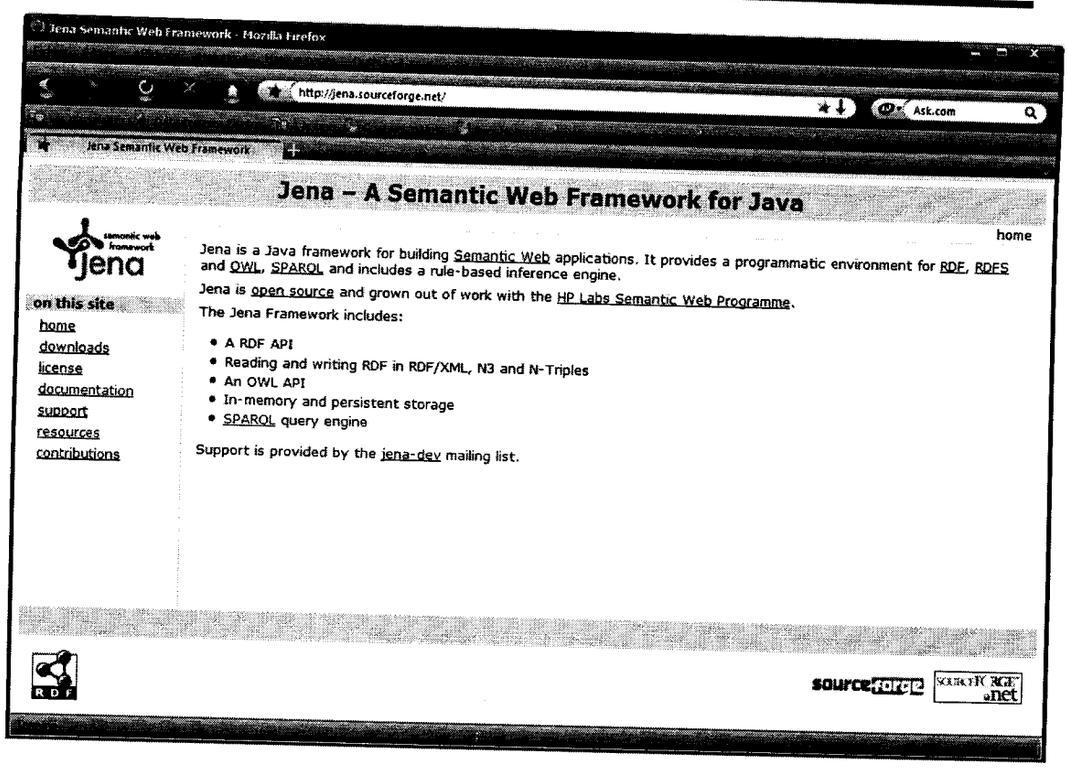


Figure 5.1 : La page d'accueil de site http://jena.sourceforge.net/

II.4 SPARQL :

Le langage SPARQL (Simple Protocol And RDF Query Language) définit la syntaxe et la sémantique nécessaires à l'expression de requêtes sur une base de données de type RDF.

SPARQL s'inspire, en partie, de la syntaxe SQL mais adaptée aux graphes de type RDF.

Dans le cadre de ce travail le langage d'interrogation SPARQL est utilisé pour effectuer des requêtes via le langage JAVA. On choisi le SPARQL pour sa facilité d'utilisation et sa très bonne intégration dans l'API Jena.

III. Construction de l'ontologie médicale :

Dans cette étape d'implémentation de notre projet on commence bien sûr par l'édition de notre ontologie médicale avec Protégé 4.1_rc4

III.1 Langage de spécification :

Le langage de spécification qu'on a choisi est OWL car dans notre projet RDFS ne suffit pas, notamment dans le coté des contraintes de cardinalité.

Exactement nous avons utilisé OWL-DL, pour certains raisons. On trouve que OWL-Lite ne permet d'exprimer que des contraintes simples de cardinalité 0 ou 1, tandis que je besoin des cardinalités multiples (0..* ou 1..*), ainsi que OWL-Full offre un plus haut niveau d'expressivité ce qu'on n'atteint pas ce niveau dans notre projet.

Tout simplement on trouve qu'OWL-DL est suffisant et convenable dans le cas de l'ontologie de notre projet.

III.2 Normalisation des noms de l'ontologie :

L'éditeur d'ontologie Protégé 4.1_rc4 qu'on va choisi comme éditeur de notre ontologie, maintient un espace de nommage unique pour les classes, les relations, les attributs et les instances (individuels).

Il est sensible à la casse, les espaces dans les noms ne sont pas permis, et les délimiteurs autorisés sont seulement « _ » et «-».

III.3 Etapes de construction de l'ontologie médicale :

Dans cette session on présente comment construit l'ontologie de notre projet afin de généré le code OWL enregistré dans le fichier d'ontologie « *Ontologie_Med.owl* » qu'on exploite après dans notre application.

III.3.a Lancement de Protégé 4.1_rc4 :

Tout d'abord on crée une nouvelle ontologie OWL (Voir Figure 5.2)
Pour la création d'un nouveau projet, on précise l'espace des noms, le langage avec lequel sera éditée l'ontologie.

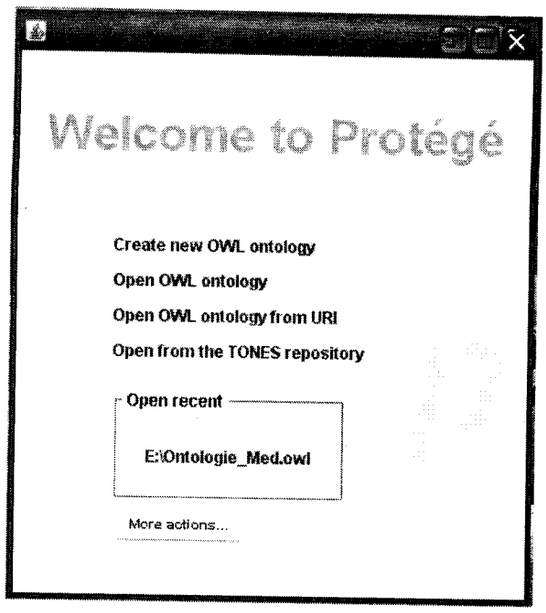


Figure 5.2 : Création d'une nouvelle ontologie OWL

Ensuite on précise l'espace de nommage (*http://localhost/ Ontologie_Med.owl*) et puis le nom de l'ontologie (*Ontologie_Med.owl*) (Voir Figure 5.3)

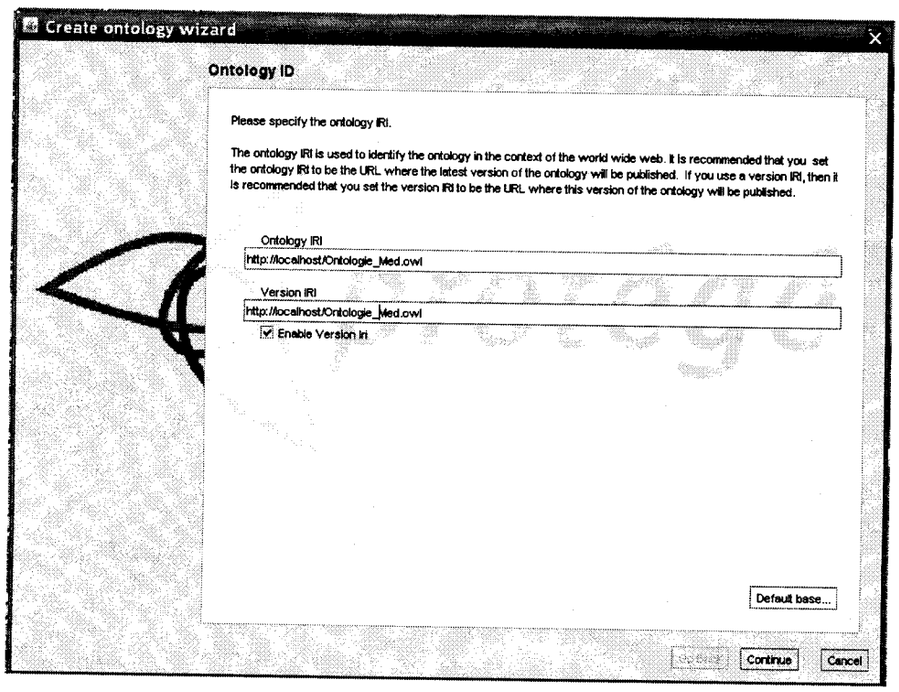


Figure 5.3 : Choix d'un espace des noms

Après la fenêtre suivante va s'afficher, pour l'édition de l'ontologie (Voir Figure 5.4) :

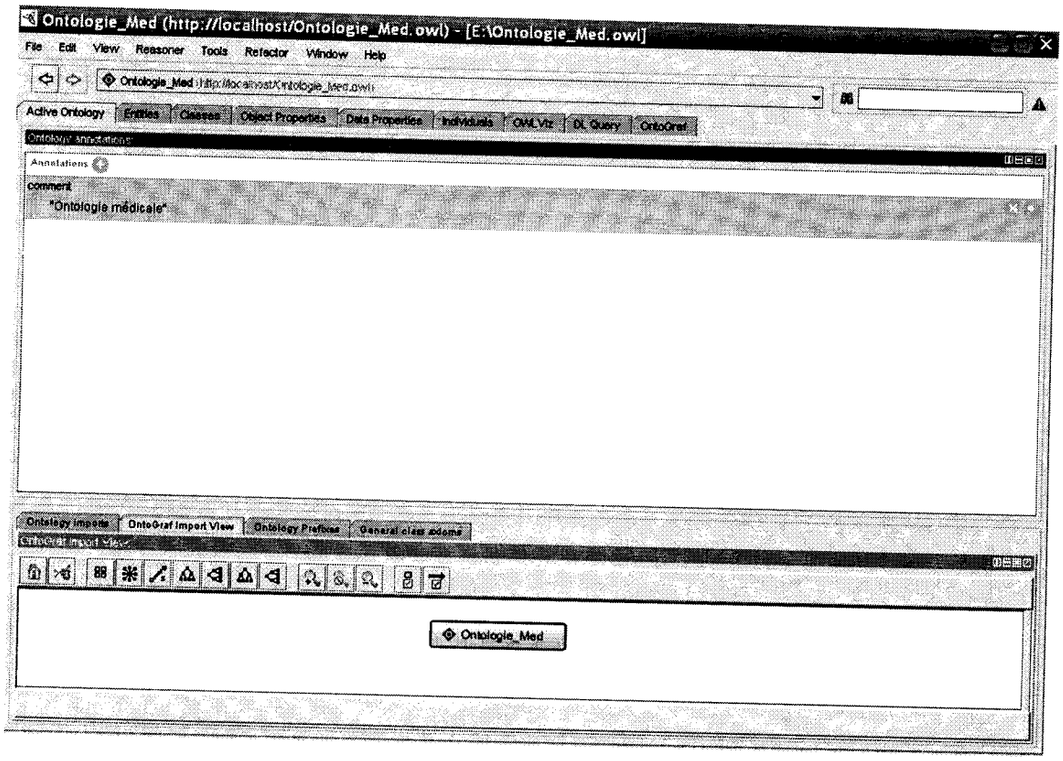


Figure 5.4 : Page d'édition de protégé 4.1_rc4

Nous Commençons par les définitions de classes, les attributs, les relations et les relations inverses tout en spécifiant les contraintes de cardinalités:

III.3.b Définition des classes :

Sur l'onglet *Classes,* on crée des classes *(concepts).*

Après la création de tous les concepts, on peut voir un aperçu de l'ontologie sous forme hiérarchique *(Classes et sous-classes)* (Voir Figure 5.5).

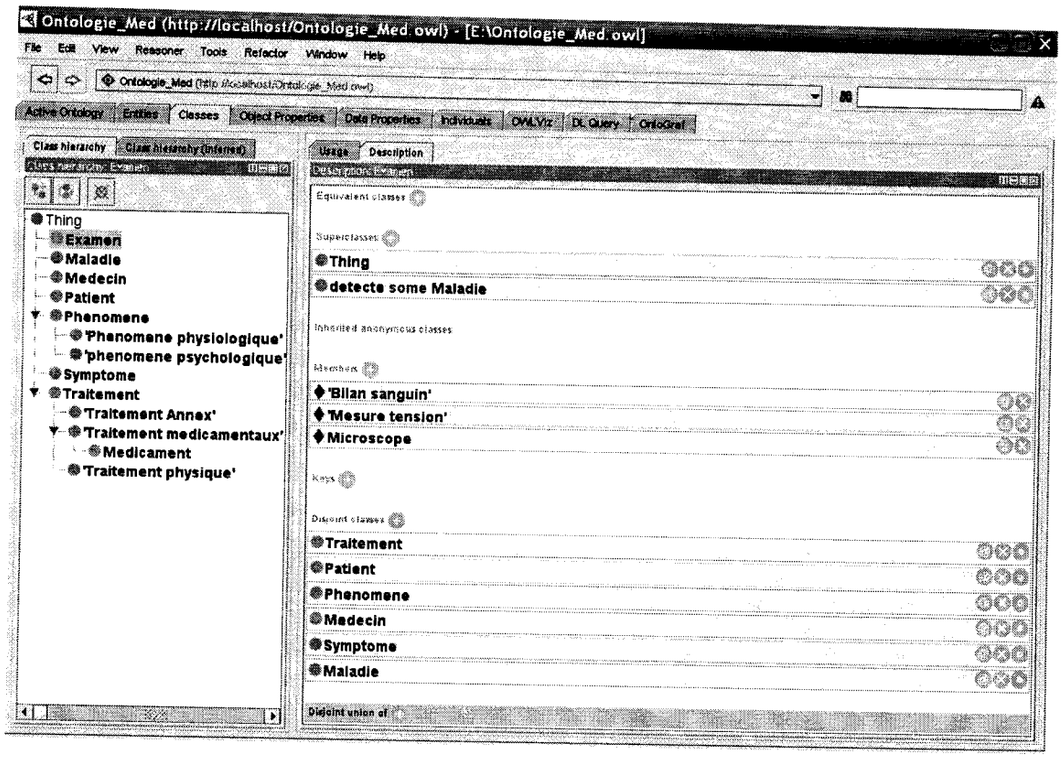


Figure 5.5 : Classes de l'ontologie médicale

III.3.c Définition des attributs :

Sur l'onglet *Data properties*, on défini des propriétés *(attributs).*

Après l'arrivé à la définition des propriétés de chaque concept en précisent également leurs type *(String, Int* par exemple*)* parmi les types prédéfinis en protégé (Voir Figure 5.6).

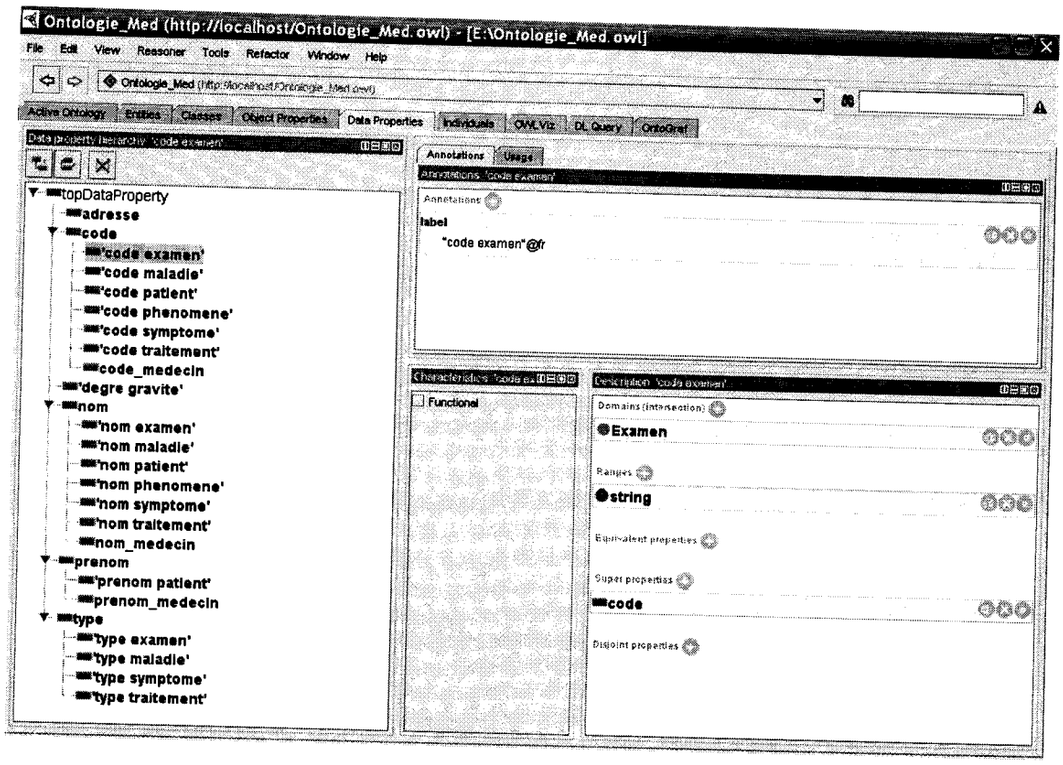


Figure 5.6 : Attributs de l'ontologie médicale

III.3.d Définition des relations :

Sur l'onglet *Object properties*, on va définir des relations *(liens).*

Après l'arrivé à la définition des relations entre les concepts tout en spécifiant les contraintes de cardinalités (Voir Figure 5.7).

Il existe également des relations inverses entre les concepts.

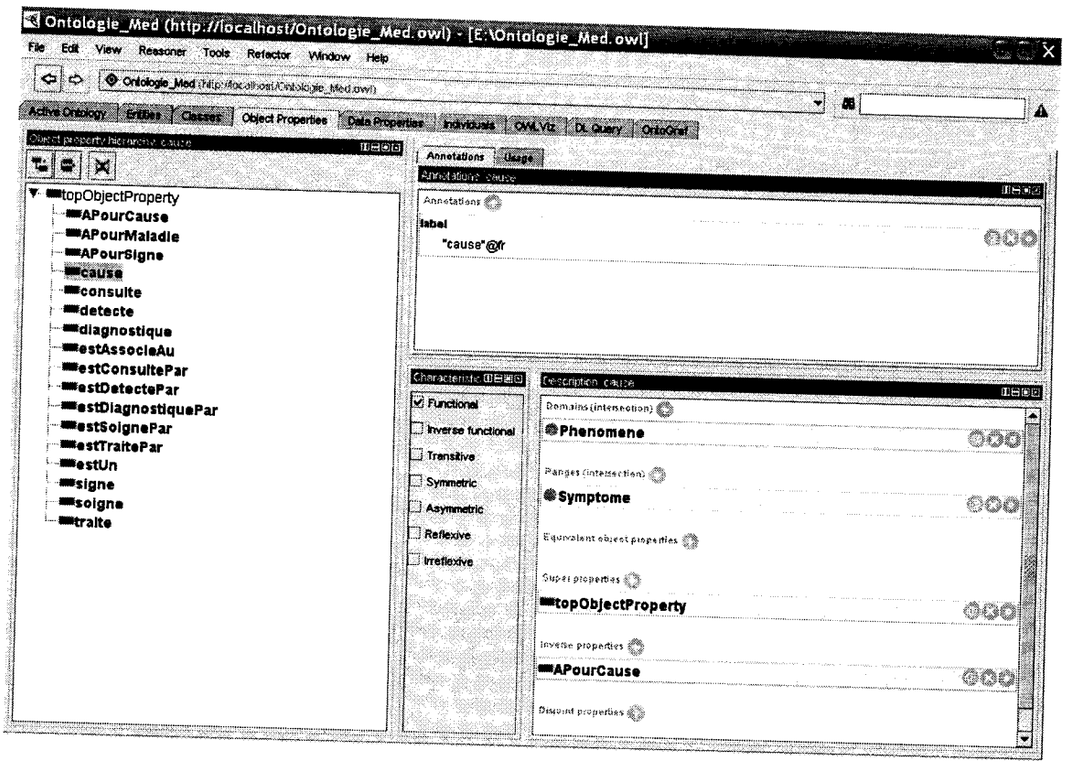


Figure 5.7 : Relations entre les concepts médicaux

III.3.e Création des instances :

Sur l'onglet *Individuals*, on va créer des instances *(individus).*

Les instances sont crées dans un espace de nommage unique *http://localhost/Ontologie _Med#,* et ils ont identifies d'une manière unique et implicite. Chaque individu est identifie par un URI (Voir Figure 5.8).

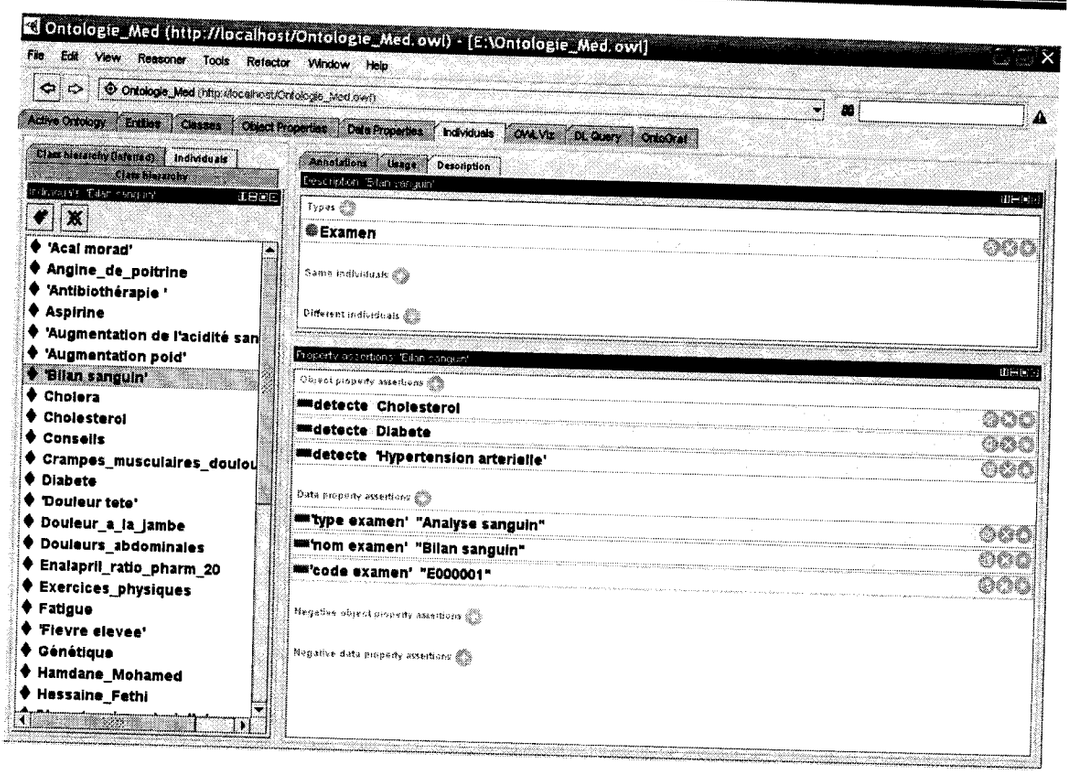


Figure 5.8: Individus de l'ontologie médicale

Après avoir achevé avec la construction de l'ontologie médicale, on la sauvegarde

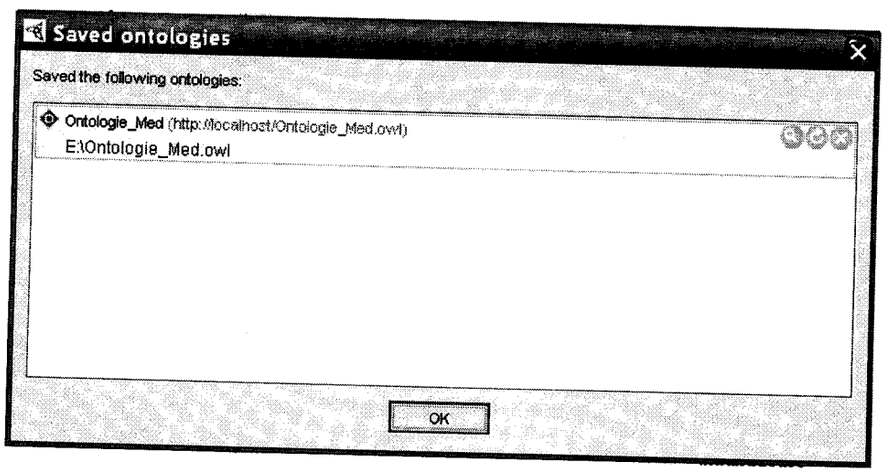


Figure 5.9 : Enregistrement de l'ontologie médicale

IV. Application :

IV.1 Application développée en JAVA :

Notre application est développée en java, et que nous l'ai nommé *Ontologie médicale.*

Cette application permettre aux patients de faire leurs recherches sur des maladies.

IV.2 Classes utilisées :

Classes	Classes mères	Attributs	Méthodes
Maladie	Thing	code_maladie	RechercherMaladie(...)
		nom_maladie	AjouterMaladie(...)
		type_maladie	ModifierMaladie(...)
		degre_gravite	SupprimerMaladie(...)
Examen	Thing	code_examen	RechercherExamen(...)
		nom_examen	
		type_examen	
Phenomene	Thing	code_phenomene	RechercherPhenomene(...)
		nom_phenomene	
Symptome	Thing	code_symptome	RechercherSymptome(...)
		nom_symptome	
		type_symptome	
Traitement	Thing	code_traitement	RechercherTraitement(...)
Medicament	Traitement	nom_traitement	RechercherMedicament(...)
		type_traitement	

Table 5.1 : Classes de bases de l'application

IV.3 Exécution de l'application :

Parmi les interfaces que notre application peut offrir, une pour le Médecin et d'autre pour le Patient : *Interface pour le Médecin* et *Interface pour le Patient.*

IV.3.a Interface pour le Patient :

Après l'accès à l'application en tant qu'un *Patient*, une interface va s'afficher pour le Patient (Voir Figure 5.10).

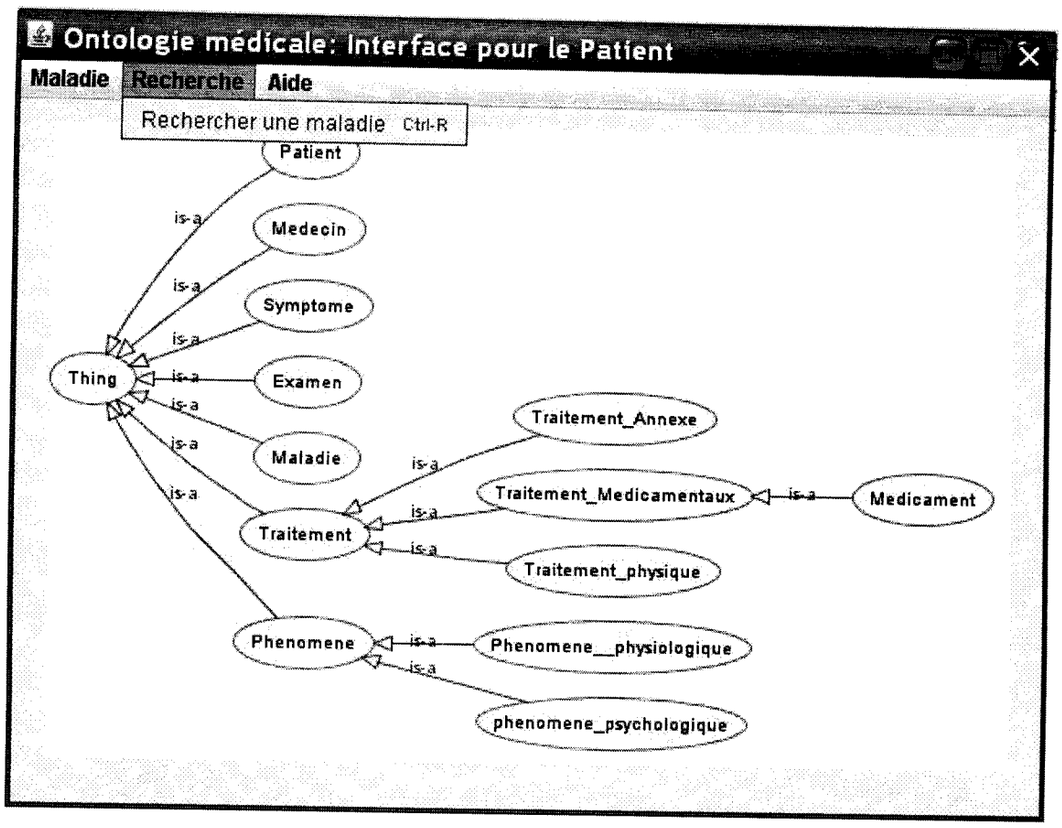


Figure 5.10 : Interface pour le Patient

Si le Patient sélectionne « Rechercher une Maladie », une fenêtre va s'afficher (voir la figure 5.11).

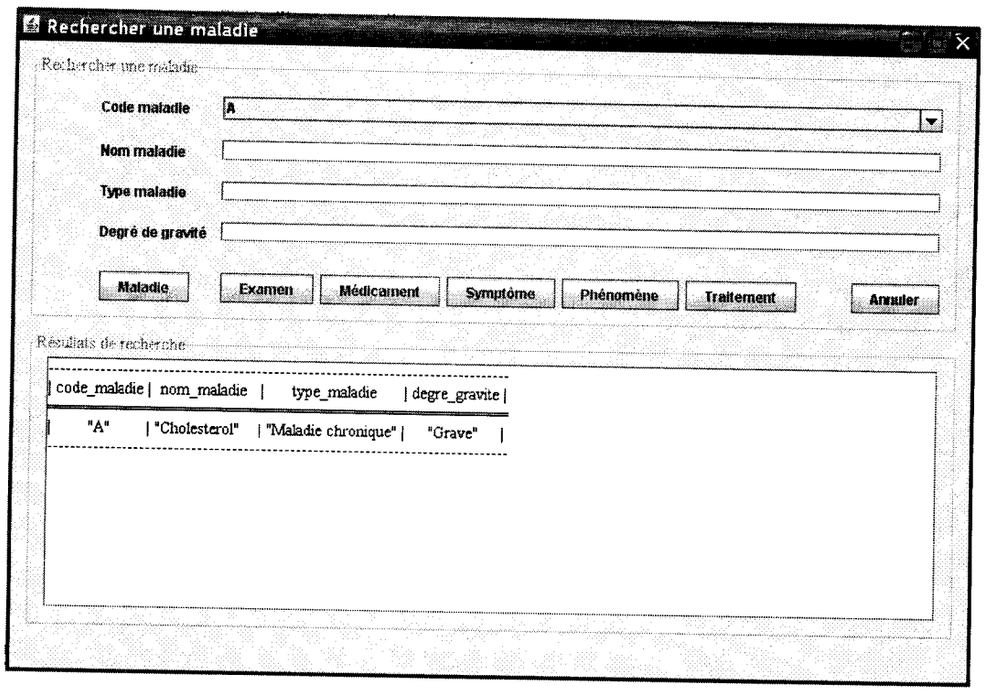


Figure 5.11 : Résultats de recherche d'une maladie de Cholestérol

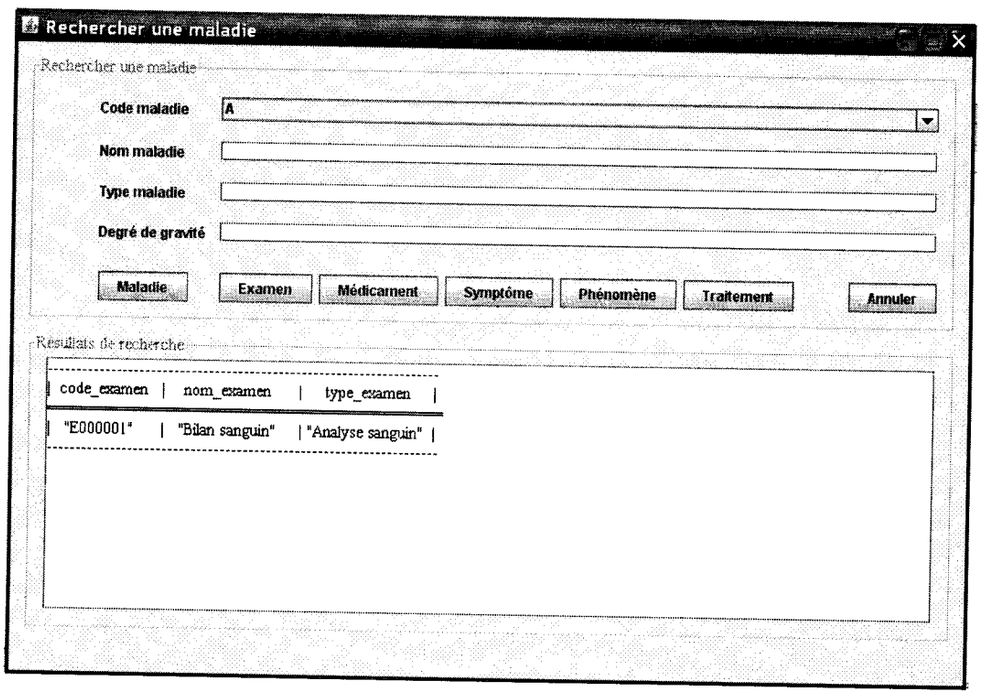


Figure 5.12 : Résultats de recherche d'un examen du cholestérol

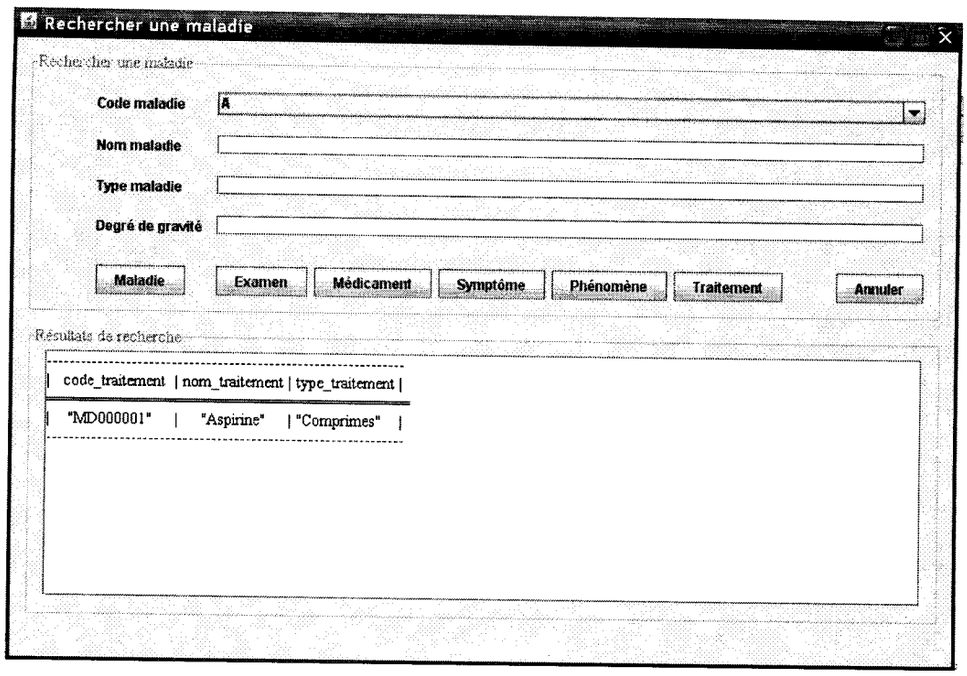


Figure 5.13 : Résultats de recherche d'un médicament du cholestérol

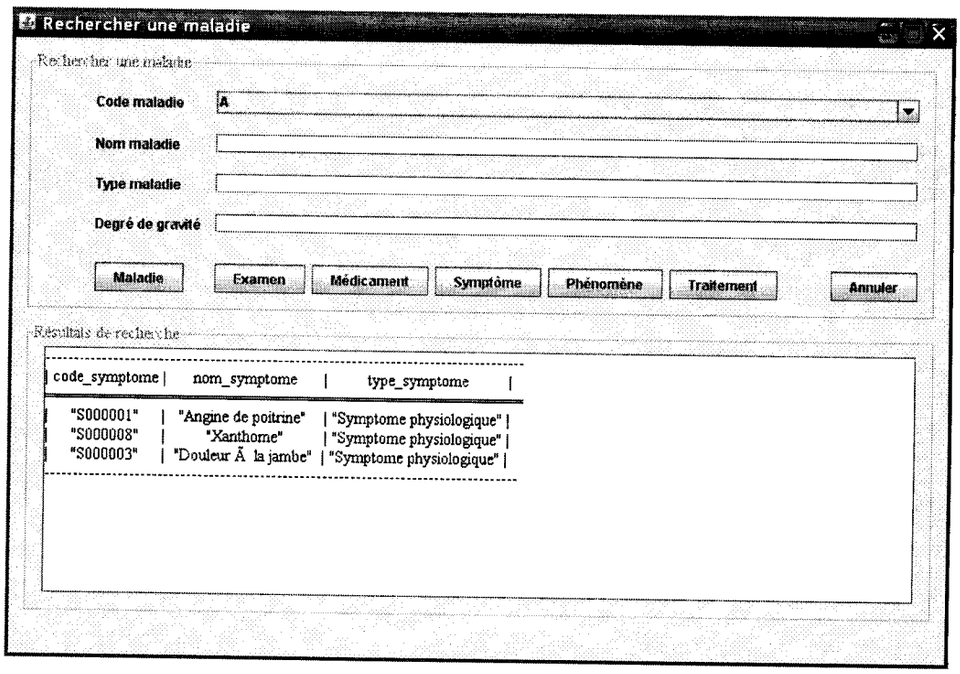


Figure 5.14 : Résultats de recherche des symptômes du cholestérol

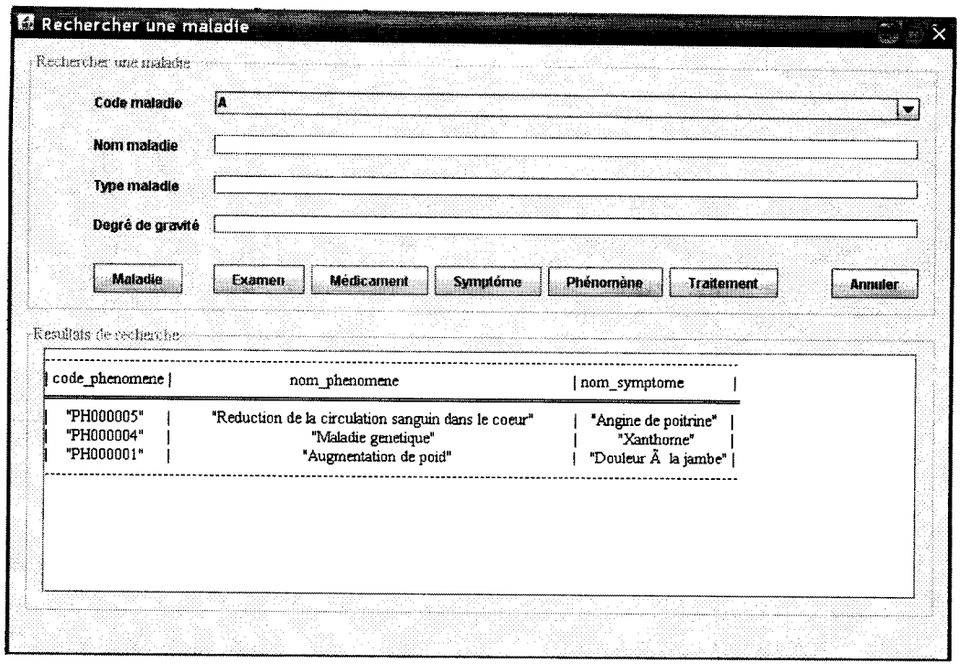


Figure 5.15 : Résultats de recherche des phénomènes du cholestérol

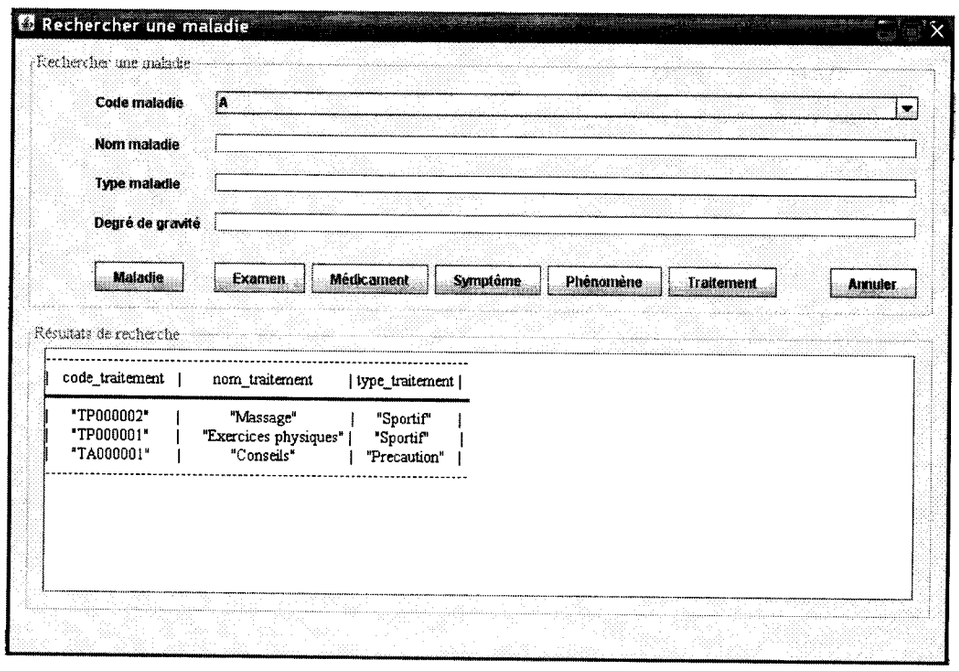


Figure 5.16 : Résultats de recherche des traitements du cholestérol

Conclusion :

Dans ce chapitre, nous avons présenté tout ce qui concerne la dernière phase d'*implémentation* de notre système (ontologie et application). Nous avons également défini les différentes outils et langages de développement que nous avons utilisé, puis nous avons présentés les principales classes développées, les attributs et les méthodes qui les constituent.

Enfin nous avons montré l'interface Patient et les résultats de recherche d'une maladie fournie par cette interface.

Conclusion générale

Dans ce projet, nous avons abordé la problématique de l'accès à l'information précise et plus spécialement à la connaissance médicale. Nous nous sommes intéressés plus particulièrement aux systèmes de question-réponse, qui visent à retourner une réponse précise à un besoin d'information exprimé en langage naturel.

Cette étude nous amené de construire une ontologie médicale avec l'outil Protégé et l'interroge par des requêtes SPARQL en utilisons le langage de programmation Java.

Nous avons tout d'abord étudié des systèmes de question réponse. En premier lieu, nous avons présenté l'architecture d'un tel type de systèmes ainsi que les différents modules intervenant dans la chaîne de traitement. Puis, nous avons présenté le principe de fonctionnement d'un système question réponse dans le domaine médicale.

Ensuite nous avons introduire l'utilité des ressources ontologiques et sémantiques pour structurer les bases de connaissances d'un système de question réponse.

A cause de l'utilité de l'ontologie de construire ses ressources, nous avons introduire un chapitre concernant l'ontologie, son composition et ces langages.

Les perspectives de notre travail est l'enrichissement de l'ontologie médicale et développer un système de question réponse qui utilise cette base de connaissance structurée.

Annexes

1. Composantes d'une ontologie OWL :

Les principaux éléments constituant une ontologie OWL sont :

a. Espaces de nommage : Afin de pouvoir employer des termes dans une ontologie, il est nécessaire d'indiquer avec précision de quels vocabulaires ces termes proviennent. C'est la raison pour laquelle, comme tout autre document XML, une ontologie commence par une déclaration d'espace de nom (parfois appelée « de nommage ») contenue dans une balise rdf:RDF. Supposons que nous souhaitons écrire une ontologie sur une population de personnes ou, d'une manière plus générale, sur l'humanité. Voici la déclaration d'espace de nom qui pourrait être employée :

```
<rdf:RDF
  xmlns = "http://domain.tld/path/humanite#"
  xmlns:humanite= "http://domain.tld/path/humanite#"
  xmlns:base = "http://domain.tld/path/humanite#"
  xmlns:vivant = "http://otherdomain.tld/otherpath/vivant#"
  xmlns:owl = "http://www.w3.org/2002/07/owl#"
  xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs = "http://www.w3.org/2000/01/rdf-schema#"
  xmlns:xsd = "http://www.w3.org/2001/XMLSchema#">
```

Les deux premières déclarations identifient l'espace de nommage propre à l'ontologie que nous sommes en train d'écrire. La première déclaration d'espace de nom indique à quelle ontologie se rapporter en cas d'utilisation de noms sans préfixe dans la suite de l'ontologie. La troisième déclaration identifie l'URI de base de l'ontologie courante.

La quatrième déclaration signifie simplement que, au cours de la rédaction de l'ontologie humanité, on va employer des concepts développés dans une ontologie vivant, qui décrit ce qu'est un être vivant.

Les quatre dernières déclarations introduisent le vocabulaire d'OWL et les objets définis dans l'espace de nommage de RDF, du schéma RDF et des types de données du Schéma XML.

Afin de simplifier l'écriture des URI dans la déclaration d'espace de nom et, surtout, dans les valeurs des attributs de l'ontologie, il est conseillé de définir des abréviations au moyen d'entités du type de document :

```
<!DOCTYPE rdf:RDF [
  <!ENTITY humanite "http://domain.tld/path/humanite#" >
  <!ENTITY vivant "http://otherdomain.tld/otherpath/vivant#" >
]>
```

Ainsi, la déclaration d'espace de nom initiale devient :

```
<rdf:RDF
  xmlns = "&humanite;"
  xmlns:humanite= "&humanite;"
  xmlns:base = "&humanite;"
```

```
  xmlns:vivant = "&vivant;"
  xmlns:owl = "http://www.w3.org/2002/07/owl#"
  xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs = "http://www.w3.org/2000/01/rdf-schema#"
  xmlns:xsd = "http://www.w3.org/2001/XMLSchema#">
```

b. En-têtes d'une ontologie :

Tout comme il existe une section d'en-tête <head>..</head> en haut de tout document XHTML bien formé, on peut écrire, à la suite de la déclaration d'espaces de nom, un en-tête décrivant le contenu de l'ontologie courante. C'est la balise owl:Ontology qui permet d'indiquer ces informations :

```
<owl:Ontology rdf:about="">
  <rdfs:comment>Ontologie décrivant l'humanité</rdfs:comment>
  <owl:imports
      rdf:resource="http://otherdomain.tld/otherpath/vivant"/>
  <rdfs:label>Ontologie sur l'humanité</rdfs:label>
</owl:Ontology>
```

2. Eléments du langage OWL:

Les éléments composant du langage d'OWL sont :

A. Classes :

Une classe définit un groupe d'individus qui sont réunis parce qu'ils ont des caractéristiques similaires. L'ensemble des individus d'une classe est désigné par le terme « extension de classe », chacun de ces individus étant alors une « instance » de la classe. Les trois versions d'OWL comportent les mêmes mécanismes de classe, à ceci près qu'OWL FULL est la seule version à permettre qu'une classe soit l'instance d'une autre classe. A l'inverse, OWL Lite et OWL DL n'autorisent pas qu'une instance de classe soit elle-même une classe.

a. Déclaration d'une classe : La description de la classe se fait directement par le nommage de cette classe. Une classe « humain » se déclare de la manière suivante :

```
<owl:Class rdf:ID="Humain" />
```

b. Héritage : Il existe dans toute ontologie OWL une superclasse, nommée Thing, dont toutes les autres classes sont des sous-classes. Ceci nous amène directement au concept d'héritage, disponible à l'aide de la propriété subClassOf :

```
<owl:Class rdf:ID="Humain">
  <rdfs:subClassOf rdf:resource="#EtreVivant" />
</owl:Class>
```

Enfin, il existe également une classe nommée noThing, qui est sous-classe de toutes les classes OWL. Cette classe ne peut avoir aucune instance.

B. Instances de classe :

Un individu consiste à énoncer un « fait », encore appelé « axiome d'individu ».

SPARQL est un langage de requêtes pour l'interrogation de métadonnées et l'extraction des données sous forme d'un graphe RDF ou plus exactement un langage d'interrogation de triplets RDF. Un triplet RDF est une association: {sujet, prédicat et objet}.

Syntaxe SPARQL :

La syntaxe des requêtes SPARQL basées sur le motif de graphe est composée généralement de :

1. La clause SELECT : les variables qui doivent apparaître dans les résultats.
2. La clause WHERE : les motifs de graphe élémentaire.
3. La clause OPTIONAL : les motifs du graphe élémentaire optionnels.
4. La clause FILTER : restriction de résultat par des expressions régulières (test).
5. La clause ORDER BY : ordonner le résultat d'une manière ascendante ou descendante.

a) Exemple d'une requête simple:

L'exemple ci-dessous montre une interrogation SPARQL pour trouver tous les courriers et leurs destinataires.

Le nom des variables est précédé toujours par « ? ».

```
SELECT DISTINCT ?x ?y
WHERE
{ ?x rdf : type courrier
?y rdf: type destinataires }
ORDER BY ?date_courrier
```

Résultat d'interrogation :

Courrier	Destinataires
Courrier14582	W25

DISTINCT est introduit dans cette requête pour éliminer les solutions en double pour ne pas retourner plusieurs fois les mêmes valeurs.

ORDER BY est introduit dans cette requête pour ordonner les résultats obtenus d'une manière ascendante par ASC( ) ou descendante par DESC( ).

b) Exemple des filtres:

La clause FILTER du langage SPARQL restreint les solutions à celles que l'expression de filtre est évaluée vrai. L'exemple ci-dessous impose des contraintes sur la variable destinataire sur l'interrogation SPARQL pour trouver tous les courriers et leurs destinataires

SELECT ?x ?y

WHERE

{ ?x rdf :type courrier

?y rdf:type destinataires

FILTER (destinataire= « W25») }

Résultat est la même que la précédente.

c) Exemple de OPTIONAL:

Le graphe RDF n'assure pas la présence de structures complètes et régulières pour toutes les données. Il est parfois nécessaire de ne pas rejeter la solution parce que des parties du motif d'interrogation ne correspondent pas. Cette fonctionnalité est assurée par la clause OPTIONAL.

Nous pouvons, aussi, avoir des contraintes (FILTER) dans un motif de graphe optionnel.

L'exemple ci-dessous montre qu'il existe des motifs de graphe élémentaires qui sont optionnelles pour trouver tous les courriers et leurs destinataires et commentaires dans le graphe de données correspondant.

SELECT ?x ?y ?commentaire

WHERE

{ ?x rdf :type courrier

?y rdf:type destinataires

OPTIONAL { ?courrier c :A ?commentaire}

FILTER (destinataire= « W25 ») }

Résultat d'interrogation :

Courrier	Destinataires	Commentaire
Courrier14582	W25	

d) Exemple de UNION:

Le langage de requête SPARQL assure la combinaison des motifs de graphe, par le mot clé UNION, d'une manière que l'un des motifs de graphe alternatifs peut correspondre. Chaque alternative de la requête peut contenir plusieurs motifs de triplet.

L'exemple ci-dessous montre comment concaténer les solutions de plusieurs requêtes

SELECT ?title

WHERE {{ ?book dca :title ?title} UNION{ ?book dcb:title ?title }}

Résultat d'interrogation :

Title
Courrier14582

e) Exemple de LIMIT et OFFSET:

OFFSET fait commencer les solutions générées après le nombre indiqué de solutions. LIMIT met une limite supérieure au nombre de solutions retournées. Si le nombre de solutions réel est supérieur à la limite, alors le nombre limite de solutions, au plus, sera retourné.

L'utilisation de LIMIT et OFFSET, pour sélectionner des sous ensembles différents parmi les solutions d'interrogation, n'aura pas d'utilité sans la clause ORDER BY.

SELECT ?name

WHERE { ?x foaf :name ?name}

ORDER BY ?name

LIMIT 3

OFFSET 5

f) Exemple de ASK:

ASK retourne un booléen indiquant si un motif d'interrogation a une solution ou non.

L'exemple ci-dessous vérifie la correspondance de la variable courrier dans le graphe RDF

ASK { ?x f :name « alice »}

Réponse : No

g) Exemple de CONSTRUCT:

La forme d'interrogation CONSTRUCT retourne un sous graphe RDF à partir du graphe RDF. Ce sous graphe est généré par la construction d'un ensemble des triplets concordant avec le motif de graphe de l'interrogation.

L'exemple ci-dessous montre comment construire un sous graphe à partir du graphe RDF.

CONSTRUCT { ?x c:objet ?objet}

WHERE { ?x courrier :objetcourrier ?objet}

Résultat de construction :

c : courrier c : objet «visite présidentielle »

Le résultat de l'application de CONSTRUCT est un graphe, par contre l'application de SELECT retourne des liaisons de variables sous forme tabulaire.

Références bibliographiques

[1] Mr. Mehdi EMBAREK, « Un système de question-réponse dans le domaine médical-Le système Esculape », thèse de doctorat, université de Paris-Est, 2008.

[2] M^elle^ Asma Ben Abacha, « Questions-réponses dans le domaine médical : une approche
Sémantique », article, France, 2009.

[3] M^elle^ Anne-Laure Ligozat, « Système de Question Réponse : apport de l'analyse syntaxique lors de l'extraction de la réponse », article, Université Paris Sud Orsay, 2006.

[4] M^elle^ Souheila KHALFI, « Construction d'une ontologie pour la prise en charge des patients à domicile », thèse de magister, Université Mentouri Constantine, 2009.

[5] Mr. Anne-Laure Ligozat, « Exploitation et fusion de connaissances locales pour la recherche d'informations précises », thèse de doctorat, Université Paris XI – Orsay, 2006.

[6] Mr. Xavier Lacot, « Introduction à OWL-un langage XML d'ontologies Web », article, 2005.

[7] Mr. Djaghloul Younes, « Intégration des ressources Web dans un environnement P2P, basée sur les ontologies et la gestion de la confiance », thèse doctorat en informatique, Universite Mentouri Constantine, 2007.

[8] M^elle^ Hasna Boumechaal, « Conversion des requêtes en langage naturel vers nRQL », thèse magister en informatique, Ecole Doctorale en Informatique de l'Est Pôle de Constantine, 2010.

[9] Mr. Boucetta Zouhel, « Appariement semantique des cvs/offres d'emploi dans le cadre du e-recrutement », thèse magister en informatique, Universite Mentouri Constantine, 2008.

[10] Mr. Ian Horrocks, « DAML+OIL a Reason-able Web Ontology Language », article, University of Manchester Oxford Road, 2003.

[11] Mr. Samer Abdul Ghafour, « Méthodes et outils pour l'intégration des ontologies », mémoire de stage de DEA, Laboratoire d'Informatique en Images et Systèmes d'information, 2004.

[12] M^elle BOUGCHICHE Lilia, « Vers une ontologie pour le dispositif d'interaction », thèse magister en informatique, Ecole Nationale Supérieure d'Informatique E.S.I : Oued-Smar, Alger, 2007.

[13] Gayo Diallo, « Une Architecture à Base d'Ontologies pour la Gestion Unifiées des Données Structurées et non Structurées », thèse doctorat, Université Joseph Fourier – Grenoble I, 2006

[14] M^r Boudaoud Mohammed El Amin, « Entrepôt de ressources pédagogiques », Thèse Ingénieur d'état, Université Abou Bekr Belkaid Tlemcen, 2010

[15] M^r Eric Prud'hommeaux et Andy Seaborne, « Langage d'interrogation SPARQL pour RDF », article, Recommandation du W3C, 15 janvier 2008

[16] M^r Francis LAPIQUE, « Le langage d'ontologie Web OWL », article, Recommandation du W3C, octobre 2006

[17] M^r Brian McBride, « An Introduction to RDF and the Jena RDF API », article, Recommandation du W3C, 2010

[18] M^r Ian Dickinson, « Jena Ontology API », article, Recommandation du W3C, 2009

[19] M^r Dipsy Kapoor ,« Jena », article, International Hellenic university, 2010

[20] M^r Tomi Kauppinen, « Jena RDF API », cours, University of Helsinki (Department of Computer Science), 2006

[20] Philip McCarthy, « Introduction to Jena », article, Recommandation du W3C ,23 juin 2004

[21] M^r Francis Lapique, « SPARQL langage et protocole d'interrogation de métadonnées », article, Recommandation du W3C, 2010

[22] M^r Matthew Horridge, Holger Knublauch, « A Practical Guide To Building OWL Ontologies Using The Protégé-OWL Plugin and CO-ODE Tools », article, University Of Manchester, 27 Aout 2004

[23] M^r Matthias Hert, « Semantic Web Engineering », cours, University of Zurich (Departement of Informatics), 2009