

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de L'Enseignement Supérieur et de la Recherche Scientifique



N°Réf:.....

Centre Universitaire

Abd elhafid boussouf mila

Institut des Sciences et de la Technologie Département de Mathématiques et Informatique

**Mémoire préparé En vue de l'obtention du diplôme de
Master**

Spécialité Sciences et Technologies de l'Information et de La communication stic

**Conception d'un système de gestion
d'ontologie**

Préparé par :

-Mehazzem Wassim

- Kara Mohamed Fouad

Soutenu devant le jury

Président: Boumassata Meriem.....M.A.B

Examineur: Khalfi Souheila.....M.A.B

Encadrer: Deffas ZinebM.A.A

Année universitaire 2014/2015

Remerciement

Nous Remercions En tout premier lieu ALLAH le tout puissant, qui m'a donné la force, la volonté et le courage pour accomplir ce modeste travail.

Nous tenons ici à remercier Mme Deffas Zineb, notre encadreur pour son aide et sa grande patience qu'elle a apporté tout au long la préparation de ce mémoire.

Nous tenons à exprimer nos sincères remerciements Aux membres du jury pour l'intérêt qui 'ils ont parlé à notre recherche en acceptant d'examiner notre travaille et de l'enrichir par leurs parasitions.

Nous n'oublions pas non plus nos enseignants qui tout au long du cycle d'étude au centre universitaire de Mila, nous ont transmis leur savoir.

Nous tenons enfin à remercier tous ceux qui ont collaborés de près ou de loin à l'élaboration de ce travail. Qu'ils acceptent nos humbles remerciements.



Dédicaces

Je dédie ce modeste travail :

A ma chère maman Bariza que j'aime le plus au monde qui ma donnée le courage et l'espoir pour atteindre mon but, et mon père qui est toujours dans mon cœur (Allah yrahmo)

A mon frère Imad qui j'estime beaucoup.

À Toute ma famille

A mon binôme WASSIM.

À Tous mes amis

et mes collègues de centre universitaire,

Hakim, Chouaib, Kamel, Djaber, Zaid, Brahim et Meryem

A tous mes enseignants.

À Tous ceux qui me sont chers.

A tout Personne qui me connaît.

Fouad

Dédicaces

Avec un énorme plaisir, un cœur ouvert et une immense joie,

Que Je dédie ce travail à ma très chère mère Fatima.

Tu es l'exemple de dévouement qui n'a pas cessé de m'encourager et de

Prier pour moi. Puisse Dieu, le tout puissant, te préserver et t'accorder

Une longue vie, pleine de bonheur et de santé.

À mon père Houcine.

Rien au monde ne vaut les efforts fournis jour et nuit pour mon

Éducation et mon bien être.

Ce travail est le fruit de tes sacrifices que tu as consentis pour mon

Éducation et ma formation.

A mes frères et sœurs.

Mohammed Amine, et Abir.

Vous avez toujours été là à n'importe quel moment où j'avais besoin de

vous. Je vous remercie infiniment.

À mes oncles, tantes, cousin et cousines.

À mon ami, collègue, et binôme Fouad qui a partagé avec moi les

Moments difficiles de ce travail et à sa famille.

À tous mes amis, surtout Hakim, Choib, Zaid,oussama,Bouba

Kamal et finalement à ma chère Romaissa

WASSIM

Tables des matières

Introduction générale

1. Contexte générale	1
2. Contribution	1
3. Organisation de mémoire	2

Chapitre 1 Les ontologies

1. Introduction.....	4
2. Historique.....	4
3. Notion d'ontologie	5
4. Définition.....	5
5. utilisation des ontologies	6
5.1. Communication.....	6
5.2. Interopérabilité	7
5.3. Ingénierie des systèmes.....	7
6. Le cycle de vie des ontologies	8
7. Les types des ontologies	8
8. Les éléments constituant une ontologie.....	10
8.1. Les concepts.....	10
8.2. Relations.....	11
8.3. Les instances	11
9. Construction d'une ontologie	12
9.1. Un squelette de méthodologie pour construire des ontologies.....	12
9.1.1 Evaluation des besoins	12
9.1.2. Conceptualisation.....	13
9.1.3. Ontologisation.....	13

9.1.4. Opérationnalisation.....	14
9.2. Quelques méthodologies de construction d'ontologies.....	14
9.2.1 Tove.....	14
9.2.2 Enterprise.....	15
9.2.3. Methontology.....	15
10. Formalisme de représentation des Ontologies.....	17
10.1. Les graphes conceptuels.....	17
10.2. Logiques de description.....	17
11. Langages et Outil de représentation d'ontologie.....	18
11.1. Langages de représentation d'ontologie.....	18
11.1.1. RDF (Resource Description Framework).....	18
11.1.2. RDFS (Resource Description Framework Schéma).....	18
11.1.3. OWL (Ontology Web Langage).....	18
11.2. Outils de représentation d'ontologie.....	19
11.2.1 .Protégé.....	19
12. Moteurs d'inférences.....	20
12.1. Racer.....	20
12.2. Pellet.....	21
13. Conclusion.....	21

Chapitre 2 Construction de l'ontologie diabète

1. Introduction.....	22
2. Processus de construction d'une ontologie OWL.....	22
2.1. Spécification des besoins.....	22
2.2. Conceptualisation.....	23
2.3. Formalisation.....	24

2.4. Implémentation.....	25
2.5. Evolution et vérification	25
3. Construction de l'ontologie Diabète dans le domaine médical	26
3.1. Le Diabète: domaine d'application	26
3.2. Les grands types de la maladie	26
3.2.1. Le diabète de type 1	26
3.2.2. Le diabète de type 2	27
3.2.3. Le diabète de type 3	27
3.2.4. Le diabète gestationnel	27
3.3. Etape de spécification	27
3.4. Etape de conceptualisation.....	27
3.4.1. Construction du glossaire de termes	28
3.4.2. Hiérarchies de concepts	32
3.4.3. Dictionnaire de concepts(DC).....	33
3.4.4. Tableau de relations binaires	35
3.4.5. Tableau des attributs.....	36
3.4.6. Tableau des axiomes logiques.....	37
3.4.7. Tableau des instances	39
3.5. Etape de formalisation.....	40
3.5.1 Construction de TBox	40
3.5.2. Construction de ABox.....	42
3.6. Implémentation.....	43
4. Conclusion	43

Chapitre 3 Conception de l'application de gestion de l'ontologie

1. Introduction.....	45
2. Proposition d'une architecture de l'application de gestion de l'ontologie	45
3. Langage de modélisation et méthode de conception.....	46
3.1. Définition.....	46
3.2. Choix d'une méthode d'analyse et de conception.....	46
4. Aperçu de La méthode simple et générique	47
4.1. Phase identification des besoins	48
4.1.1. Besoin	48
4.1.2. Diagramme de cas d'utilisation	49
4.1.3. Diagramme de séquence de système	49
4.1.4. Maquette de l'IHM.....	50
4.2. Phase d'analyse.....	51
4.2.1 Modèle du domaine	51
4.2.2 Diagramme de classes participantes.....	52
4.2.3. Diagrammes d'activités de navigation	53
4.3. Phase de conception	56
4.3.1. Diagramme d'interaction.....	56
4.3.2. Diagramme de classe de conception.....	57
5. Conclusion	58

Chapitre 4 Implémentation

1. Introduction.....	59
2. l'implémentation de l'ontologie.....	59
2.1. Les outils d'implémentation.....	59

2.1.1 Plateforme Protégé	59
2.1.2. Le classifieur Pellet	60
2.2 Création de l'ontologie avec Protégé OWL	61
2.2.1. Création des classes et hiérarchie de classes	61
2.2.2. Création des propriétés	62
2.2.3. Création d'attributs (DataTypeProperty)	63
2.2.4. Création des relations (ObjectProperty)	64
2.2.5. Restrictions sur les propriétés	65
2.2.6. Création des instances	67
2.2.7. Génération du code	68
2.2.8. Test & évolution de l'ontologie	69
3. l'implémentation de l'application de gestion de l'ontologie	71
3. 1. Java	71
3.2. Jena	72
3.3. NetBeans	73
3.4. Interfaces	74
4. Conclusion	79
Conclusion général	81
References bibliographiques	82
Glossaire	86
Liste des acronymes	87

Liste des figures

FIGURE1. 1. DOMAINES D'UTILISATION DES ONTOLOGIES	6
FIGURE1. 2. LE CYCLE DE VIE DES ONTOLOGIES	8
FIGURE1. 3. PROCESSUS DE CONSTRUCTION D'ONTOLOGIE	12
FIGURE2. 1. HIERARCHIES DE CONCEPTS.....	32
FIGURE2. 2. DIAGRAMME DES RELATIONS BINAIRES.....	33
FIGURE3. 1. ARCHITECTURE DE L'APPLICATION DE GESTION DE L'ONTOLOGIE PROPOSE...	45
FIGURE3. 2. LA METHODE SIMPLE ET GENERIQUE	47
FIGURE3. 3. MODELE DU DOMAINE.	52
FIGURE3.4. DIAGRAMME DE CLASSE PARTICIPANTE.	53
FIGURE3. 5. DIAGRAMMES D'ACTIVITES DE NAVIGATION.	54
FIGURE3.6. DIAGRAMMES D'ACTIVITES DE NAVIGATION.	55
FIGURE4.1.1. L'INTERFACE PRINCIPALE DE PROTEGE.....	60
FIGURE4.2. LE CLASSIFIEUR PELLET.....	60
FIGURE4.3. CREATION DE CLASSES.	61
FIGURE4.4. HIERARCHIES DES CONCEPTS.	62
FIGURE4.5. CREATION DE PROPRIETES POUR UNE CLASSE.	63
FIGURE4.6. CREATION D'UN ATTRIBUT.....	64
FIGURE4.7. CREATION D'UNE RELATION ENTRE DEUX CLASSES.	65
FIGURE4.8. LES RESTRICTIONS CREEES SUR LES PROPRIETES D'UNE CLASSE.	66
FIGURE4. 9. CREATION D'UNE RESTRICTION.	67
FIGURE4.10. CREATION D'INSTANCES.	68
FIGURE4.11. FRAGMENT DU CODE OWL GENERE PAR PROTÉGÉ.....	69
FIGURE4.12. LE TEST DE CONSISTANCE.	70
FIGURE4.13. LE TEST DE CLASSIFICATION.....	71
FIGURE4.14. PAGE D'ACCUEIL JENA.	72
FIGURE4.15. LE PROJET ONTOLOGIE AVEC LA BIBLIOTHEQUE JENA.....	73
FIGURE4.16. COPIE D'ECRAN DE L'INTERFACE PRINCIPALE.....	74
FIGURE4.17. COPIE D'ECRAN DE L'IMPORTATION DE L'ONTOLOGIE.	75
FIGURE4.19. COPIE D'ECRAN D'AJOUT D'UN INDIVIDU.	77
FIGURE4.20. COPIE D'ECRAN D'AJOUT D'UN ATTRIBUT.....	77
FIGURE4.21. COPIE D'ECRAN SUPPRESSION DU CONCEPT.	78

Liste des tableaux

TABLEAU2. 1.SYNTAXE DU LANGAGE SHIQ [1].	24
TABLEAU2. 2.GLOSSAIRE DES TERMES.	31
TABLEAU2. 3.DICTIONNAIRE DES CONCEPTS.	34
TABLEAU2. 4.TABLE DES RELATIONS BINAIRES.....	35
TABLEAU2. 5.TABLE DES ATTRIBUTS.	37
TABLEAU2. 6.TABLE DES AXIOMES LOGIQUES.	38
TABLEAU2. 7.TABLE DES INSTANCES.	40
TABLEAU2. 8.DEFINITION DES CONCEPTS ET SUBSOMPTION.	41
TABLEAU2. 9.LES DEFINITIONS DES DIFFERENTS ROLES.	42
TABLEAU2. 10.DESCRPTION DES ASSERTIONS DE CONCEPTS.	43
TABLEAU2. 11.DESCRPTION DES ASSERTIONS DE RELATIONS.....	43

Résumé

Depuis quelques années, l'usage des ontologies est plus en plus grandissant dans les différents domaines de recherche. Les ontologies médicales servent de plate-forme aux praticiens pour une prise de décision adéquate dans leur diagnostic. La prise en compte des différents aspects liés aux patients, aux maladies, aux remèdes et qui sont très nombreux est facilité par la création de telles ontologies médicales.

Nous nous intéressons au développement d'une application de gestion d'ontologie médicale en utilisant les outils NetBeans IDE, Jena, et le langage JAVA, afin de faciliter l'actualisation de notre ontologie.

Mots clés: Ontologie médicale, application de gestion d'ontologie, JAVA.

Introduction générale

1. Contexte générale

L'exploitation de connaissances en informatique a pour objectif de ne plus faire manipuler en aveugle des informations à la machine mais de permettre un dialogue (une coopération) entre le système et les utilisateurs. Alors, le système doit avoir accès non seulement aux termes utilisés par l'être humain mais aussi à la sémantique qui leur est associée, afin qu'une communication efficace soit possible. Actuellement, la connaissance visée par ces ontologies est un sujet de recherche populaire dans diverses communautés (l'ingénierie des connaissances, la recherche d'information, le traitement du langage naturel, les systèmes d'information coopératifs, l'intégration intelligente d'information et la gestion des connaissances). Elles offrent une connaissance partagée sur un domaine qui peut être échangée entre des personnes et des systèmes hétérogènes. Elles ont été définies en intelligence artificielle afin de faciliter le partage des connaissances et leur réutilisation. La définition explicite du concept ontologie soulève un questionnement qui est tout à la fois d'ordre philosophique, épistémologique, cognitif et technique.

Le problème est que les ontologies qui sont une forme de représentation de connaissances d'un domaine s'appliquent à des environnements ouverts et dynamique ou les connaissances de domaine sont amené à évoluer et les besoins utilisateur auxquels elle doit reprendre change, cela rend l'ontologie obsolète.

De ce fait l'ontologie ne doit pas être vue comme un modèle figé et doit être mise à jour régulièrement afin qu'elle sera capable de reprendre notamment :

- Aux évolutions des connaissances du domaine conceptualisé.
- Aux besoins changeants des divers acteurs humains ou logiciels.
- Aux intégrations de nouveau concept, propriété, individu et attribut.

2. Contribution

La transmission des informations des activités médicales des patients par les établissements hospitaliers est devenue obligatoire. Ces informations sous forme de résumés standardisés concernent les diagnostics principaux et secondaires et sont effectuées à partir de la classification internationale des maladies. En utilisant un thésaurus

Introduction générale

de spécialité, proposés par des sociétés prestataires de services, les médecins réalisent leur codage le plus souvent manuellement. Ces codages se révèlent dans la majorité des cas ambigus et non exhaustifs. Les systèmes informatiques de codage disponibles sur le marché utilisent généralement des thésaurus et de portée limitée. Pour pallier à cette insuffisance il est nécessaire de décrire la sémantique des objets et de l'organisation du domaine médical, et ceci dans le but de développer des modèles conceptuels qui ne sont ni contextuel ni ambiguë, et dont le sens est inscrit dans la structure même du modèle. Cette modélisation est appelé «ontologie». L'un des principaux objectifs de la recherche est que le développement de ces ressources ontologiques garantisse des outils puissants, fiables, et maintenables pour le codage.

La problématique à laquelle nous nous attaquons est donc la gestion de ces ontologies pour faciliter l'utilisation des ontologies et Afin de ne pas rester confinés à leur utilisation sur les informaticien, et pour cela nous avons construire notre ontologie «Diabète » et après nous avons développées notre application de gestion de cette ontologie.

3. Organisation de mémoire

Le mémoire est organisé en quatre chapitres :

Chapitre 1 : Les ontologies

Dans ce chapitre nous avons présenté la notion de l'ontologie ses composants et son utilisation, les types d'ontologie, leurs processus de construction et leurs déférentes méthodologies de construction, ainsi le cycle de vie d'ontologie et le formalisme de représentation d'ontologie. Et après nous avons présenté les Langages et les Outils de représentation d'ontologie.

Chapitre 2 : Construction de l'ontologie diabète

Nous commençons ce chapitre par une Définition détaillée sur les étapes de construction de notre ontologie de domaine médicale et nous donnons une petite description de notre domaine d'application médical qui est le domaine de diabète. Ainsi nous nous sommes

commencés les étapes de construction d'ontologie par : spécification des besoins, conceptualisation, formalisation, implémentation, evolution et vérification.

Introduction générale

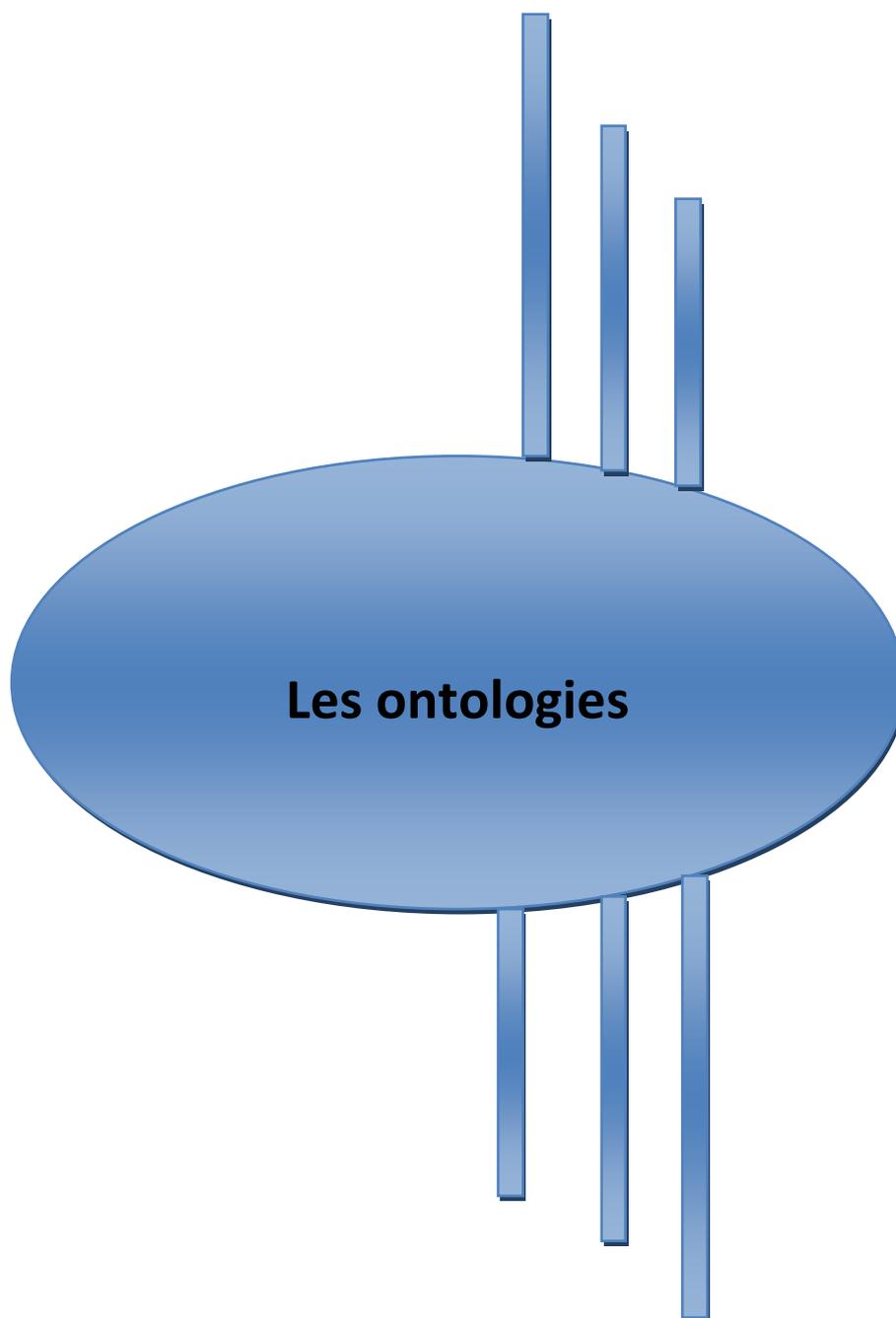
Chapitre 3 : Conception de l'application de gestion de l'ontologie

Dans Ce chapitre nous avons fait une description sur notre travail (la conception de l'application).Nous avons défini le langage de modélisation UML. Puis nous avons choisi la démarche simple et générique comme une méthode de conception d'application et leurs étapes. A partir du besoin utilisateur jusqu'à le code d'application.

Chapitre 4 : implémentation

Dans ce chapitre, nous avons implémenté l'ontologie médicale «diabète».Par le langage OWL qui facilité beaucoup cette phase, grâce à son pouvoir d'expression, et pour sa décidabilité, aussi l'utilisation de protégé OWL simplifié cette tâche à travers son interface graphique sophistiquée. Et finalement, nous avons présenté les outils utilisés dans le développent de notre application « santé » nous utilisons les outils NetBeans IDE, Jena, et le langage JAVA. Les différentes opérations de gestion de l'ontologie ont été développées dans ce chapitre. Nous avons montré comment on peut ajouter ou supprimé un concept, relation, individu ou attribut Après l'importation d'une ontologie.

Chapitre 1



1. Introduction

Le développement et l'exploitation des connaissances en informatique a tellement évolué, qu'un des principaux objectifs de la recherche actuelle est de ne plus considérer l'ordinateur comme une boîte noire, dans laquelle, sont stockées des informations, mais plutôt comme une machine intelligente avec laquelle on peut dialoguer et créer une coopération. Le système doit alors, avoir accès non seulement aux termes utilisés par l'être humain mais aussi à la sémantique qui leurs est associée, afin qu'une communication efficace soit possible.

Dans le contexte du web sémantique, les ontologies visent à améliorer l'exploitation des ressources informationnelles en se positionnant comme une modèle de représentation de connaissances .Cependant, la pertinence de l'information qu'elles contiennent nécessite une mise a jour régulière.

Dans ce chapitre nous parlerons sur l'ontologie de tous ses cotées a partir de l'historique, la définition, L'utilisation, le cycle de vie et son types, puis nous présentons les éléments constituant de l'ontologie et les étapes de construction d'une ontologie. Nous décrivent ensuite les méthodologies de construction .Enfin nous définissons le formalisme et les langages de représentation.

2. Historique

À l'origine, l'ontologie est une notion philosophique, et on considérait que l'étude de l'ontologie était une partie de la métaphysique, qui s'intéresse à l'étude des propriétés de l'être, et par extension de l'existence. Le mot ontologie provient du grec «*onto*» qui signifie ce qui existe, comme l'être, et l'existant, et «*logos*» qui veut dire étude. John McCarthy a introduit l'ontologie en intelligence artificielle en 1980, par le principe que les concepteurs de systèmes intelligents fondés sur la logique devraient d'abord établir une base de données complète de ce qui existe, et ensuite utiliser ces données.

L'apparition du terme «*Ontologie* » dans l'informatique a vu le jour au début des années 1990, grâce au projet d'ARPA Knowledge Sharing Effort (Effort de partager la connaissance). Selon son sens philosophique, une ontologie est une explication systématique de l'être [1] [2].

3. Notion d'ontologie

Afin de faciliter le partage et la réutilisation de connaissances formellement représentées dans des systèmes d'intelligence artificielle, il est très utile de définir un vocabulaire commun dans lequel la connaissance partagée est représentée. La spécification de ce vocabulaire est communément appelée une ontologie. De ce fait, les ontologies définissent actuellement des vocabulaires structurés, regroupant des concepts utiles d'un domaine et de leurs relations et qui servent à organiser et échanger des informations de façon non ambiguë. Leur développement progressif en (IA) Intelligence Artificielle parvient de leur intérêt, pour associer la sémantique à des ressources ou bien à des entités textuelles, pour faciliter la localisation et la gestion des connaissances dans diverses applications [3].

4. Définition

Dans le cadre de l'intelligence artificielle Neeches et ses collègues [4] furent les premiers à proposer une définition à savoir « une ontologie définit les termes et les relations de base de vocabulaire d'un domaine, ainsi que les règles qui indiquent comment combiner les termes et les relations de façon à pouvoir étendre le vocabulaire». En 1993, GRUBER [5] propose la définition suivante: «une ontologie est une spécification explicite d'une conceptualisation». En 1995, GUARINO [6] a modifié légèrement la définition de GRUBER, et la définit par: «une ontologie est une spécification partielle et formelle d'une conceptualisation». En 1997, ces deux dernières définitions sont regroupées dans celle de BORST [7] comme « une ontologie est définit comme étant une spécification formelle d'une conceptualisation partagée».

- ❖ Conceptualisation : le modèle abstrait d'un phénomène du monde réel par identification des concepts clefs de ce phénomène.
- ❖ Explicite : le type de concepts utilisés et les contraintes liés à leur usage sont définis explicitement.
- ❖ Formelle : une ontologie doit être traduite en langage interprétable par une machine.

- ❖ Partagée : capture la connaissance consensuelle, c'est-à-dire non réservée à quelques individus, mais partagée par un groupe ou une communauté.

5. utilisation des ontologies

Dans cette section, nous allons voir pourquoi a-t-on besoin des ontologies. Les ontologies sont utilisées dans plusieurs domaines, les plus répandus sont :

- Communication.
- Interopérabilité entre les systèmes.
- Ingénierie des systèmes.

La figure ci-dessous montre les domaines d'utilisations des ontologies :

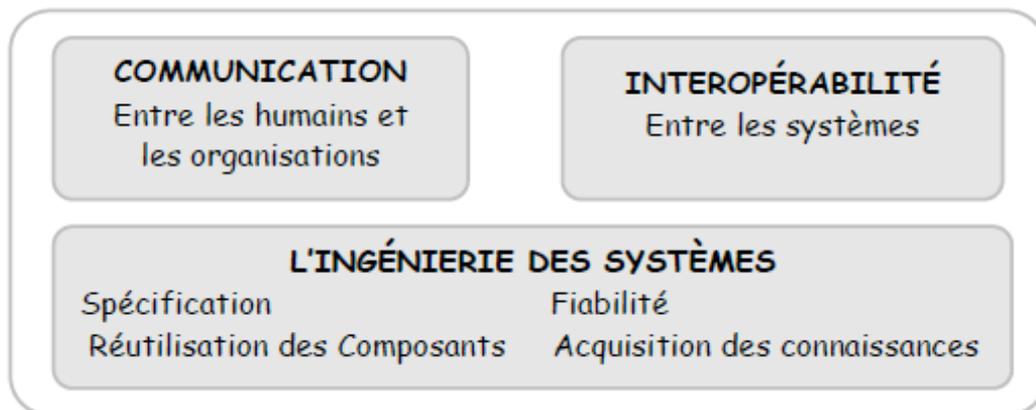


Figure1. 1.Domains d'utilisation des Ontologies [8].

5.1. Communication

Les humains peuvent communiquer efficacement s'ils ont des connaissances ou des points de vue partagés. Ces connaissances partagées peuvent être obtenues si le domaine est explicitement décrit sans confusion terminologique ou conceptuelle pour être compris de la même façon par tout le monde.

Une ontologie facilite la communication en fournissant une spécification explicite d'un domaine qui représente un modèle normatif. De plus, les ontologies permettent d'assurer la consistance et d'enlever l'ambiguïté dans les descriptions des connaissances concernant un domaine spécifique. Finalement, les ontologies peuvent intégrer différentes

perspectives des utilisateurs. Quand les utilisateurs (qui ont différentes perspectives d'un domaine) partagent une ontologie, ils ont une perspective standard.

5.2. Interopérabilité

L'interopérabilité implique la possibilité de pouvoir demander et recevoir des services entre des systèmes interopérables. Deux systèmes sont considérés interopérables s'ils vérifient les deux conditions suivantes :

- Ils opèrent comme une unité afin de réaliser une tâche commune.
- Ils peuvent échanger des messages et des requêtes.

Les ontologies permettent de faciliter l'interopérabilité en intégrant les connaissances concernant différents domaines dont l'objectif est de décrire un domaine unifié ou accomplir une tâche commune. Elles permettent aussi d'intégrer les différents vocabulaires concernant certains domaines. Pour ce faire, les ontologies de ces domaines doivent être intégrées par les méthodes d'intégration d'ontologies afin de partager un même vocabulaire.

5.3. Ingénierie des systèmes

Le développement des systèmes basé sur les ontologies a donné un profit à l'ingénierie de systèmes qui peut être résumé comme suit:

- Réutilisabilité : l'ontologie encode les informations relatives à un domaine (y compris les composants logiciels) de sorte que le partage et la réutilisation sont possibles.
- Acquisition des connaissances : l'ontologie guide l'acquisition des connaissances.
- Sûreté : l'ontologie rend possible l'automatisation du processus de vérification de consistance.
- Spécification : l'ontologie aide le processus d'identification des besoins et la définition des spécifications des systèmes [8].

6. Le cycle de vie des ontologies

Le cycle de vie d'une ontologie comprend des étapes liées à la construction d'ontologie et des étapes liées à son évolution ou sa maintenance. Il comprend l'étape des besoins, conception, diffusion, utilisation, d'évaluation et évolution.

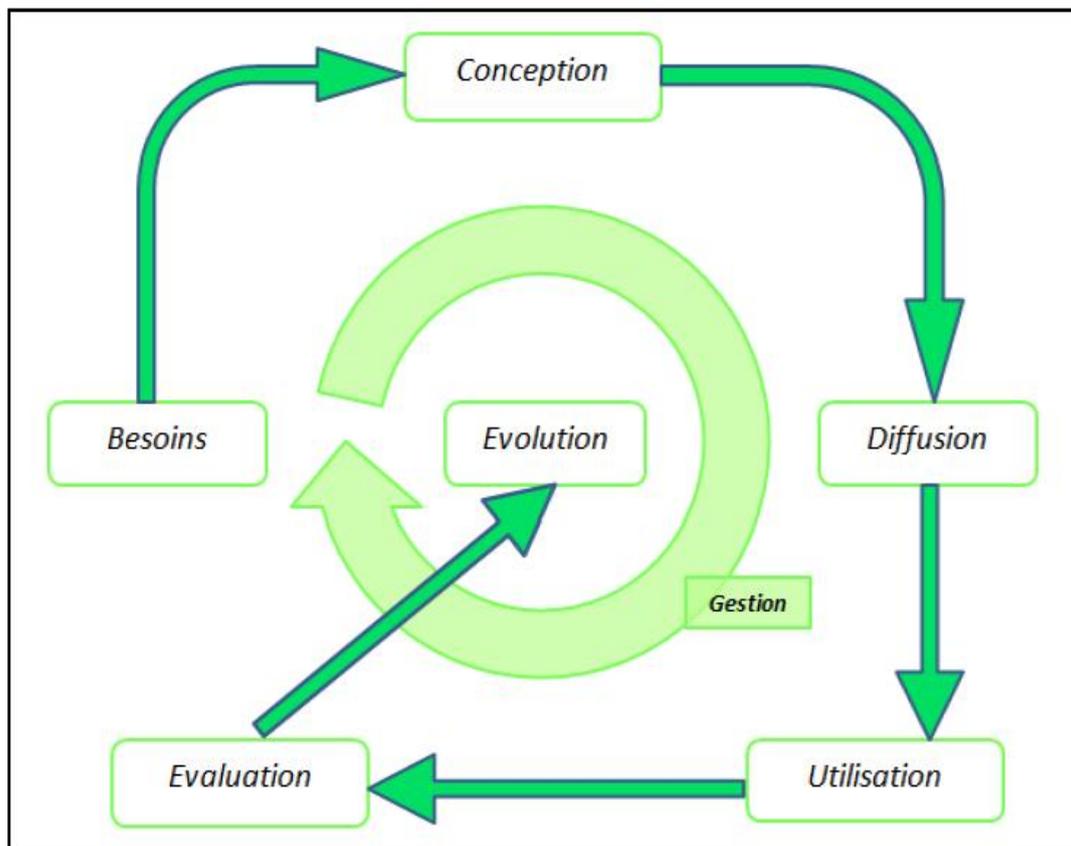


Figure1. 2.Le cycle de vie des ontologies [16].

7. Les types des ontologies

Gomez [9] propose une classification selon le sujet de conceptualisation des ontologies. Cette classification est une extension des travaux de Mizoguchi [10], ceux de Van Heijst [11], et ceux de Guarino [12].

a) Les ontologies de représentation de connaissances

Elles regroupent les primitives utilisées pour formaliser les connaissances sous un paradigme de représentation de connaissances.

L'exemple le plus expressif est l'ontologie de frame (Frame Ontologie) qui intègre les primitives de représentation des langages à base de frames: classes, instances, propriétés, relations, etc. [13].

b) Les ontologies générales ou communes ou génériques

Les connaissances modélisées dans ce type d'ontologie doivent être générales pour être réutilisées dans différents domaines. Elle comprend le vocabulaire relatif au temps, espace, unités, etc. [10].

c) Les ontologies de niveau supérieur ou de haut niveau

Les ontologies de niveau supérieur modélisent des concepts de haut niveau auxquels ces derniers doivent être reliés au sommet des ontologies de plus bas niveaux. Cependant, il existe plusieurs ontologies de haut niveau qui se diffèrent par le critère utilisé pour classer les concepts généraux de la taxonomie.

d) Les ontologies du domaine

Elles sont réutilisables au sein d'un domaine donné, mais pas d'un domaine à un autre. Les connaissances représentées dans ce type d'ontologies sont spécifiques à un domaine particulier. Elles fournissent un vocabulaire d'un domaine spécifique au travers de, concepts et de relations qui modélisent les principales activités, les théories du domaine en question. Les concepts et les relations des ontologies de domaine sont souvent des spécialisations de concepts et des relations définis dans des ontologies de haut niveau [12].

e) Les ontologies de tâches

Ce type d'ontologies est utilisé pour décrire un vocabulaire relatif à une tâche ou une activité générique (faire un diagnostic, planifier une activité ...) en spécialisant certains termes des ontologies de haut niveau. Ces ontologies fournissent un ensemble de termes au moyen desquels on peut décrire, au niveau générique, comment résoudre un type de problème [13].

f) Les ontologies d'application

Ce sont les ontologies les plus spécifiques. Contrairement à l'ontologie de domaine, l'ontologie d'une application donnée ne peut pas être réutilisée pour d'autres applications. Elle contient les connaissances requises pour une application particulière. Ce type d'ontologie décrit des concepts qui dépendent à la fois d'un domaine particulier et d'une tâche particulière. Par conséquent, elle spécialise souvent des ontologies de domaine et des ontologies de tâches pour une application donnée [12].

8. Les éléments constituant une ontologie

Comme mentionné plus haut, les ontologies produisent un vocabulaire commun d'un domaine et définissent, de façon plus ou moins formelle, la signification des termes et des relations entre eux. Les connaissances intégrées dans les ontologies sont formalisées en mettant en jeu quatre types de composants : concepts, instances, relations, axiomes [14] [15].

8.1. Les concepts

Ou classes, définissant un ensemble d'objets, abstraits ou concrets, que l'on souhaite modéliser pour un domaine donné. Les connaissances portent sur des objets auxquels on se réfère à travers des concepts. Un concept peut représenter un objet matériel, une notion, une idée [16].

Les concepts manipulés dans un domaine de connaissance sont organisés au sein d'un réseau de concepts liés par des propriétés conceptuelles. Les propriétés portant sur des concepts sont:

- **L'abstraction** : un concept est abstrait si toute instance de ce concept est aussi instance d'un de ses Concepts fils. Par exemple, dans une hiérarchie comportant les concepts homme et femme, fils du Concept humain, le concept humain est abstrait.
- **La subsomption** : un concept C1 subsume un concept C2 si toute propriété sémantique de C1 est aussi une propriété sémantique de C2, c'est-à-dire si C1 est plus spécifique que C2.

L'extension d'un concept subsumé est forcément plus réduite que celle du concept qui le subsume. Son intension est par contre plus riche. La subsomption sert à la hiérarchisation de l'ensemble des concepts de l'ontologie. Par exemple, homme subsume humain.

- **L'équivalence** : deux concepts sont équivalents s'ils ont la même extension. Par exemple, étoile du matin et étoile du soir.
- **La disjonction (on parle aussi d'incompatibilité)** : deux concepts sont disjoints si leurs extensions sont disjointes. Par exemple, homme et femme. Si tous les concepts fils d'un concept abstrait sont disjoints deux à deux, ils forment une partition du concept abstrait.

8.2. Relations

Représentent un type d'interaction, ou bien des associations existant entre les concepts d'un domaine. Elles se définissent formellement à partir d'un produit de n concepts :

$$R : C_1 \times C_2 \times \dots \times C_{n-1} \square \square G_n$$

Voici quelques relations les plus courantes dans la littérature :

1) **L'équivalence** : une relation R est une relation d'équivalence si et seulement si : R est symétrique, réflexive et transitive. On écrit :

$$(R \text{ est une relation d'équivalence}) \square ((R \text{ symétrique}) \wedge (R \text{ réflexive}) \wedge (R \text{ transitive}))$$

2) **la cardinalité** : c'est le nombre possible de relations de ce type entre les mêmes concepts (ou instances de concept). Les relations portant une cardinalité représentent souvent des attributs. Exemple : une pièce a au moins une porte, un humain a entre zéro et deux jambes.

3) **L'incompatibilité** : Deux relations sont incompatibles si elles ne peuvent lier les mêmes instances de concepts. Exemple : les relations «être rouge » et «être vert» sont incompatibles.

4) **L'inverse** : Deux relations binaires sont inverses l'une de l'autre si, quand l'une lie deux instances I_1 et I_2 , l'autre lie I_2 et I_1 . Exemple : les relations « a pour père » et « a pour enfant » sont inverses l'une de l'autre.

5) **L'exclusivité** : Deux relations sont exclusives si, quand l'une lie des instances de concepts, l'autre ne lie pas ces instances, et vice-versa. L'exclusivité entraîne l'incompatibilité. Exemple : l'appartenance et le non appartenance sont exclusives.

Et bien d'autres relations...

8.3. Les instances

Où individus, constituent la définition extensionnelle de l'ontologie (pour représenter les éléments spécifiques).

9. Construction d'une ontologie

9.1. Un squelette de méthodologie pour construire des ontologies

Le processus de construction d'une ontologie est une collaboration qui réunit des experts du domaine de connaissance, des ingénieurs de la connaissance, voire les futurs utilisateurs de l'ontologie. Cette collaboration ne peut être fructueuse que si les objectifs du processus ont été clairement définis, ainsi que les besoins qui en découlent. La figure ci-dessous représente le processus de construction d'ontologie.

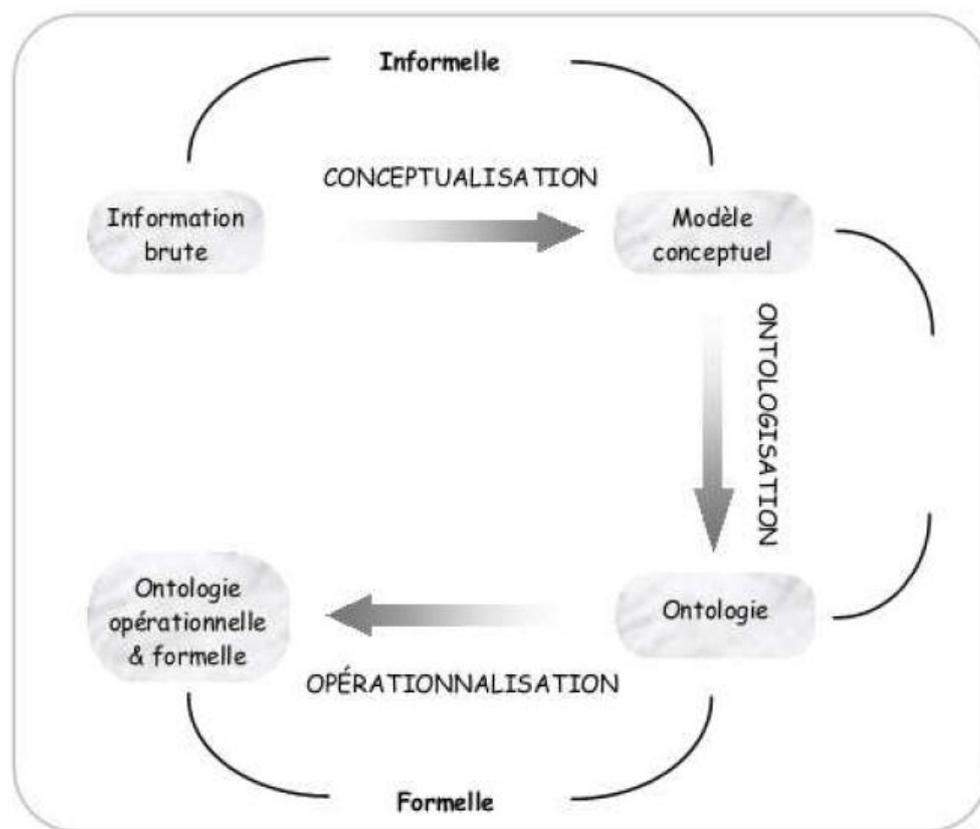


Figure1. 3.Processus de construction d'ontologie [17].

9.1.1 Evaluation des besoins

Le but visé par la construction d'une ontologie se décline en 3 aspects:

- L'objectif opérationnel : il est indispensable de bien préciser l'objectif opérationnel de l'ontologie, en particulier à travers des scénarios d'usage.

- Le domaine de connaissance : il doit être délimité aussi précisément que possible.
- Les utilisateurs : ils doivent être identifiés autant que faire se peut, ce qui permet de choisir, en accord avec l'objectif opérationnel, le degré de formalisme de l'ontologie, et sa granularité.

Une fois le but défini, le processus de construction de l'ontologie peut démarrer, en commençant par la phase de conceptualisation.

9.1.2. Conceptualisation

Cette étape permet d'aboutir à un modèle informel, donc sémantiquement ambiguë et généralement exprimé en langage naturel. Elle consiste, à partir des données brutes, à dégager les concepts et les relations entre ces concepts permettant de décrire de manière informelle les entités cognitives du domaine.

L'objectif est d'aboutir à un modèle conceptuel : le modèle obtenu consiste en un ensemble de termes désignant les entités du domaine de connaissances (concepts, relations, propriétés des concepts et des relations, ...), assortis d'informations exprimant leur sémantique. La découverte des connaissances d'un domaine peut s'appuyer à la fois sur l'analyse de documents et sur l'interview d'experts du domaine. Ces activités doivent être raffinées au fur et à mesure que la conceptualisation émerge.

9.1.3. Ontologisation

L'Ontologisation consiste en une formalisation partielle, sans perte d'information, du modèle conceptuel obtenu dans l'étape précédente. Ce qui permet de faciliter sa représentation ultérieure dans un langage complètement formel et opérationnel.

Elle effectue une transcription des connaissances dans un certain formalisme de connaissances, ce formalisme devant être aussi générique que possible, mais sémantiquement clair. S'imposer de conserver toutes les connaissances conduit à intégrer, à l'ontologie du domaine, des connaissances qui ne peuvent être formalisées, ou dont la sémantique est ambiguë.

Le modèle obtenu est souvent qualifié de semi-formel (car certaines connaissances ne peuvent pas être totalement formalisées).

En revanche, une ontologie, contenant toutes les connaissances d'un domaine, constitue le support idéal de communication et de partage des connaissances de ce domaine.

9.1.4. Opérationnalisation

Cette étape consiste à formaliser complètement l'ontologie obtenue dans un langage de représentation de connaissances formel (i.e. possédant une syntaxe et une sémantique) et opérationnel (i.e. doté de services inférentiels permettant de mettre en œuvre des raisonnements), par exemple, le modèle des Graphes Conceptuels ou la Logique de Descriptions.

On obtient alors une représentation formelle des connaissances du domaine. Ainsi, le caractère formel de l'ontologie permet à une machine, via cette ontologie, de manipuler des connaissances du domaine. La machine doit donc pouvoir utiliser des mécanismes opérant sur les représentations de l'ontologie.

9.2. Quelques méthodologies de construction d'ontologies

Les méthodologies peuvent porter sur l'ensemble du processus et guider l'otologiste dans toutes les étapes de la construction. Bien qu'aucune méthodologie générale n'ait pour l'instant réussi à s'imposer, de nombreux critères de construction d'ontologies ont été proposés pour des méthodologies. ENTERPRISE, TOVE et METHONTOLOGY sont les méthodologies les plus représentatives pour construire des ontologies.

9.2.1 Tove

TOVE (Toronto Virtual Enterprise) développé par l'université de Toronto, cette méthodologie repose sur les expériences de développement d'une entreprise [18] [19]. Elle s'appuie également, pour le développement d'une ontologie, sur les principales étapes suivantes :

- Capturer des scénarios de motivations : Cette étape consiste à identifier des scénarios qui clarifient le domaine que l'on investit et les différentes applications dans lesquelles l'ontologie sera employée.
- Formuler des questions de compétences informelles : Cette étape consiste à formuler un ensemble de questions (basées sur les scénarios), exprimées en langage naturel, afin de déterminer la portée de l'ontologie. Ces questions et leurs réponses sont utilisées pour extraire les concepts principaux, leurs propriétés et les relations qui existent entre ces concepts.

- Spécifier la terminologie de l'ontologie : Cette étape consiste à représenter les termes (Concepts, propriétés et relations), identifier dans l'étape précédente, en utilisant le formalisme de la logique du premier ordre. Les concepts seront représentés sous forme de constantes ou bien des variables. Par ailleurs, les propriétés et les relations seront représentées par des prédicats.
- Evaluer la complétude de l'ontologie.

9.2.2 Enterprise

Uschold et King's [24], proposent le squelette d'une méthode basé sur l'expérience de construction d'ontologies dans le domaine de la gestion des entreprises. La méthode ENTERPRISE repose sur les trois étapes suivantes :

- Identifier le rôle et la portée de l'ontologie, Dans cette étape, l'ontologie est réellement

construite. Les activités suivantes sont distinguées : identifier les concepts et relations fondamentaux et des définitions provisoires de ces éléments, coder l'ontologie dans un langage adapté, intégrer des ontologies existantes.

- Evaluer l'ontologie.
- Rédiger une documentation et une trace des actions réalisées lors des différentes phases.

Les étapes et sous-tâches de la méthode ENTERPRISE, sont décrites de façon abstraite.

Les techniques utilisées pour les sous-tâches ne sont pas précisées (par exemple : Comment identifier les concepts fondamentaux ? Quel langage utiliser pour représenter l'ontologie ?

9.2.3. Methontology

La méthodologie de construction d'ontologies «METHONTOLOGY» se situe entre le GL (Génie Logiciel) et l'IC (Ingénierie des Connaissances). Elle identifie une séquence d'activités techniques à appliquer pour le développement de l'ontologie. Cette méthodologie a été motivée par le constat suivant : l'absence de méthodes ou de guides structurés est un obstacle à la construction d'ontologies partagées et consensuelles. Il est également un obstacle à l'extension d'une ontologie existante ou à sa réutilisation dans d'autres ontologies.

L'approche METHONTOLOGY distingue les étapes suivantes:

1) Spécification

Le développement d'une ontologie commence par la définition du domaine et du porté de celle-ci. Cela est basé sur la réponse à certaines questions : Quel est le domaine que l'ontologie va couvrir ? À quoi cette ontologie va servir ? Qui va utiliser et maintenir l'ontologie ?, etc. Les réponses à ces questions peuvent changer durant le processus de développement de l'ontologie, mais à chaque étape, elles permettent de limiter la portée du modèle. L'une des solutions qui permet de déterminer la portée d'une ontologie consiste à définir ou planifier une liste de questions auxquelles une base de connaissance, basée sur l'ontologie, doit être capable de répondre («competency questions») [18].

Ces questions peuvent servir à un test ultérieur de l'ontologie (Est-ce que l'ontologie contient des informations suffisantes pour répondre à ces questions ? Est-ce que les réponses nécessitent un certain niveau de détail ou la représentation d'un espace particulier ?), mais elles ne doivent pas être exhaustives.

2) Conceptualisation

Elle consiste à identifier et à structurer les connaissances du domaine, à partir des sources d'informations. L'acquisition de ces connaissances peut s'appuyer à la fois sur l'analyse de documents et sur l'interview des experts du domaine. Une fois que les concepts sont identifiés par leurs termes, leur sémantique est décrite dans un langage semi-formel (tables et graphes) à travers leurs propriétés, leurs instances connues et les relations qui les lient entre eux.

3) Implémentation

Cette étape consiste à formaliser le modèle conceptuel obtenu dans l'étape précédente par un formalisme de représentation d'ontologie telles que les logiques de description. Puis, à coder l'ontologie dans un langage d'ontologie formel.

4) Maintenance

Cela peut s'agir d'une maintenance corrective ou évolutive de l'ontologie (nouveaux besoins de l'utilisateur), ce qui permet la validation et l'évolution de celle-ci. Cette activité

est généralement faite par le constructeur et des experts du domaine. La validation se base sur l'exploitation des services d'inférences, qui sont offerts par des raisonneurs.

Pour conclure, nous avons constaté que la démarche METHONTOLOGY présente un certain nombre de phases spécifiées de manières très détaillées, notamment la phase de conceptualisation. De ce fait, nous allons adopter cette méthodologie et l'adapter pour les besoins de notre travail.

10. Formalisme de représentation des Ontologies

Pour qu'une machine puisse manipuler les connaissances d'un particulier et afin qu'elle puisse raisonner, il est impératif de représenter ces connaissances. Cette représentation consiste à décrire et à coder les entités de ce domaine. Deux grands paradigmes sont utilisés pour représenter des Connaissances : les graphes conceptuels et les logiques de description.

10.1. Les graphes conceptuels

Le formalisme des graphes conceptuels est développé par John Sowa en 1984[20]. Il tient son origine des réseaux sémantiques [21], et permet de représenter des connaissances sous forme d'un graphe (concepts et relations conceptuelles), le nœud étant associé à des concepts et les arêtes à des relations.

10.2. Logiques de description

Logiques de description Appelé aussi logiques de descriptives, introduite par Brachman en 1979 est une famille de langages de représentation de connaissances, elles permettent la représentation des connaissances relatives à un domaine à l'aide de <<description>> qui peuvent être des concepts (classes d'individus), des rôles (relation entre concepts), et des individus. Les concepts modélisent des classes d'individus et les rôles de la relation entre classes. La sémantique est associée aux descriptions par l'intermédiaire d'une fonction d'interprétation.

11. Langages et Outil de représentation d'ontologie

Une des principales décisions à prendre le procédé de développement d'ontologie consiste à choisir le langage ou les langages dans lesquels l'ontologie sera exprimée et utilisée [22], pour cela plusieurs langages ont été développés. Nous allons présenter brièvement certains langages de balisage XML.

11.1. Langages de représentation d'ontologie

11.1.1. RDF (Resource Description Framework)

Le W3C a adopté le langage RDF (Resource Description Framework) comme un langage du formalisme standard de représentation de connaissances sur le Web. Comme un langage basé sur des réseaux sémantique pour décrire les ressources web. Son développement a commencé en 1997. La syntaxe de RDF repose sur le langage de balisage extensible XML.

XML fournit une syntaxe pour encoder des données tandis que RDF fournit un mécanisme décrivant le sens de données. Un des buts de RDF est de rendre possible la spécification de la sémantique de données basées sur XML d'une manière standardisée et interopérable [23].

11.1.2. RDFS (Resource Description Framework Schéma)

RDF(S) construit par le W3C comme une extension de RDF avec des primitives basées sur les "frames". Ce langage a été proposé comme une recommandation candidate du W3C en 2000 avec une révision importante en novembre 2002, de sorte que son document de références ait été édité comme une ébauche de travail du W3C. Plus tard il a été révisé en 2003. La combinaison de RDF et de RDF schéma est connue sous le nom RDF(S). RDF schéma permet de décrire les ressources du vocabulaire RDF. Il définit la notion de classe et de propriété. Il permet de définir des hiérarchies de classes et de notion de classes et propriété [24].

11.1.3. OWL (Ontology Web Language)

OWL est un langage RDF pour la définition de classes et de propriétés (RDFS), et aussi OWL, est un langage pour représenter des ontologies dans le Web sémantique. Ce

langage est divisé en trois sous langages : OWL-Lite, le plus simple, OWL-DL, basé sur la logique de description et OWL-Full, le plus expressif mais dans la calculabilité n'est pas garantie.

- **OWL-Full [25]**: contient toutes les constructions possibles avec le langage OWL.
- **OWL-DL** : c'est un sous ensemble d'OWL Full qui suit les règles suivantes :
 - Une classe ne peut être membre d'une autre classe.
 - Des restrictions sont établies sur les propriétés fonctionnelles, et transitives.
 - Une restriction est ordonnée sur OWL : lorsqu'on développe avec OWL DL on ne peut importer un document écrit avec OWL Full [26].
- **OWL-Lite** : ne contient qu'un sous-ensemble réduit des constructeurs disponibles, Il a la complexité formelle la plus basse et l'expressivité minimale dans la famille, il est suffisant pour la représentation de classification hiérarchique avec des contraintes simples OWL [27].

11.2. Outils de représentation d'ontologie

11.2.1 .Protégé

L'environnement Protégé, Créé par les chercheurs de l'université de Stanford, est un éditeur d'ontologies développé en Java, gratuit et open source. Il s'agit d'une plateforme d'aide à la création, la visualisation et la manipulation d'ontologies dans divers formats de représentation (RDF, RDFS, OWL, etc.). Ce logiciel peut également être utilisé en combinaison avec un moteur d'inférence (tel que Racer ou Pellet) afin d'effectuer des raisonnements et d'obtenir de nouvelles assertions.

La plateforme Protégé permet la création et l'édition d'ontologies grâce à deux outils distincts:

- Protégé Frame permet de créer facilement une interface graphique afin de gérer une ontologie. Cet outil ne demande aucune notion de programmation. Il génère automatiquement les formulaires nécessaires en se basant sur le schéma d'ontologie créé. Il offre également la possibilité de personnaliser l'interface selon les besoins de l'utilisateur.

- Protégé OWL est une extension de Protégé qui supporte le langage OWL. Il permet de décrire plus précisément les classes, les propriétés et les instances grâce aux nombreuses propriétés offertes par OWL. Il est également possible d'interroger un raisonneur via une interface DIG afin de contrôler l'intégrité du modèle et de créer un modèle d'inférences. [28]

12. Moteurs d'inférences

La sémantique formelle du langage OWL permet l'application des techniques de raisonnement pour effectuer des dérivations logiques. Ces dérivations sont effectuées par des moteurs d'inférence (également nommés moteurs de raisonnement, raisonneurs sémantiques, ou tout simplement des raisonneurs), ce sont des programmes qui peuvent lire des ontologies à partir de fichiers OWL ou des serveurs web distants, ce sont donc des systèmes capables de gérer et d'utiliser la sémantique du langage de l'ontologie.

Comportant deux phases:

1. Phase de transformation (mapping) des connaissances vers un formalisme de représentation de connaissance.
2. Application d'un mécanisme d'inférence pour calculer les inférences. Pellet et Racer sont à l'heure actuelle les deux seuls moteurs d'inférence, permettant le raisonnement sur la ABox et la TBox et exploitent des ontologies possédant un niveau d'expressivité en logique de description et acceptent en entrée des fichiers OWL.

12.1. Racer

Le système Racer (Renamed ABox and Concept Expression Reasoner ou raisonneur d'expression de concept et de ABox renommées) est un système de représentation de connaissance pour le calcul DL.

Racer est le moteur d'inférence sans doute le plus connu et l'un des plus utilisés pour ces performances et sa stabilité. Racer travaille sur les ontologies modélisées par son langage, mais il accepte des ontologies décrites en RDF ou OWL, ces dernières étant traduites vers le langage utilisé par Racer. Ce moteur d'inférence possède également son propre langage de requête nRQL (new Racer proquery Language) pour interroger les ontologies sur la

ABox et la TBox. Racer se présente sous la forme d'un serveur qui peut être accédé par le protocole TCP ou http [29].

- ✓ JRacer est l'interface TCP java qui peut être utilisé depuis des programmes clients qui souhaitent accéder au serveur.
- ✓ Pour un accès HTTP, on utilisera DIG qui est l'interface HTTP.
- ✓ Rice (Racer Interactive Client Environment) est le client graphique de Racer.
- ✓ Racer peut être utilisé dans Protégé OWL.

12.2. Pellet

Le moteur Pellet est beaucoup plus récent. Pellet est un des projets du MINDSWAP Group, un groupe de recherche sur le web sémantique de l'université du Maryland. Il est disponible en Open Source et offre des évolutions fréquentes. Pellet travaille sur des ontologies décrites en RDF ou OWL [30].

13. Conclusion

Dans ce chapitre nous avons présenté la notion d'ontologie ses composants et son utilisation, les types d'ontologie, leurs processus de construction et leurs différentes méthodologies de construction ainsi le cycle de vie d'ontologie et le formalisme de représentation d'ontologie.

Dans un seconde tempe nous avons entamé les Langages et les Outils de représentation d'ontologie.

Dans le prochain chapitre nous allons faire une description sur les étapes de construction de l'ontologie et faire la création de notre ontologie diabète.

Chapitre2



1. Introduction

Avec l'explosion du Web et l'accroissement des connaissances médicales, les utilisateurs de ce monde ont « pratiquement » accès à des informations de plus en plus nombreuses.

Dans ce chapitre nous présentons les étapes de création de l'ontologie. Puis nous avons appliqué ces étapes à partir de spécification de besoin jusqu'à évolution et vérification.

Pour arriver à notre ontologie finale.

2. Processus de construction d'une ontologie OWL

Les ontologies étant destinées à être utilisées comme des composants logiciels dans des systèmes répondant à des objectifs opérationnels différents, leur développement doit appuyé sur les mêmes principes que ceux appliqués en génie logiciel. Différentes méthodes de construction ont été proposées (METHONTOLOGY, TOVE, ...).

Nous avons proposé un processus de construction d'une ontologie d'application partant de connaissances brutes et arrivant à une ontologie d'application opérationnelle représentée par le langage OWL. Ce processus est composé de cinq étapes [31] :

- ✓ Spécification des besoins.
- ✓ Conceptualisation.
- ✓ Formalisation.
- ✓ Implémentation.
- ✓ Test & évolution de l'ontologie.

2.1. Spécification des besoins

Le but visé par cette étape est d'établir un document de spécification des besoins. Ce dernier permet de décrire l'ontologie à construire à travers les cinq aspects suivants:

- **Le domaine de connaissance** : déterminer aussi précisément que possible le domaine que va couvrir l'ontologie.
- **L'objectif** : le but de l'ontologie à créer pour le domaine considéré.
- **Les utilisateurs** : identifier au maximum les futurs utilisateurs de l'ontologie à créer.

- **Les sources d'informations** : déterminer les sources d'informations d'où les connaissances seront obtenues, par exemple, les experts du domaine, les documents techniques, Internet etc., sont des sources de connaissance.
- **La portée de l'ontologie** : déterminé la liste des termes les plus importants pour le domaine à représenter.

Le résultat de cette étape est un document de spécification, généralement exprimé en langage naturel.

2.2. Conceptualisation

C'est l'étape la plus importante dans le processus de construction de l'ontologie. Elle est inspirée de la méthodologie METHONTOLOGY qui consiste à identifier et à structurer, à partir des sources d'informations, les connaissances du domaine. La découverte de ces connaissances peut s'appuyer à la fois sur l'analyse de documents et sur l'interview d'experts du domaine. Dans cette étape, les termes de l'ontologie sont choisis, ainsi que leurs propriétés, les relations qu'ils entretiennent entre eux, les contraintes qui s'appliquent sur eux, etc. Pour cela on distingue les principales activités suivantes :

- **Construction d'un glossaire de termes** : La première chose à faire quand le constructeur de l'ontologie essaye de capturer la connaissance d'un domaine donné est de construire un glossaire de terme. Le glossaire recueille et décrit tous les termes qui sont utiles et potentiellement utilisables dans le domaine que l'on investit.
- **Hiérarchies de concepts**: Une hiérarchie de Concepts organise un groupe de concepts entre eux sous forme d'une taxonomie. Il faut Vérifier que tous les concepts qui sont représentés dans les différentes hiérarchies de concepts, sont dans le glossaire. Dans le cas contraire ces derniers devraient être rajoutés au glossaire.
- **Diagramme relation binaire** : représenter les relations qui existent entre les différents concepts par un diagramme de relations binaires lie un concept source à un concept cible, qu'ils appartiennent ou non à la même hiérarchie.
- **Dictionnaire de concepts (DC)** : Identifier les concepts par leurs instances, leurs attributs ainsi que leurs Concepts synonymes dans un dictionnaire de concepts (DC).

- **La table de relations binaires:** La table des relations binaires définit pour chaque relation utilisée dans le diagramme, le nom de la relation, le nom des concepts source et cible, le nom de la relation inverse et les cardinalités source et cible.
- **La table d'attributs:** Pour chaque attribut d'un concept inclus dans le DC, la table des attributs spécifie le nom de cet attribut, son concept, sa description et son type.
- **La table d'axiomes logiques:** un axiome permet de définir certains concepts au moyen des expressions logiques. Sa description doit comporter le nom du concept sur lequel il porte, une définition en langage naturel et son expression logique.
- **La table d'instances:** La table des instances décrit toutes les instances (incluses dans le champ instances de dictionnaire de concepts) avec leurs attributs et valeurs.

2.3. Formalisation

Cette étape consiste à formaliser l'ontologie conceptuelle obtenue dans la phase précédente afin de faciliter sa représentation ultérieure dans un langage complètement formel et opérationnel. Notre choix est porté sur le formalisme de représentation de la logique de description en s'appuyant sur sa syntaxe de type SHIQ qui présente une logique de description très expressive et qui offre un certain nombre de constructeurs pour décrire les concepts.

Syntaxe	Interprétation
T	Le concept le plus général
\perp	Le concept le plus spécifique
C	Le nom d'un concept
R	Le nom d'un rôle (une relation binaire ou bien un attribut)
A	Le nom d'un individu
$C1 \cap C2$	Conjonction de concepts pour définir de nouveaux concepts
$C1 \cup C2$	Disjonction de concepts pour définir de nouveaux concepts
$\neg C$	Utilisé pour définir le complément d'un concept
$\forall R.C$	Définit le Co-domaine du rôle R
$\exists R.C$	Il y a au moins un objet relié par le rôle R au concept C
$(\geq n R)$ $(\leq n R)$	Cardinalité Minimum/Maximum (n est un nombre entier non négatif).
R^-	Le rôle inverse

Tableau2. 1.Syntaxe du langage SHIQ [1].

La logique de descriptions est constituée de deux parties : une partie terminologique (TBox) permettant de décrire les concepts et les rôles et une partie assertion elle (ABox) décrivant les instances.

2.4. Implémentation

L'ontologie que nous avons obtenue dans la phase précédente est appelée une ontologie formelle. Le but de cette étape sera donc de coder l'ontologie formelle en OWL qui dispose de fonctionnalités sémantiques plus riches que ses prédécesseurs comme RDFS et DAML+OIL. A la fin de cette phase, nous aurons une ontologie opérationnelle.

Afin de faciliter le processus de codification, nous utilisons PROTEGE-OWL disposant d'une interface modulaire, permettant l'édition, la visualisation, le contrôle (vérification des contraintes) d'ontologies, issu du modèle des frames et contient des classes (concepts), des slots (propriétés) et des facettes (valeurs des propriétés et contraintes), ainsi que des instances des classes et des propriétés.

2.5. Evolution et vérification

Cette étape consiste à suivre l'évolution de l'ontologie, c'est-à-dire lorsque le contexte d'usage est modifié, ou le domaine de connaissance est élargi lors de l'ajout d'un nouveau concept à l'ontologie. Une classification a lieu chaque fois qu'une définition de concept est nouvellement créée. Le mécanisme de raisonnement de base des logiques de description est la classification de concepts. Elle est réalisée par un algorithme de classification, appelé « le classifieur ». Le classifieur utilise la description d'un nouveau concept pour le placer à l'endroit correspondant dans la hiérarchie. Afin de trouver la place appropriée au nouveau concept, l'algorithme de classification détermine les relations de subsomption entre ce concept et les autres. Ces relations peuvent être spécifiées directement, trouvées par transitivité ou calculées à partir de la sémantique des conditions des rôles. Le test de classification est réalisé par le raisonneur Pellet.

Pour la vérification de l'ontologie résultante, nous appliquons les principales opérations de tests qui sont fournies par le système Pellet, à savoir le test de satisfiabilité d'un concept et le test de consistance.

3. Construction de l'ontologie Diabète dans le domaine médical

Dans cette section, nous construisons notre ontologie médicale qui concerne le service Diabétique. Pour ce faire nous allons suivre les étapes du processus de construction d'ontologie illustré dans la section précédente. Mais avant d'aller à la construction nous devons d'abord faire un petit prospect sur le domaine de notre application : le service Diabétique.

3.1. Le Diabète: domaine d'application

Le diabète est défini par une augmentation du sucre dans le sang (il s'agit du glucose). L'Association Américaine de Diabétologie (ADA) a proposé, en 1995, une modification des critères biologiques du diabète, en raison de données épidémiologiques internationales qui montraient que le diagnostic du diabète était tardif. Afin de faciliter la mise en œuvre du dépistage à l'échelle des praticiens de terrain, elle a simplifié les critères diagnostiques : le seuil glycémique à partir duquel on parle de diabète était de 1,40 g/l il est désormais de 1,26 g/l, puis la France à travers l'Agence nationale d'accréditation et d'évaluation en santé (ANAES) en 1999, ont également adopté ces valeurs afin de permettre une prise en charge plus précoce du diabète et d'élargir son domaine de définition à un plus grand nombre de sujets à risque.

Cette maladie chronique persiste toute la vie. La prise en charge correcte de la maladie par le patient lui-même, avec l'aide de son médecin, doit permettre d'en éviter les complications. Le diabète est responsable de mauvais états de santé et des décès prématurés [32].

3.2. Les grands types de la maladie

3.2.1. Le diabète de type 1 : insulino-dépendant (DID) aussi appelé diabète "maigre" car l'un des premiers symptômes est l'amaigrissement, ou "juvénile" parce qu'il touche des sujets jeunes.

- Il compte pour environ 10 % des cas et il est traité obligatoirement par l'insuline.

3.2.2. Le diabète de type 2 : non insulino-dépendant (DNID), aussi décrit sous le nom de diabète "gras" ou diabète de la maturité, puisqu'il survient souvent autour de la cinquantaine chez des personnes en surpoids [32].

- Il compte pour environ 90 % des cas et il est traité par régime, plus médicaments pris par voie orale si nécessaire, et éventuellement insuline, après quelques années d'évolution [32].

3.2.3. Le diabète de type 3 : C'est une forme rare de diabète (moins de 2% des diabètes sucrés) [32].

3.2.4. Le diabète gestationnel : Le diabète gestationnel, appelé aussi "diabète de grossesse", survient chez la femme enceinte vers la fin du 2e trimestre. Il peut durer le temps de la grossesse ou être révélateur d'un diabète antérieur [32].

3.3. Etape de spécification

Pour commencer le développement de l'ontologie, nous commençons d'abord la phase de spécification qui consiste à établir un document de spécification des besoins. Nous dériverons l'ontologie à construire à travers les cinq aspects suivants:

-**Le domaine de connaissance :** spécialité Diabétique de la médecine.

-**L'objectif :** pour automatiser le système et faciliter la gestion et la communication.

- **Les utilisateurs :** gestionnaires (médecin ou l'infirmier du service Diabétique).

- **Les sources d'informations :** nous nous sommes basés sur la recherche sur internet, mais aussi d'après des experts du domaine pour construire notre ontologie.

- **La portée de l'ontologie (liste des termes importants) :** parmi ces termes les plus Apportant sont : Hôpital de jour, Patiente, Médecin, Maladie, Infirmière, Malade, Les analyses, ...etc.

3.4. Etape de conceptualisation

Une fois la majorité des connaissances acquise, on doit les organiser et les structurer en utilisant des représentations qui sont faciles à comprendre et indépendantes de tout langage d'implémentation. Cette phase contient plusieurs étapes qui sont :

3.4.1. Construction du glossaire de termes

Thème	Description
Personne	Est un humain qui fait partie de la communauté de l'hôpital de jour.
L'interrogatoire	C'est le dialogue entre le médecin et le Patient.
Secrétariat	Est un bureau qui reçoit la carte de patient.
Psychologue	Est un docteur qui parle avec les patients qui ont un problème Psychologie.
Médecin Généraliste	Un médecin est un professionnel de la santé titulaire d'un diplôme de docteur en médecine. Il est chargé de soigner les maladies.
Bureau de consultation	Est un endroit contient des médecins pour consulter les patients.
Salle d'attend	Est une salle pour les malades qui attendent leur tour.
Bureau de chef de service	Est un bureau pour le stock de matériel.
Salle d'hospitalisation	Est une salle de soigne de patient dans le cas de danger (injection d'insuline, surhomme Glucagon)
Laboratoire	st une structure où des professionnels de la santé prélèvent et analysent des échantillons de sang.
Consultation	L'examen de la patiente par un médecin pour découvrir des cas de patient.
Infirmier	L'infirmier soigne les malades et veille à leur bien-être.
chef de service	Est le chef des infirmiers
Agent d'accueil	Personne chargé de recevoir et d'orienter les patients.
Médecin chef	Est un médecin généraliste qui faire un stage de deux ans dans la spécialité de diabétique (c'est lui qui gère le service).
Hémodialyse	La dialyse est une technique médicale permettant de supplanter les reins qui ne peuvent plus effectuer leur travail en cas d'insuffisance rénale très avancée.

Chapitre 2 Construction de l'ontologie diabète

Médecin interne	La médecine interne est une spécialité médicale. Elle s'intéresse au diagnostic et à la prise en charge globale des maladies de l'adulte avec une prédilection pour les maladies systémiques (par exemple le lupus érythémateux disséminé) et les maladies auto-immunes en général.
Infection	Une maladie infectieuse est une maladie provoquée par la transmission d'un micro-organisme : virus, bactérie, parasite, champignon, protozoaires.
Centre de Transfert de sang (CTS)	Est un service de don de sang.
Diabétique	diabète est une maladie qui se caractérise par un excès de sucre dans le sang (ou hyperglycémie). Il est défini par un taux de sucre dans le sang (glycémie) supérieur à 1,26 g/L à jeun, à deux reprises.
Registre de médecin	Pour la notification des informations de patient (nom, prénom, Age ...)
Glycémie capillaire	La glycémie capillaire ou hémoglucotest est une technique de lecture instantanée de la glycémie vasculaire capillaire.
Le diabète de type 1	Le diabète de type 1, anciennement appelé diabète insulino-dépendant (DID) est habituellement découvert chez les sujets jeunes : enfants, adolescents ou adultes jeunes.
Insuline	L'insuline est une hormone sécrétée par le pancréas. Elle est qualifiée d'hypoglycémisante, parce qu'elle diminue le taux de glucose (sucre simple) dans le sang (glycémie). Son insuffisance de fabrication provoque le diabète. (diminue le taux de glucose dans le sang)
Hôpital	Etablissement où l'on soigne les malades.
Carte de visite	Chaque patient a une carte qui contient son information.
Traitement	Des opérations faites par l'infirmier sur le malade (Soins de PLAIE).
Hôpital de jour	Service de diabétique dans l'hôpital.

Chapitre 2 Construction de l'ontologie diabète

Le diabète de type 2	Le diabète de type 2 apparaît généralement chez le sujet de plus de 40 ans ; cependant les premiers cas d'adolescents et d'adultes jeunes touchés apparaissent en France.
L'acidose lactique	L'acidose lactique est une complication grave du diabète correspondant à une accumulation excessive d'acide lactique dans le sang. Elle se rencontre essentiellement chez une personne diabétique souffrant également d'insuffisance rénale ou hépatique.
L'acidocétose	L'acidocétose est la conséquence médicale de la production par l'organisme de corps cétoniques en trop grosse quantité.
L'hypoglycémie	L'hypoglycémie (ne pas confondre avec hyperglycémie) est une concentration en sucre dans le sang (glycémie) anormalement basse c'est-à-dire inférieur à 0,60g/l.
Le coma hyperosmolaire	Le coma hyperosmolaire est une forme de coma qui affecte essentiellement les diabétiques de type 2 d'un âge avancé. Il se caractérise surtout par une hyperglycémie (taux de sucre élevé dans le sang), une polyurie (urines abondantes),
La micro angiopathie	La micro-angiopathie atteint les petits vaisseaux capillaires surtout au niveau des yeux, des reins et des nerfs.
Le diabète de type 3 MODY	Le diabète de type MODY est une forme particulière de diabète, différent du diabète auto-immun de type 1. Il est en rapport avec une anomalie de la régulation de la sécrétion d'insuline.
Le diabète gestationnel	Le diabète gestationnel est un état d'intolérance au glucose, quelle que soit sa sévérité, apparu au cours de la grossesse chez une femme sans diabète sucré connu antérieurement
La glycémie post prandiale	La glycémie post prandiale est la mesure se fait mesurée 1 h ½ à 2 h après le début du repas.
Urée	L'urée constitue la majeure partie azotée de l'urine. Sa production se déroule essentiellement dans le foie et une faible partie est produite par les reins.
La créatinine	La créatinine est une substance constituée d'azote qui provient de la dégradation de la créatine (un des constituants du tissu musculaire). Normalement, la créatinine doit être éliminée par les reins dans les urines.
Un médicament antidiabétique	Un médicament antidiabétique est un médicament utilisé pour traiter le diabète sucré. Les antidiabétiques agissent en général en abaissant la glycémie. Il y a différents types d'antidiabétiques, et leur utilisation dépend de la nature du diabète, de l'âge et de la situation de la personne, et de bien d'autres facteurs.

Chapitre 2 Construction de l'ontologie diabète

La metformine	La metformine (commercialisée sous les noms de Galactophage, Stagid et leurs génériques) est un antidiabétique oral de la famille des biguanides normoglycémiant utilisé dans le traitement du diabète de type 2.
Les sulfamidés	Les sulfamidés hypoglycémiant exercent essentiellement leur action en stimulant la sécrétion d'insuline par le pancréas
Les glinides	Les glinides sont des médicaments de la classe des antidiabétiques oraux qui agissent de façon semblable aux sulfamides hypoglycémiant, en forçant le pancréas à sécréter de l'insuline au moment des repas. Ce sont donc des insulino sécréteurs.
L'insuline rapide	est une insuline fabriquée en laboratoire, qui est identique à l'insuline fabriquée naturellement par le corps humain.
L'insuline semi lente	Pour éviter de multiplier les injections d'insuline, on a eu l'idée de mélanger l'insuline lente et l'insuline rapide
Hyperglycémie	Une hyperglycémie est une concentration en sucre dans le sang anormalement trop élevée. Pour un être humain,
coma hypoglycémique	Perte de conscience induite par une baisse excessive du sucre dans le sang.
L'acidose lactique	L'acidose lactique est une maladie survenant rarement, et qui correspond à un taux de lactate sanguins supérieure à 7 ml par litre et une acidité (PH) dosé dans le sang,
Patient	Est un (e)personne qui est caractérisé par : le nom, l'âge, la dernier date de visite
Agent-santé	Est un(e) personne affiliée dans le domaine de la santé. Il peut être : infirmier, Chef service
Injection Glucagon	Le glucagon est une hormone qui provoque le foie pour libérer le glucose dans le sang. Il est utilisé pour augmenter rapidement les niveaux de sucre dans le sang chez les diabétiques avec le sucre dans le sang (hypoglycémie).

Tableau2. 2.Glossaire des termes.

3.4.2. Hiérarchies de concepts

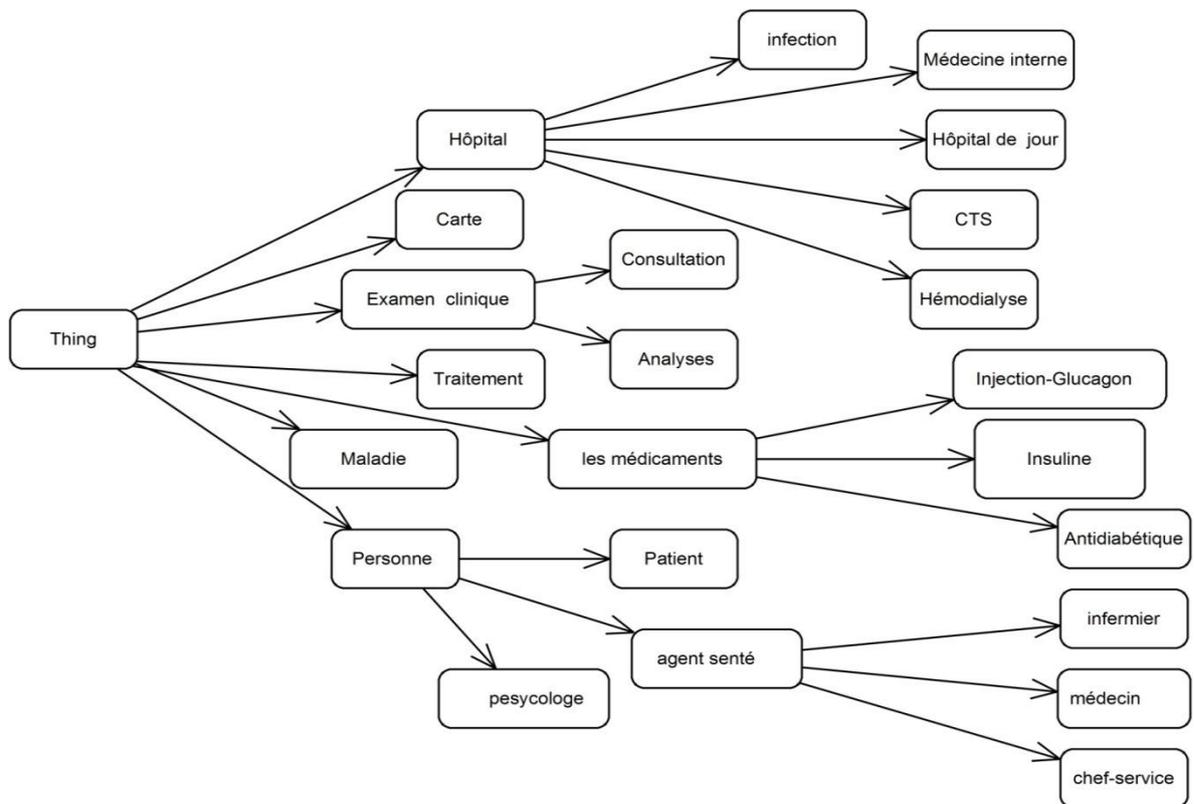


Figure 2. 1.Hiérarchies de concepts.

Dans la deuxième étape, nous représentons les relations binaires et la hiérarchie entre les classes par un diagramme. Dans ce diagramme, les classes sont représentées par des rectangles et les relations par des arcs orientés (du domaine vers le Co-domaine) et étiquetés par le nom de la relation. La figure ci-dessous (Figure 2.2) représente le diagramme de relations binaires.

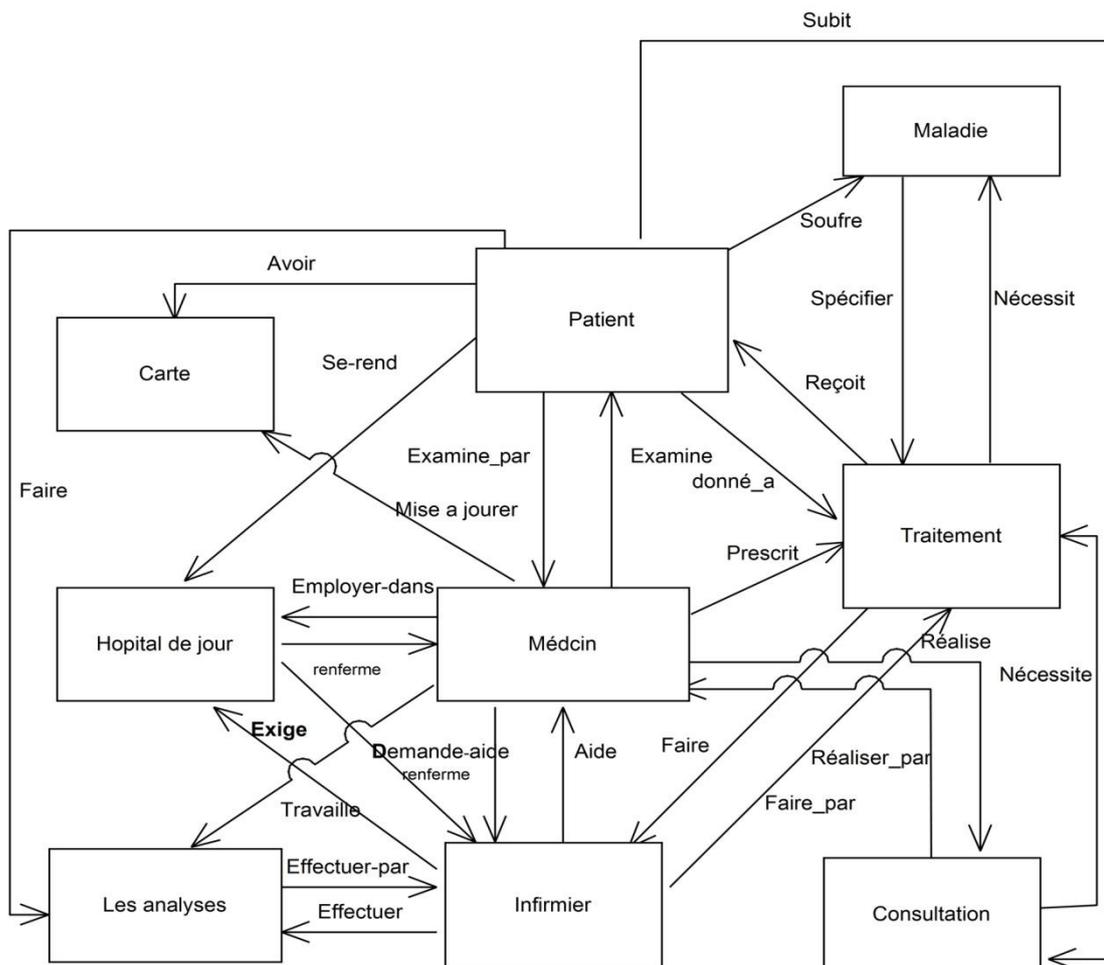


Figure 2. 2. Diagramme des relations binaires.

3.4.3. Dictionnaire de concepts (DC)

Dans cette étape nous allons donner une description formelle des concepts qui ont été présentés dans la hiérarchie des classes. Ce processus correspond à la création du dictionnaire de concepts accordé à METHONTOLOGY. Dans ce dictionnaire, nous définissons pour chaque concept : les instances, les attributs, les relations dont la source est ce concept, les synonymes et les acronymes de ce concept. Le tableau 2.3 représente le dictionnaire des concepts pour l'ontologie définie par la hiérarchie précédente.

Chapitre 2 Construction de l'ontologie diabète

Nom de concept	Concepts synonymes	Attributs	Instances
Personne	Humain	Nom Prénom Tel Adresse Age Sexe	Teriha Zakaria 066666666 Oued Segan 25 Homme
Agent -d'santé	Infirmier, Médecin	-	-
Patient	Homme, Femme, malade.		
Maladie	Syndrome	Nom-maladie	-diabète
Traitement	Cure, soins	-	-
Consultation	Examen, visite	-	-
Hôpital de jour	Centre hospitalier	-	-
Hôpital	Etablissement hospitalier,	Dénomination Adresse- Hôpital Tel-Hôpital Fax- Hôpital	Frères Tobal Centre ville Mila 031000000 031000001
Médecin	-		-
Medicament	-	Nom- Medicament	-
Chef service	Contrôleur	-	-
.....

Tableau2. 3.Dictionnaire des concepts.

3.4.4. Tableau de relations binaires

Les relations binaires sont représentées sous forme de propriétés ou attributs qui lient un concept à un autre, ce sont des attributs de type instance : c'est-à-dire les attributs ayant pour type de valeur Instance [2].

Nom de relation	Concept source	Cardinalité Source	Concept cible	Cardinalité Cible	Relation Inverse
Souffre	Patient	(1.1)	Maladie	(1.n)	-
Reçoit	Patient	(1.n)	Traitement	(0.n)	Donné-a
Examine	Médecin	(1.n)	Patient	(0,1)	Examine-par
Demande-aide	Médecin	(1.n)	Infirmier	(1.n)	Aide
Renferme	Hôpital de jour	(1.n)	Médecin	(1.1)	Employer-dans
Se-rend	Patient	(1.n)	Hôpital de jour	(1.n)	-
Réaliser	Médecin	(1.n)	Consultation	(1.n)	Réaliser-par
Effectuer	Infirmier	(1.n)	Analyses	(1.1)	Effectuer-par
Se-fait	Consultation	(1.n)	Hôpital de jour	(1.n)	-
Travail	Infirmier	(1.n)	Hôpital de jour	(1.n)	Renferme
Avoir	Patient	(1.1)	Carte	(1.1)	-
Nécessit	Maladie	(1.n)	Traitement	(1.n)	Spécifier
Subit	Patient	(1.1)	Consultation	(1.1)	-
Exige	Médecin	(0.n)	Analyses	(0.n)	-
Faire-part	Traitement	(0.n)	Infirmier	(0.n)	Fair
Prescrit	Médecin	(1.n)	Traitement	(1.1)	-
nécessite	Consultation	(0.1)	Traitement	(1.1)	-

Tableau2. 4.Table des relations binaires.

Chapitre 2 Construction de l'ontologie diabète

Pour chaque relation nous définissons : son nom, le nom du concept source, le nom du concept cible, la cardinalité et le nom de la relation inverse. Le tableau 2.4 illustre la spécification des relations binaires entre les différentes hiérarchies pour notre ontologie.

3.4.5. Tableau des attributs

Les attributs sont des propriétés qui prennent leurs valeurs dans les types prédéfinis (String, Intègre, Booléen, Date...). Par exemple le concept Patient a comme attributs : Nom, Prénom, âge...

Pour chaque attribut apparaissant dans le dictionnaire de concepts nous spécifions: son nom, type et intervalle de ses valeurs possibles, et sa cardinalité (pour spécifier qu'une instance possède une ou plusieurs valeurs). Le tableau 2.5 spécifie ces informations pour chaque attribut.

Nom de l'attribut	Type	Cardinalité (min/max)	Valeur par défaut	Domaine des valeurs
Nom	String	(1,1)	-	-
Prénom	String	(1,1)	-	-
Tel	Intègre	(1,1)	-	-
Adresse	String	(1,1)	-	-
Age	Intègre	(1,1)	-	-
Date-dér-visite	Date	(1,1)	-	-
Nom-maladie	String	(1,1)	-	-
Résultat-De-Consultation	String	(1,1)	-	-
Date-consultation	Date	(1,1)	-	-

Dénomination	String	(1,1)	-	-
Adresse	String	(1,1)	-	-
Tel- Hôpital	Intègre	(1,1)	-	-
Fax- Hôpital	Intègre	(1,1)	-	-
Nom- Médicament	String	(1,1)	-	Insuline
Résultat- Analyses	String	(1,1)	-	Positif
Type	String	(1,1)	-	-
Sexe	String	-	-	-
.....

Tableau2. 5.Table des attributs.

3.4.6. Tableau des axiomes logiques

Ces tableaux contiennent des définitions de concepts à l'aide des expressions logiques qui sont toujours vraies. Dans ce tableau nous définissons pour chaque axiome sa description en langage naturel, le nom du concept auquel l'axiome se réfère, les attributs utilisés dans l'axiome et l'expression logique. Pour notre ontologie nous spécifions quelques axiomes comme il est représenté dans le Tableau 2.6.

Nom du concept	Description	Expression logique
Personne	Chaque personne du service Diabétique est soit un patient, un agent de santé, psychologue.	$\forall (X), \text{Personne}(X) \Rightarrow \text{agent de santé}(X) \vee \text{psychologue}(X) \vee \text{Médecin}(X).$
Patient	Un patient doit au moins se rendre dans un Hôpital de jour être examiné par un médecin et reçoit un traitement.	$\forall (X), \text{patient}(X) \Rightarrow \exists Y \text{ Hôpital de jour}(Y) \wedge \text{Se-Rend}(X, Y) \wedge \exists Z \text{ médecin}(Z) \wedge \text{Examiné-par}(Z, X) \wedge \exists W \text{ traitement}(W) \wedge \text{reçoit}(X, W)$
Médecin	Un médecin travaille dans un Hôpital de jour, examine des patients, réalise consultation et	$\forall (X), \text{médecin}(X) \Rightarrow \exists Y \text{ Hôpital de jour}(Y) \wedge \text{Travaille}(X, Y) \wedge \exists Z$

	demande l'aide de Infirmier, prescrit le traitement, exige les analyses et fait la mise jour de la carte.	$consultation(Z) \wedge réalise(X, Z) \wedge \exists K \text{ Infirmier}(K) \wedge demande\ aide(X,K) \wedge \exists N \text{ Analyses}(N) \wedge exige(X,N) \wedge \exists M \text{ traitement}(M) \wedge prescrit(X,M) \wedge \exists S \text{ carte}(S) \wedge mise\ a\ jour(X,S)$
Traitement	Le Traitement est spécifier une maladie et doit donné a un patient.	$\forall (X), \text{ Traitement}(X) \Rightarrow \exists K \text{ maladie}(K) \wedge spécifier(X, Y) \wedge \exists S \text{ patient}(S) \wedge donné-a(X, S)$
Maladie	Peut-être Type1, Type2.	$\forall (X), \text{ Maladie}(X) \Rightarrow \exists Y \text{ Type1}(Y) \vee \exists Z \text{ Type2}(Z)$
Hôpital	Hôpital est contient des services de 5 services un service d'infection, service Médecine interne, service Hôpital de jour, service CTS, service Hémodialyse.	$\forall (X), \text{ Hôpital}(X) \Rightarrow \text{service d'infection}(X) \wedge \text{Médecine interne}(X) \wedge \text{Hôpital de jour}(X) \wedge \text{CTS}(X) \wedge \text{service Hémodialyse}(X)$.
Médicament	Peut-être injection Glucagon, Antidiabétique ou insuline	$\forall (X), \text{ Médicament}(X) \wedge \text{injection Glucagon}(X) \vee \text{Antidiabétique}(X) \vee \text{insuline}(X)$.
Consultation	Consultation doit réaliser par le médecin et nécessite un traitement.	$\forall (X) \text{ Consultation}(X) \Rightarrow \exists Y \text{ médecin}(Y) \wedge effectuées(Y, X) \wedge \exists S \text{ traitement}(S) \wedge nécessite(X,S)$
Les analyses	Les analyses seront exiger par le Médecin et effectuées par Infirmier.	$\forall (X), \text{ analyses}(X) \Rightarrow \exists Y \text{ Infirmier}(Y) \wedge effectuées(X, Y) \wedge \exists K \text{ Médecin}(K) \wedge exiger(K,X)$
Infirmier	Un Infirmier travaille dans Hôpital de jour, faire traitement des Patients, effectue les analyses et aide le Médecin.	$\forall (X), \text{ Infirmier}(X) \Rightarrow \exists Y \text{ traitement}(Y) \wedge faire(X, Y) \wedge \exists Z \text{ analyses}(Z) \wedge effectue(X, Z) \wedge \exists W \text{ Hôpital de jour}(W) \wedge travaille(X, W) \wedge \exists S \text{ Médecin}(S) \wedge aide(X, S)$
Psychologue	Un Psychologue travail dans Hôpital de jour et interroger avec le Patient.	$\forall (X), \text{ Psychologue}(X) \Rightarrow \exists Y \text{ Hôpital de jour}(Y) \wedge travail(X, Y) \wedge \exists C \text{ Patient}(C) \wedge interroger(X,C)$.

Tableau2. 6.Table des axiomes logiques.

3.4.7. Tableau des instances

Dans cette section nous donnons une description de quelques instances de l'ontologie. Pour cela, nous spécifierons les noms des individus et les valeurs des attributs pour chacun d'eux. Le Tableau 2.7 illustre quelques instances pour chaque classe.

Concept	Nom de l'instance	Attribut	Valeur
Médecin	Selma Boulayoun	Nom- Médecin	Selma
		Prénom- Médecin	Boulayoun
		Tel- Médecin	0777777777
		Adresse- Médecin	Cité Frères Bendib Mila
		Age- Médecin	45
		Sexe	Femme
Hôpital	Frères Tobal	Dénomination	Frères Tobal
		Adresse	Commune centre ville Mila
		Tel	03100000000
		Fax	03100000001
Patient	Imen Nabeti	Nom-patiente	Imen
		Prénom-patiente	Nabeti
		Tel-patiente	0666656463
		Adresse-patiente	Ain tine
		Age-patiente	35
Maladie	Type1	Nom-maladie	Type1
Chef service		Nom	Messoud
		Prénom	Djebli
			0774757678

		Tel Adresse Age	Constantine 50
Consultation	Vérification de billon	Nom-consultation	Vérification de billon
Traitement	Soigné de plait	Nom-Traitement	Soigné de plait
.....

Tableau2. 7.Table des instances.

3.5. Etape de formalisation

Dans cette étape, nous utilisons le formalisme des logiques de descriptions pour formaliser le modèle conceptuel que nous avons obtenu dans l'étape de conceptualisation.

3.5.1 Construction de TBox

Nous définissons les concepts et les rôles relatifs à notre domaine, en utilisant les constructeurs fournis par les logiques de descriptions pour donner des descriptions structurées aux concepts et rôles.

De plus, nous spécifions les relations de subsomption qui existent entre les différents concepts, par exemple pour spécifier que la classe patient est subsumée par la classe personne on écrit :

$$\text{Patient} \sqsubseteq \text{Personne}$$

Concept	Définition	Relation de subsomption
Personne	Personne := Patient \cup Agent santé \cup Psychologue	Personne \sqsubseteq Thing
Médecin	<i>Médecin</i> := Agent santé \wedge (\exists Employer-dans hôpital de jour) \wedge (\exists examine patient) \wedge (\exists réalise consultation) \wedge (\exists exige analyses) \wedge (\exists prescrit traitement)	<i>Médecin</i> \sqsubseteq Agent santé

Traitement	$Traitement := (\exists \text{spécifier maladie}) \cap (\exists \text{ donné a patient}) \cap (\exists \text{ faire-part infirmier})$	$Traitement \subseteq T$
Patient	$Patient := \text{personne} \cap (\exists \text{ se rend hôpital de jour}) \cap (\exists \text{ examine par médecin}) \cap (\exists \text{ reçoit Traitement}) \cap (\exists \text{ souffre Maladie}) \cap (\exists \text{ Avoir carte}) \cap (\exists \text{ faire analyses}) \cap (\exists \text{ subit consultation})$.	$patient \subseteq \text{Personne}$
Hôpital de jour	$Hôpital\ de\ jour := (\geq 1 \text{ Renferme. Agent santé}) \cap (\geq 1 \text{ Se-rend Patient})$	$Hôpital\ de\ jour \subseteq \text{Hopital}$
Infirmier	$\text{Infirmier} := \text{Médecin} \cap (\exists \text{ faire traitement}) \cap (\exists \text{ effectue analyses}) \cap (\exists \text{ travaille Hôpital de jour}) \cap (\exists \text{ aide Médecin})$	$\text{Infirmier} \subseteq \text{Agent santé}$
Agent santé	$\text{Agent santé} := \text{Infirmier} \cup \text{médecin} \cup \text{chef-service}$	$\text{agent santé} \subseteq \text{Personne}$
.....

Tableau2. 8.Définition des concepts et subsomption.

En ce qui concerne les rôles, nous les définissons en donnant les couples des concepts sources et cibles de chacune, et/ou en spécifiant son rôle inverse.

Rôle	Couple (domaine, Co-domaine)	Rôle inverse
Souffre	(Patient, Maladie)	-
Reçoit	(Patient, Traitement)	Donné-a
Examine	(Médecin, Patient)	Examine-par
Demande-aide	(Médecin, Infirmier)	Aide
Effectue	(Infirmier, Analyses)	Effectue-par
Travaille	(Infirmier, Hôpital de jour)	Renferme
Avoir	(Patient, carte)	-
Réalise	(Médecin, Consultation)	Réaliser-par
Nécessite	(Maladie, Traitement)	Spécifier
Employer-dans	(Médecin, Hôpital de jour)	Renferme
Subit	(Patient, Consultation)	-

Exige	(Médecin, Analyses)	-
Faire	(Infirmier, Traitement)	Faire-par
Se-fait	(Consultation, Hôpital de jour)	-
Se-rend	(Patient, Hôpital de jour)	-

Tableau2. 9. Les définitions des différents rôles.

3.5.2. Construction de ABox

Le langage assertionnelle est dédié à la description des faits, en spécifiant les individus (avec leurs classes) et les relations entre eux de la manière suivante :

A : C Pour dire que A est une instance de la classe C.

Exemple : Daas Housin : Infirmier

(A1, A2) : R Pour dire que les deux individus A1 et A2 sont reliés par la relation R

Exemple : (diabétique type1, Surhomme Glucagon) : Nécessite

Le tableau 2.10 définit quelques assertions de concepts, et le tableau 2.11 définit quelques assertions de relations de notre ontologie.

Concept	Description
Médecin	Médecin : Ben saleh Amar.
Hôpital	Hôpital : Frères Tobal
Infirmier	Infirmier : Daas Housin
Patient	Patient : Nebti Imen
	Patient : Azizi Elyas
Les analyses	Les analyses : Chimie des urines Les analyses : Glycémie Post Prandial Les analyses : La créatinine
Maladie	Maladie: Type1
	Maladie: Diabète Gestationnel

Psychologue	Psychologue :Nedjar Adel
Agent de santé	Agent d'accueil : Ben cherchem Hossem

Tableau2. 10.Description des assertions de concepts.

Relation	Description
Se-rend	(Azizi Elyas,FrèresTobal) : Se-rend
Effectue	(Daas Housin, Chimie des urines) : Effectue
Réalise-par	(Consultation, Ben saleh Amar) : Réalise-par
Examine-par	(Nebti Imen, Ben saleh Amar) : Examine-par
Employer-dans	(Ben saleh Amar,FrèresTobal): Employer-dans
Nécessite	(DID,Insulin): Nécessite
Faire	(Azizi Elyas, Créatinine): Faire
Soufre	(Azizi Elyas, diabète de type1) : Soufre
Renfermé	(FrèresTobal, Daas Housin) : Renfermé

Tableau2. 11.Description des assertions de relations.

3.6. Implémentation

Après avoir faire les étapes précédents spécification des besoins, conceptualisation et formalisation Nous devons construire l'ontologie Diabète Une ontologie dans le domaine médical, spécialement dans le service de Diabète pour structurer les connaissances, les organiser, et plus précisément raisonner dessus. Pour y arriver, nous devons utiliser le langage OWL pour la création des concepts, relation, Instances et attributs dans l'outil de représentation d'ontologie le Protégé3.4.4.

4. Conclusion

Nous commençons ce chapitre par une Définition détaillée sur les étapes de construction de notre ontologie dans le domaine médicale et nous avons donné une petite description de notre domaine de diabète. Ainsi nous nous sommes appliqué les étapes de

Chapitre 2 Construction de l'ontologie diabète

construction d'ontologie : spécification des besoins, conceptualisation, Formalisation, implémentation, Evolution et vérification.

Dans le prochain chapitre nous allons présenter une conception détaillée sur notre application de gestion.

Chapitre 3



1. Introduction

Après avoir établi une étude complète sur l'ontologie ce chapitre est consacré à la conception de l'application tout en présentant le langage modélisation UML ainsi que la méthode simple et générique comme Méthode de conception.

L'objectif de ce projet est d'analyser, concevoir et développer une application de gestion d'ontologie. Cette application doit permettre, en premier lieu, l'automatisation de système et En deuxième lieu, elle permettre la gestion de l'ontologie.

2. Proposition d'une architecture de l'application de gestion de l'ontologie

Après la construction de l'ontologie, nous allons s'intéresser au développement d'une application qui permet au gestionnaire de faire des mis à jour de cette ontologie. De ce fait, nous avons proposé une architecture de l'application de gestion de l'ontologie dont les composants sont les suivants :

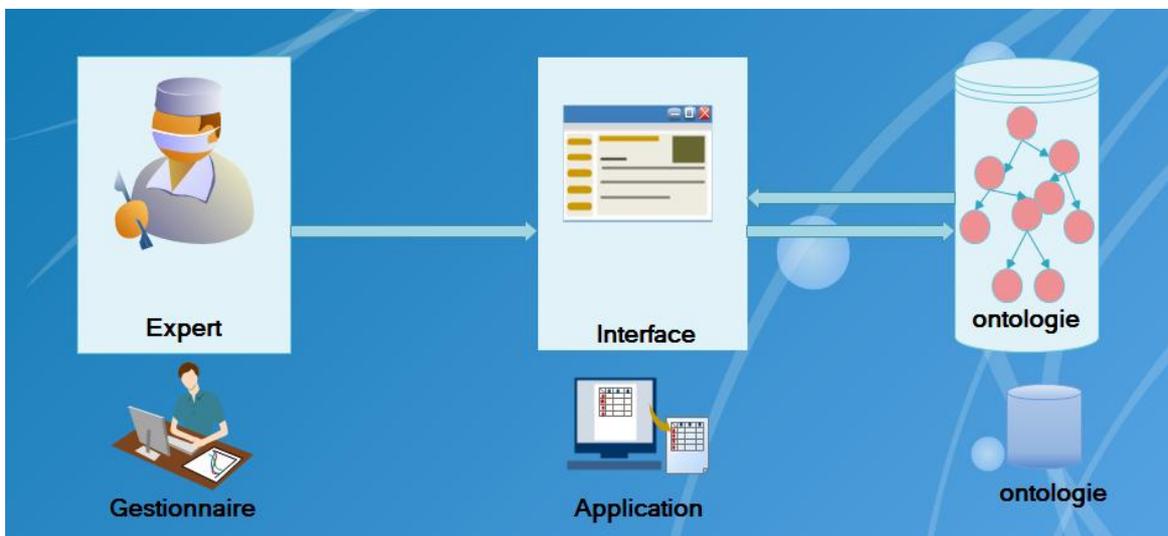


Figure3. 1. Architecture de l'application de gestion de l'ontologie proposé.

- 1) **Le gestionnaire** : est un personne responsable d'actualiser les informations concernant un domaine médical. Ce dernier peut faire la gestion de l'ontologie a partir d'une interfaces simple qui affiche tous les opérations qui peut faire sur notre ontologie.
- 2) **L'Application** : une application qui permet la gestion d'une ontologie par le gestionnaire Ce dernier peut faire les mis à jour de l'ontologie par l'ajout d'un concept, relation, individu ou attribut et peut aussi les supprimer.
- 3) **L'ontologie** : Elle inclut l'ensemble des informations concernant un domaine précise.

3. Langage de modélisation et méthode de conception

3.1. Définition

UML (Unified Modeling Language), que l'on peut traduire par (Langage de Modélisation Unifié), est une notation permettant de modéliser un problème de façon standard. Ce langage est né de la fusion de plusieurs méthodes existant auparavant, et est devenu désormais la référence en termes de modélisation objet.

UML se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier, concevoir des solutions et communiquer des points de vue. Il unifie les notations nécessaires aux différentes activités d'un processus de développement et offre, par ce biais, le moyen d'établir le suivi des décisions prises, depuis la définition des besoins jusqu'au codage.

UML permet de représenter un système selon différentes vues complémentaires : les diagrammes [33].

3.2. Choix d'une méthode d'analyse et de conception

Un projet informatique, quelle que soit sa taille et la portée de ses objectifs, nécessite la mise en place d'un planning organisationnel tout au long de son cycle de vie. C'est ainsi qu'est apparue la notion de méthode.

Une méthode, dans le contexte informatique, peut être définie comme une démarche fournissant une méthodologie et des notations standards qui aident à concevoir des logiciels de qualité. Pour cela nous avons choisi la Méthode simple et générique qui se situe à mi-chemin entre UP (Unified Process) et XP (Xtreme Programming), cette méthode contient Trois phases :

- ✓ Identification des besoins
 - Diagramme de cas d'utilisation
 - Diagramme de séquence système
 - Maquette de l'IHM

- ✓ Phase d'analyse
 - Modèle du domaine
 - Diagramme de classes participantes
 - Diagramme d'activités de navigation

- ✓ Phase de conception
 - Diagrammes d'interaction
 - Diagramme de classes de conception

4. Aperçu de La méthode simple et générique

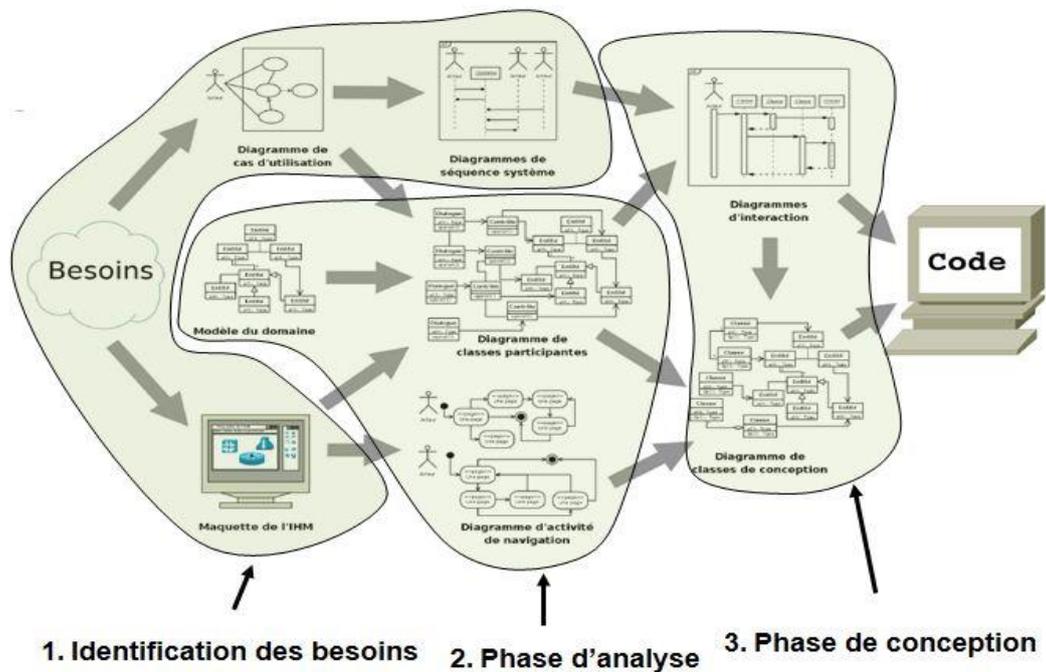


Figure 3. 2. La méthode simple et générique [34].

4.1. Phase identification des besoins

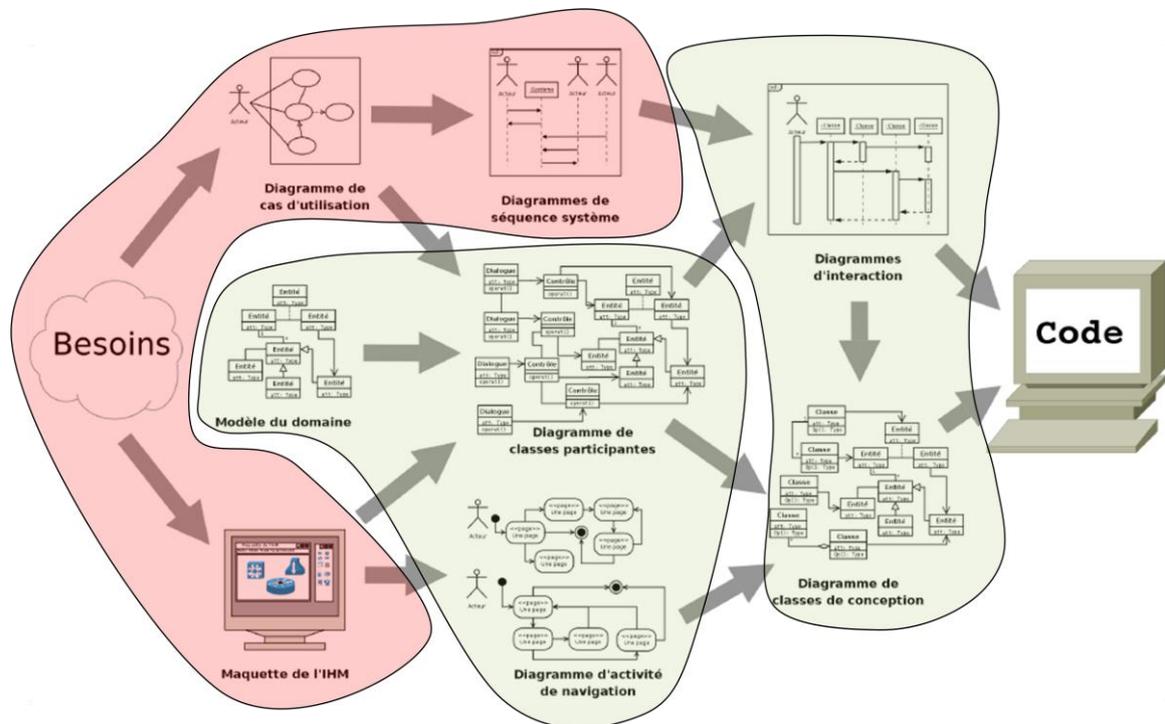


Figure 3.2.Phase Identification du besoin.

4.1.1. Besoin

Pour bien comprendre le fonctionnement du logiciel, et son interaction avec son environnement, nous avons intérêt à travailler non pas sur le logiciel demandé, mais sur le fonctionnement intégral du processus de service diabète. Ainsi nous aurons une meilleure approche du logiciel. Ainsi nous considérerons l'acteur gestionnaire qui peut être un médecin ou infirmier.

4.1.2. Diagramme de cas d'utilisation

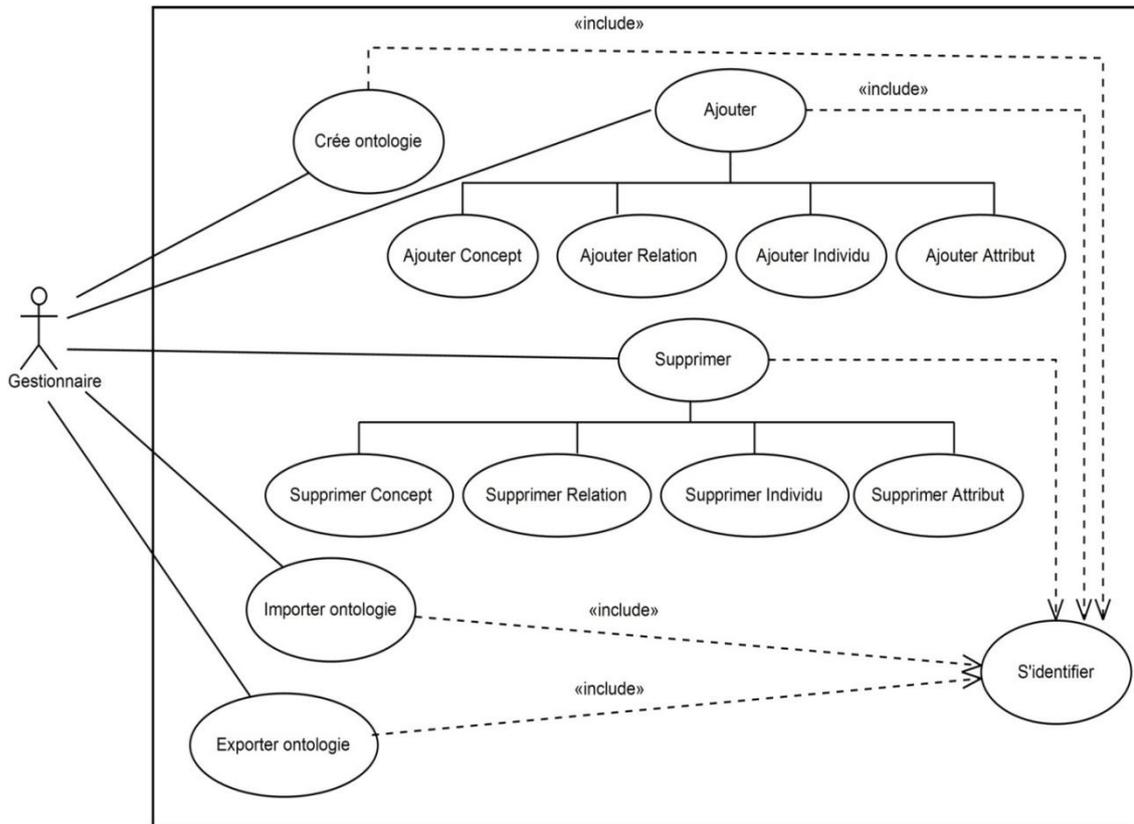


Figure3. 3. Diagramme de cas d'utilisation

4.1.3. Diagramme de séquence de système

Chaque cas d'utilisation est décrit textuellement de façon détaillée, mais donne également lieu à un diagramme de séquence représentant graphiquement la séquence des interactions entre les acteurs et le système vu comme une boîte noire, dans le cadre du scénario nominal. Nous appellerons ce diagramme : diagramme de séquence système [34].

-Supprimer Individu :

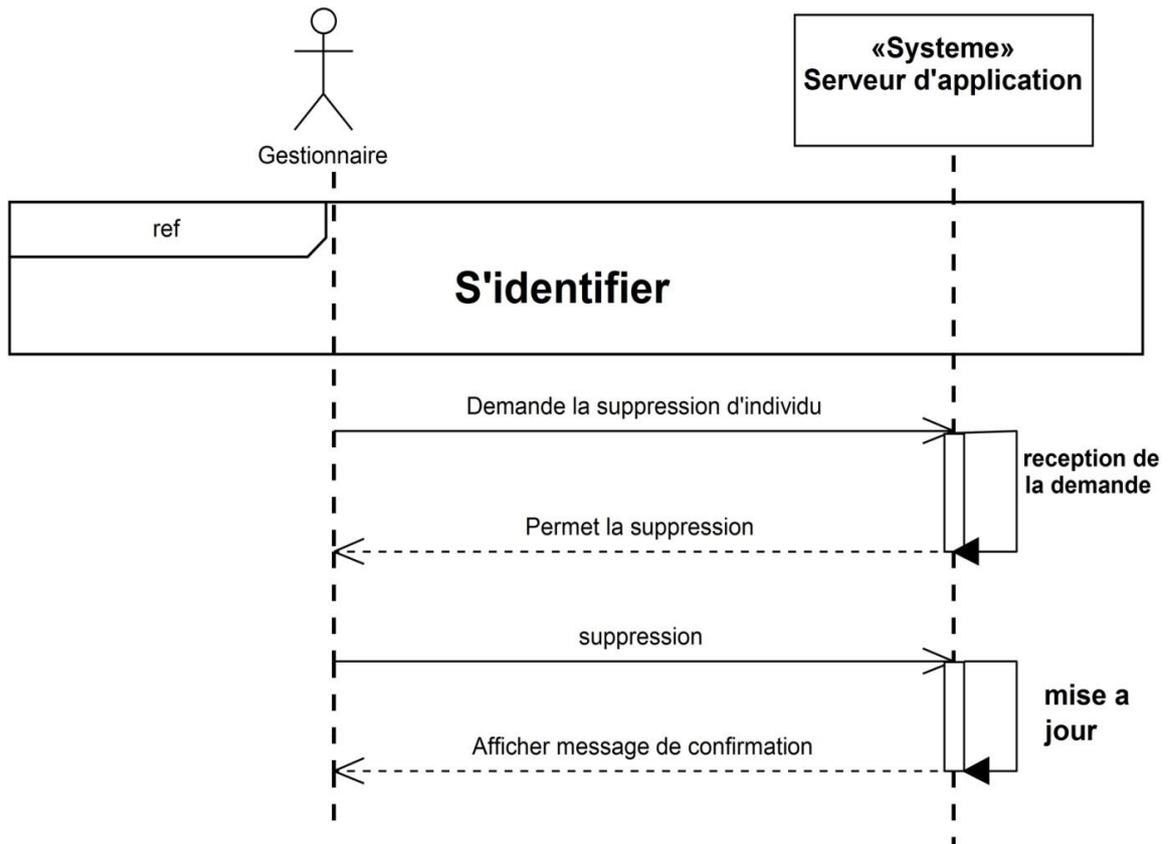


Figure 3. 4. Diagramme de séquence de système.

4.1.4. Maquette de l'IHM

Une maquette d'IHM (Interface Homme-Machine) est un produit jetable permettant aux utilisateurs d'avoir une vue concrète mais non définitive de la future interface de l'application.

4.2. Phase d'analyse

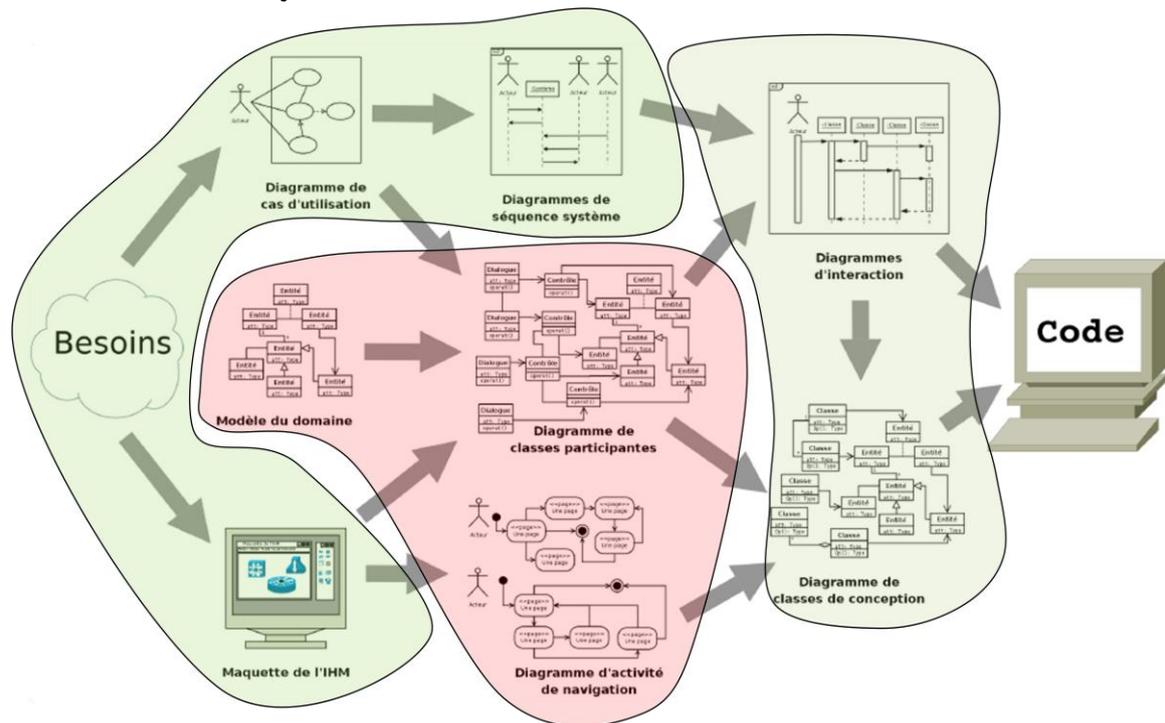


Figure 3.5. Phases d'analyse.

4.2.1 Modèle du domaine

Les classes logicielles représentant l'informatisation des concepts métier manipulés par les experts du domaine et les utilisateurs sont assez directement trouvées par une analyse du domaine. Pour matérialiser cette analyse, nous allons construire un modèle du domaine, sorte de glossaire détaillé et formalisé en UML des concepts fondamentaux de l'espace du problème. Ces concepts, leurs attributs et leurs relations vont être décrits en UML par un diagramme de classes simplifié utilisant des conventions particulières. Comme indiqué sur la figure suivante, le modèle du domaine fournit une partie des classes de conception, celles correspondant directement aux concepts métier. Il découle des cas d'utilisation et de l'analyse des besoins [34].

Le modèle du domaine issu de notre analyse comprend 6 classes et associations qui sont :

La classe Gestionnaire.

La classe Concept.

La classe Individu.

La classe Relation.

La classe Sous concept.

Chapitre3 Conception de l'application de gestion de l'ontologie

La classe Attribut.

Nous avons comme attributs possible dans chaque classe :

Gestionnaire : nom, prénom, user Name, mot de passe

Concept : nom de concept.

Individu: nom d'individu.

Relation : nom de relation.

Sous concept : nom de sous concept.

Attribut : nom d'attribut.

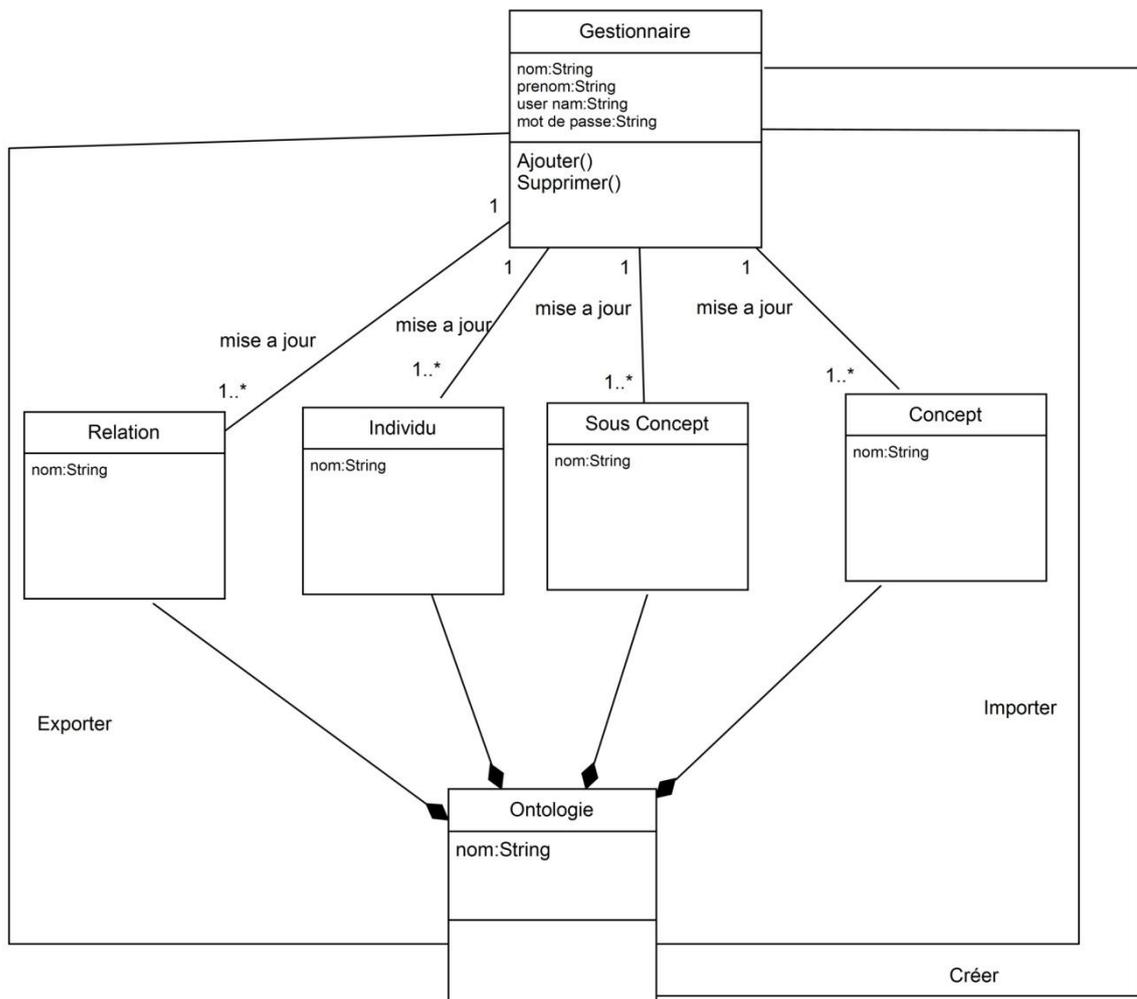


Figure 3. 3.Modèle du domaine.

4.2.2 Diagramme de classes participantes

Mais le modèle du domaine à lui seul ne permet pas d'identifier les principales classes d'IHM ni celles qui décrivent la cinématique de l'application. Le chaînon manquant

de notre démarche s'appelle les diagrammes de classes participantes. Il s'agit là encore de diagrammes de classes UML qui décrivent, cas d'utilisation par cas d'utilisation, les trois principales classes d'analyse (dialogue, contrôle, entité) et leurs relations [34].

-Ajouter concept :

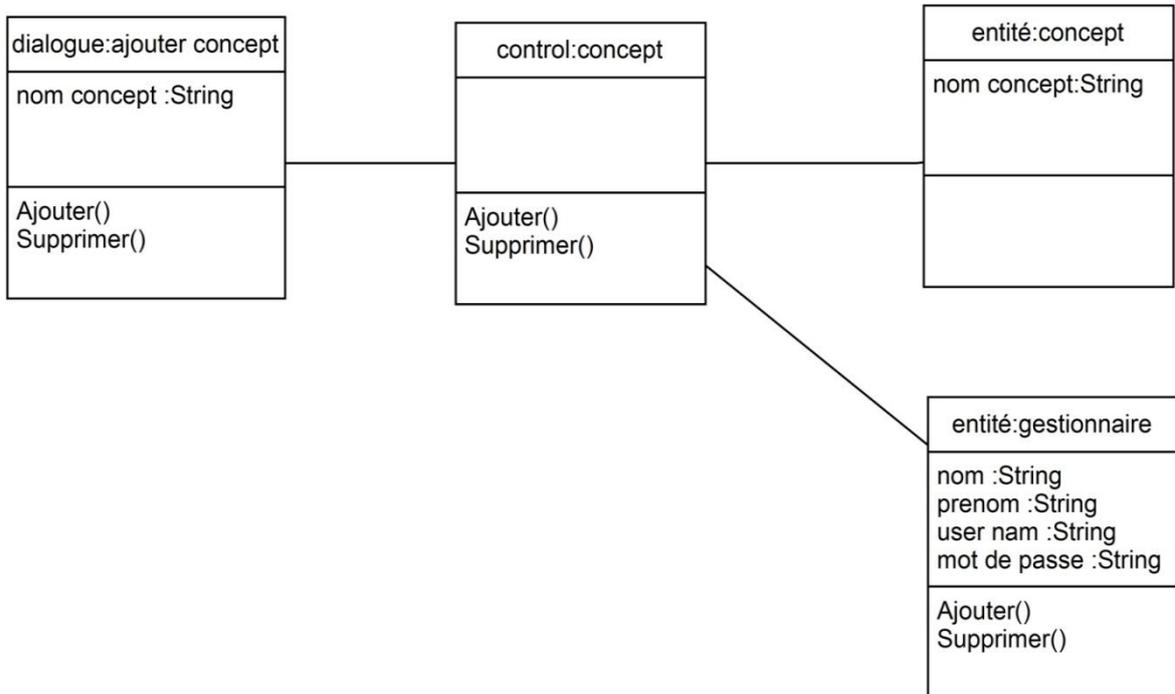


Figure 3.4. Diagramme de classe participante.

4.2.3. Diagrammes d'activités de navigation

Il nous reste à détailler une exploitation supplémentaire de la maquette. Elle va nous permettre de réaliser des diagrammes dynamiques représentant de manière formelle l'ensemble des chemins possibles entre les principaux écrans proposés à l'utilisateur. Ces diagrammes s'appellent des diagrammes d'activités de navigation [34].

- Supprimer individu :

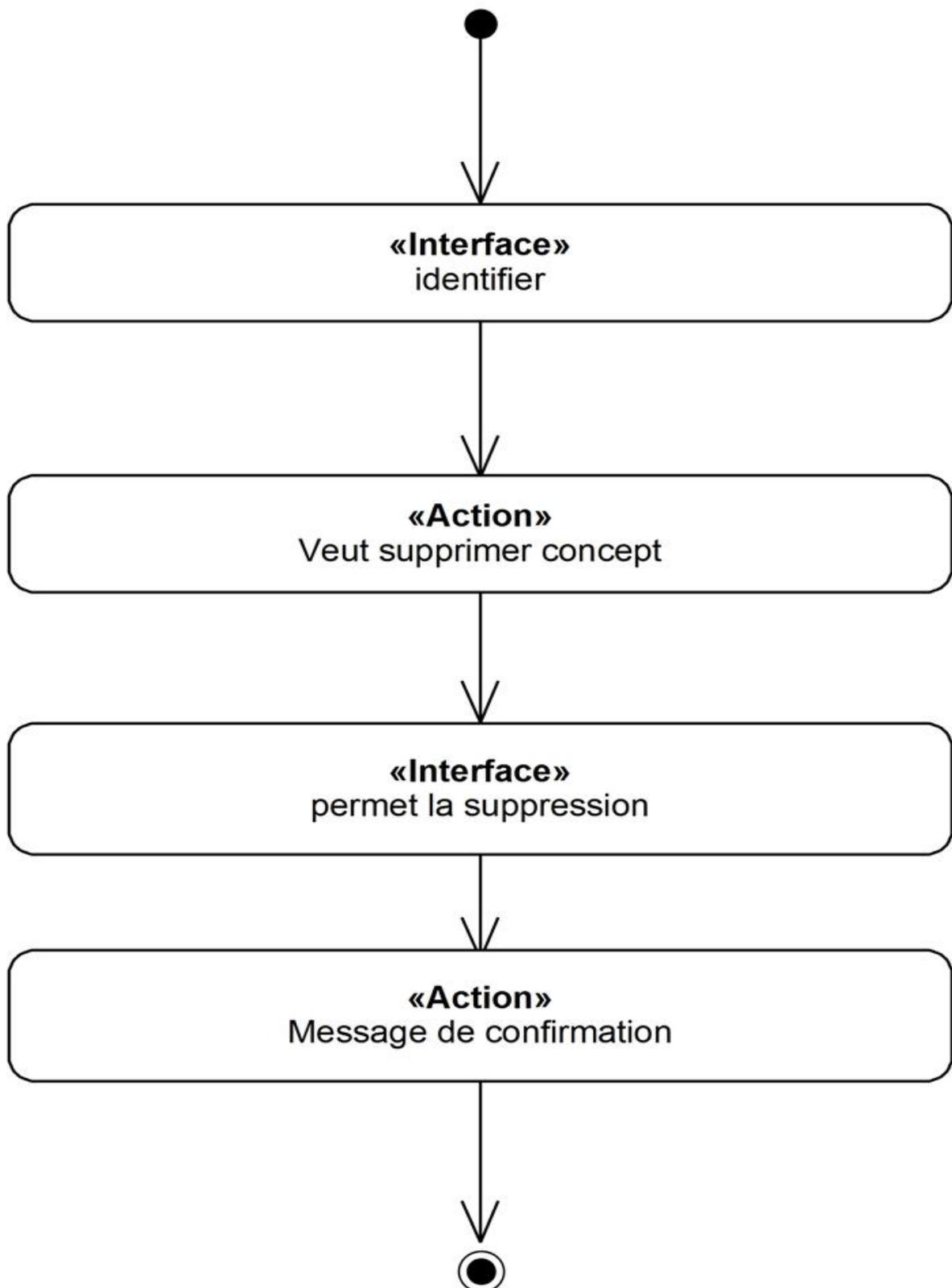


Figure 3. 5.Diagrammes d'activités de navigation.

- S'identifier :

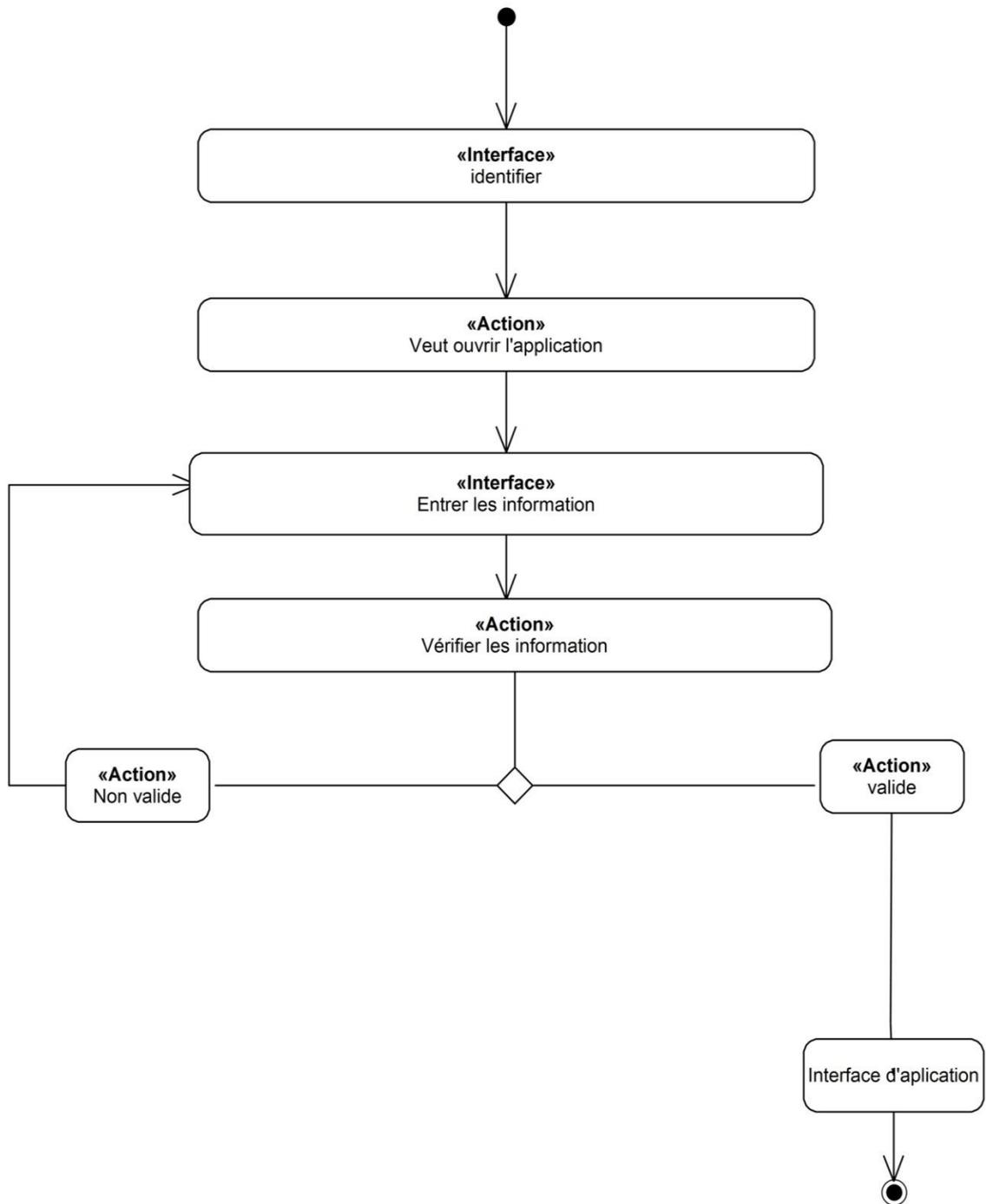


Figure3.6.Diagrammes d'activités de navigation.

4.3. Phase de conception

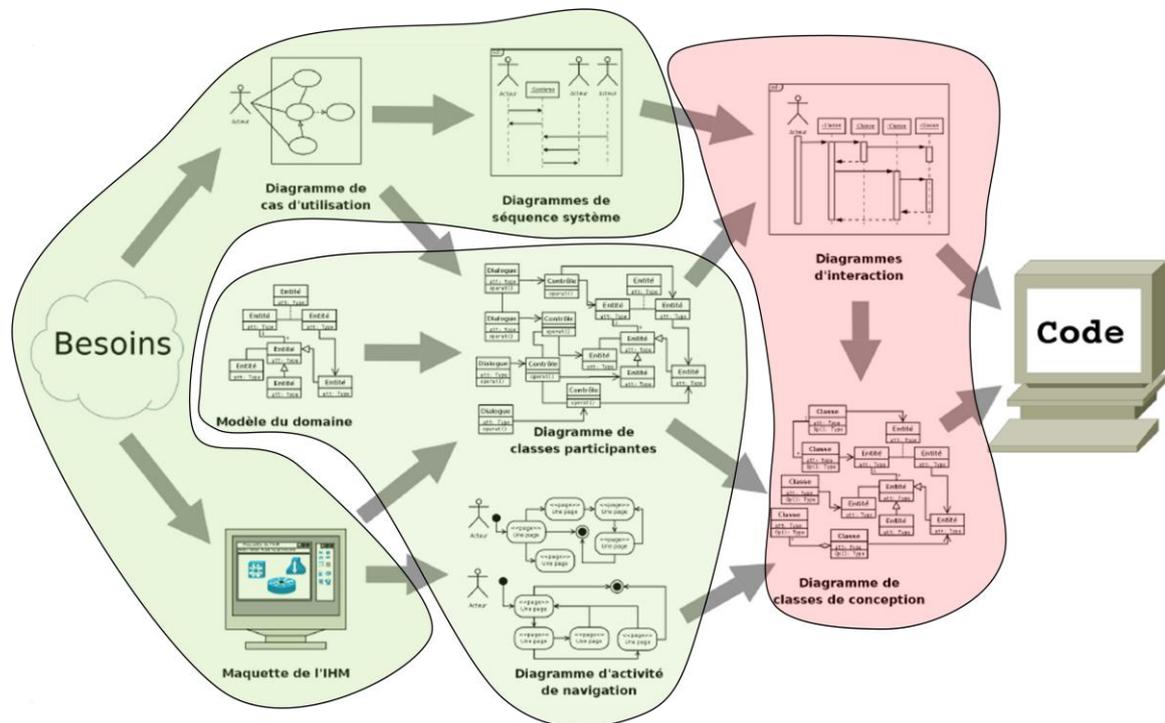


Figure 3.10. Phases de conception.

4.3.1. Diagramme d'interaction

Les diagrammes d'interactions UML sont particulièrement utiles au concepteur pour représenter graphiquement ses décisions d'allocation de responsabilités. Chaque diagramme d'interaction va ainsi représenter un ensemble d'objets de classes différentes collaborant dans le cadre d'un scénario d'exécution du système. Dans ce genre de diagramme, les objets communiquent en s'envoyant des messages qui invoquent des opérations sur les objets récepteurs [34].

-Supprimer Concept :

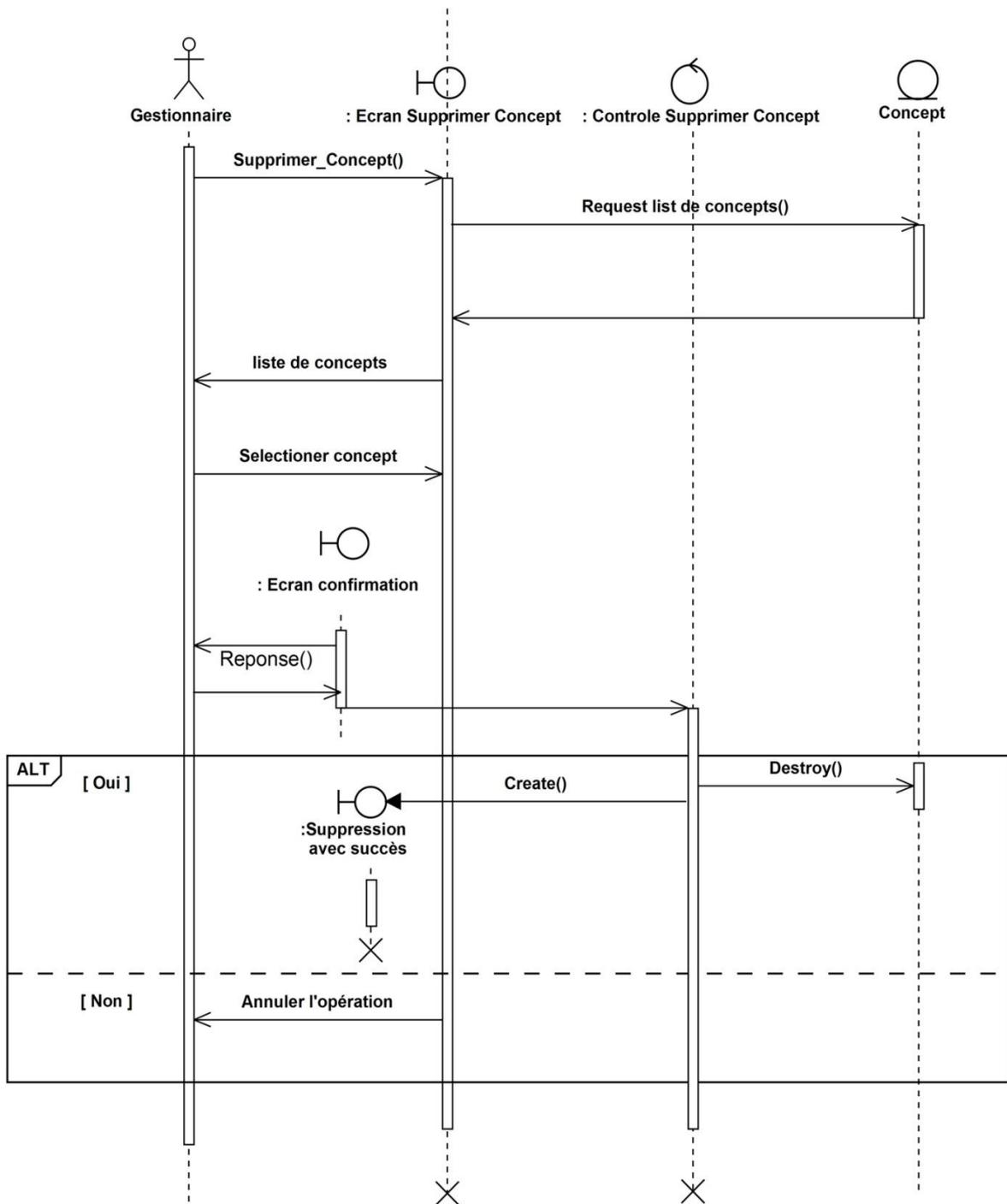


Figure 3.11. Diagramme d'interaction.

4.3.2. Diagramme de classe de conception

Dans le cadre des systèmes orientés-objets, la structure du code est définie par les classes logicielles et leurs regroupements en ensembles appelés package. Nous avons donc besoin de diagrammes représentant les classes logicielles et montrant les données qu'elles

contiennent (attributs), les services qu'elles rendent (opérations) ainsi que leurs relations. UML propose les diagrammes de classes pour véhiculer toutes ces informations [34].

-S'identifier :

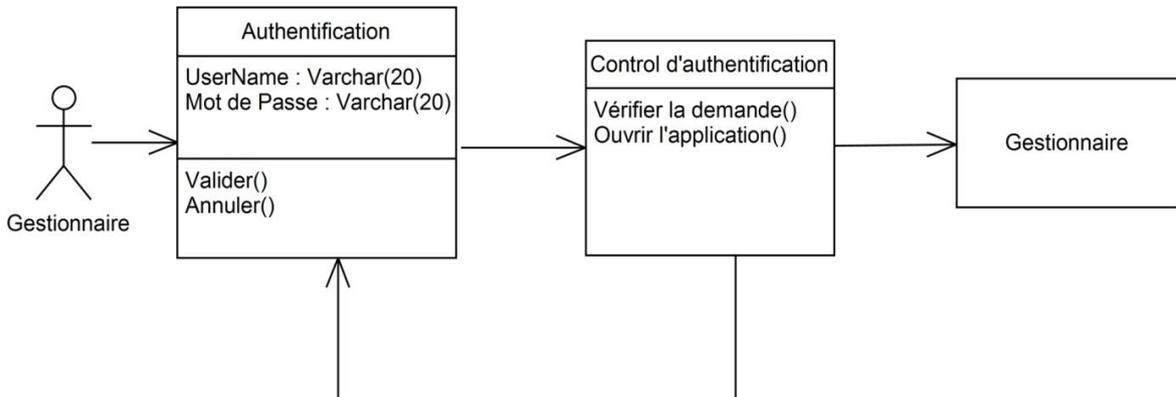


Figure3.12. Diagramme de classe de conception.

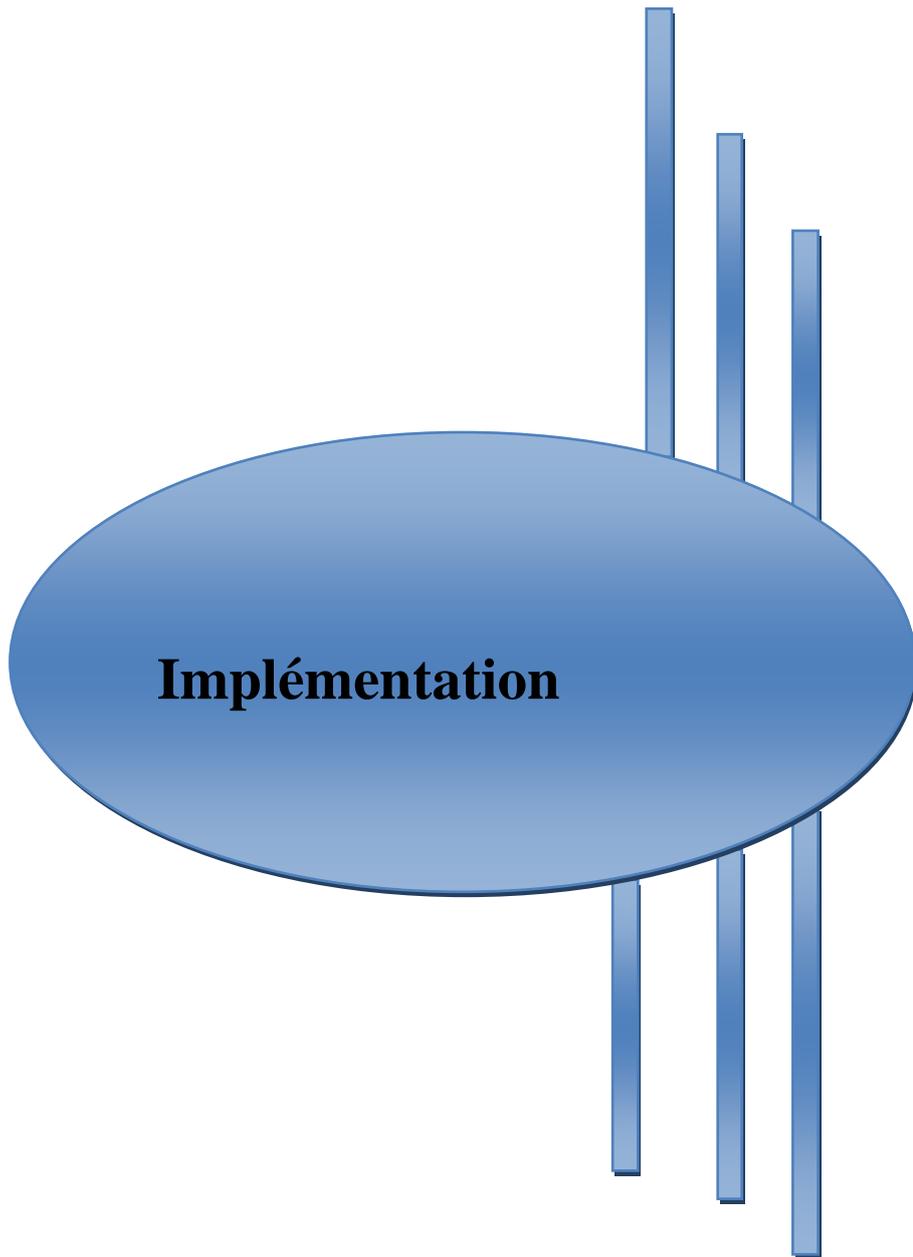
5. Conclusion

Dans Ce chapitre nous avons fait une description sur notre travail (conception de l'application).

Nous avons définie le langage de modélisation UML. Puis nous avons choisie la démarche simplifiée comme une méthode de conception d'application et leurs étapes. A partir du besoin utilisateur jusqu'à le code d'application.

Dans le prochain chapitre nous allons présenter l'implémentation de notre ontologie et application.

Chapitre 4



1. Introduction

Après avoir créé l'ontologie, il nous reste maintenant à l'implémenter. Pour cela, nous disposons de nombreux langages et outils qui permettent d'implémenter de l'ontologie et l'application proposée. Un langage de définition d'ontologie est nécessaire pour implémenter notre ontologie « diabète ». Pour cela nous avons choisi des outils standards: Protégé-OWL et le raisonneur Pellet.

Aussi nous parlons des outils utilisés pour construire notre application « santé », qui facilite l'interaction de l'utilisateur avec l'ontologie, nous avons choisi le langage java comme langage de programmations, et nous utilisons l'interface de programmations Jena pour charger le fichier de l'ontologie « diabète » avec l'extension (OWL) et nous utiliser NetBeans pour faire notre interface.

2. l'implémentation de l'ontologie

2.1. Les outils d'implémentation

2.1.1 Plateforme Protégé

PROTEGE OWL est une interface modulaire, développée au StanfordMedicalInformatics de l'Université de Stanford⁷, permettant l'édition, la visualisation, le contrôle (vérification des contraintes) d'ontologies, l'extraction d'ontologies à partir de sources textuelles, et la fusion semi-automatique d'ontologies [35]. PROTEGE OWL autorise la définition de méta-classes, dont les instances sont des classes, ce qui permet de créer son propre modèle de connaissances avant de bâtir une ontologie. De nombreux plug-ins sont disponibles ou peuvent être ajoutés par l'utilisateur. L'interface, très bien conçue, et l'architecture logicielle permettant l'insertion de plug-ins pouvant apporter de nouvelles fonctionnalités (par exemple, la possibilité d'importer et d'exporter les ontologies construites dans divers langages opérationnels de représentation tels que OWL ou encore la spécification d'axiomes) ont participé au succès de PROTEGE OWL, qui regroupe une communauté d'utilisateurs très importantes et constitue une référence pour beaucoup d'autres outils [35].

La figure (figure 4.1) présente l'interface de PROTEGE OWL.

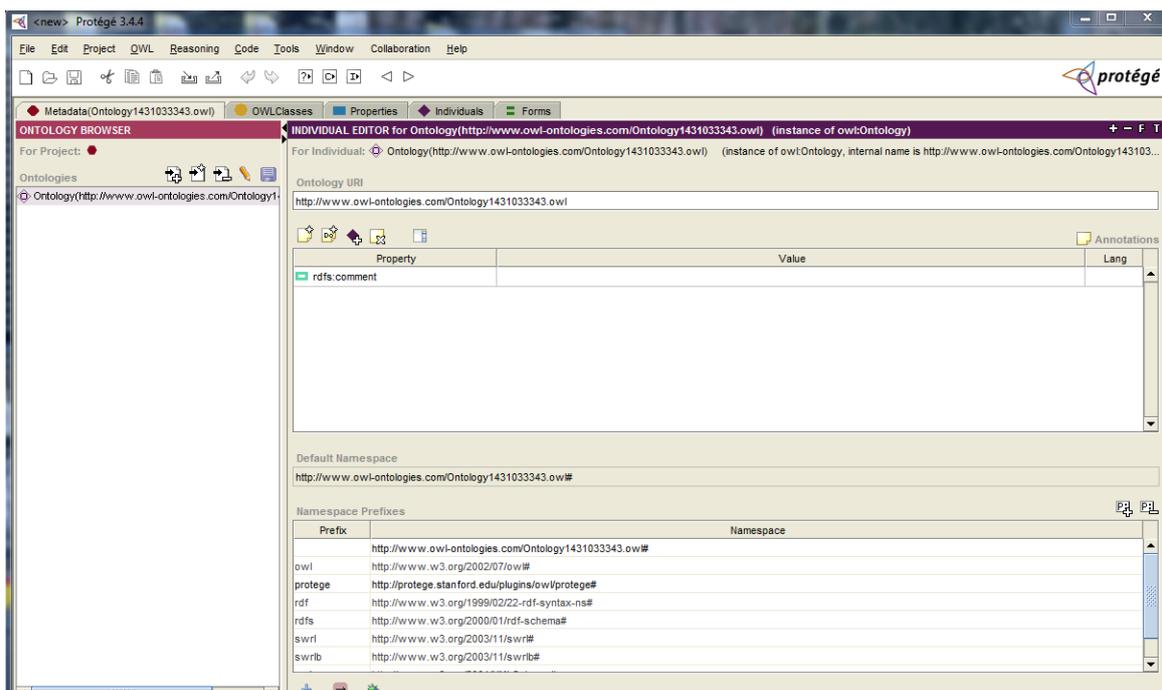


Figure4.1.L'interface principale de Protégé.

2.1.2. Le classifieur Pellet

Le raisonneur Pellet disponible directement depuis Protégé 3.4.4, Pellet est open-source, développer en Java, adapte au raisonnement sur OWL. Nous exploitons le raisonneur Pellet 1.5.2 disponible directement depuis Protégé 3.4.4.

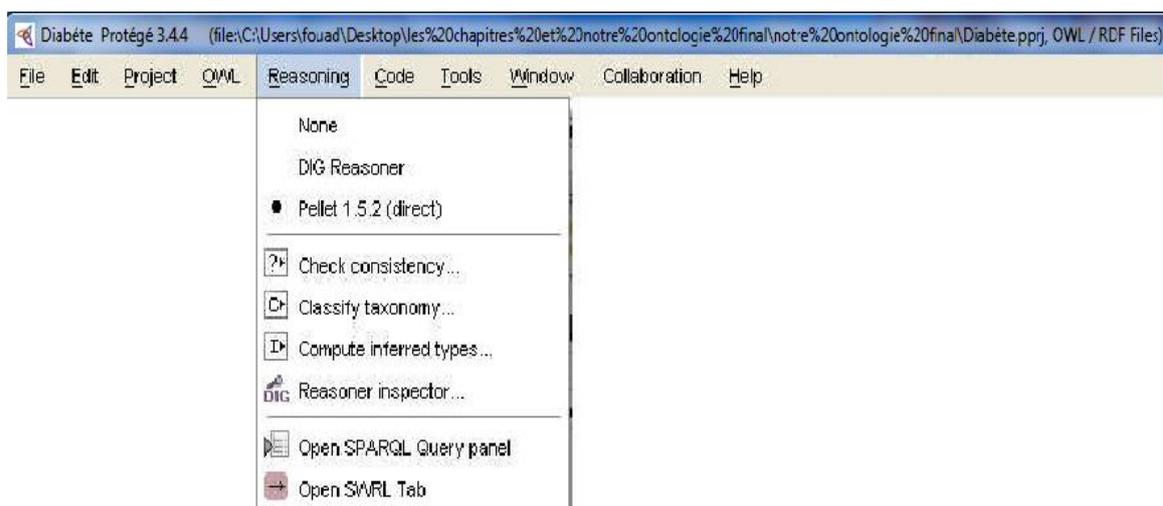


Figure4. 2.Le classifieur Pellet.

2.2 Création de l'ontologie avec Protégé OWL

2.2.1. Création des classes et hiérarchie de classes

Nous commençons tout d'abord par la création des concepts spécifiés dans l'étape de Conceptualisation. Protégé OWL nous offre le moyen de construire la hiérarchie des concepts.

La figure 4.3 présente la procédure de création d'un concept sous Protégé OWL.

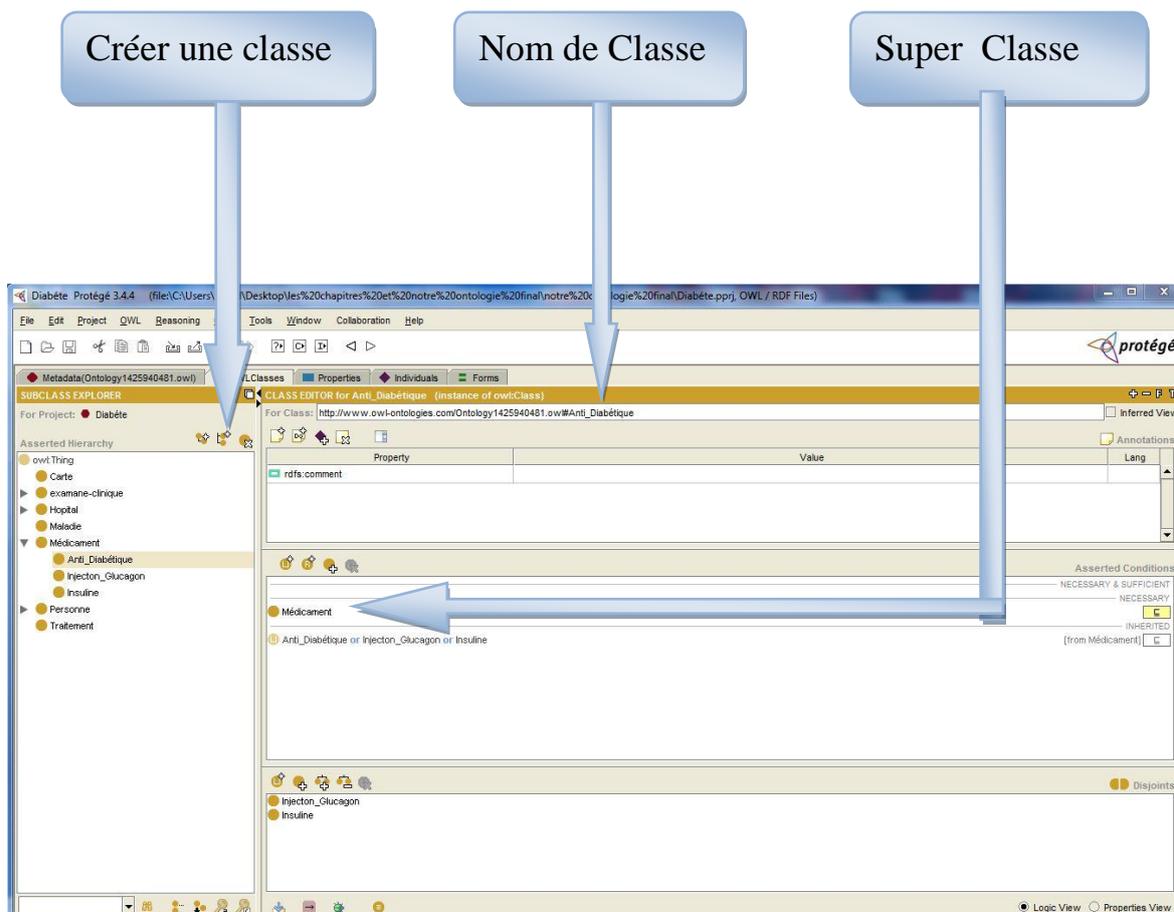


Figure4. 3.Création de classes.

Chapitre 4 Implémentation

L'ensemble de concepts créés se synthétise dans une hiérarchie comme montre la figure :

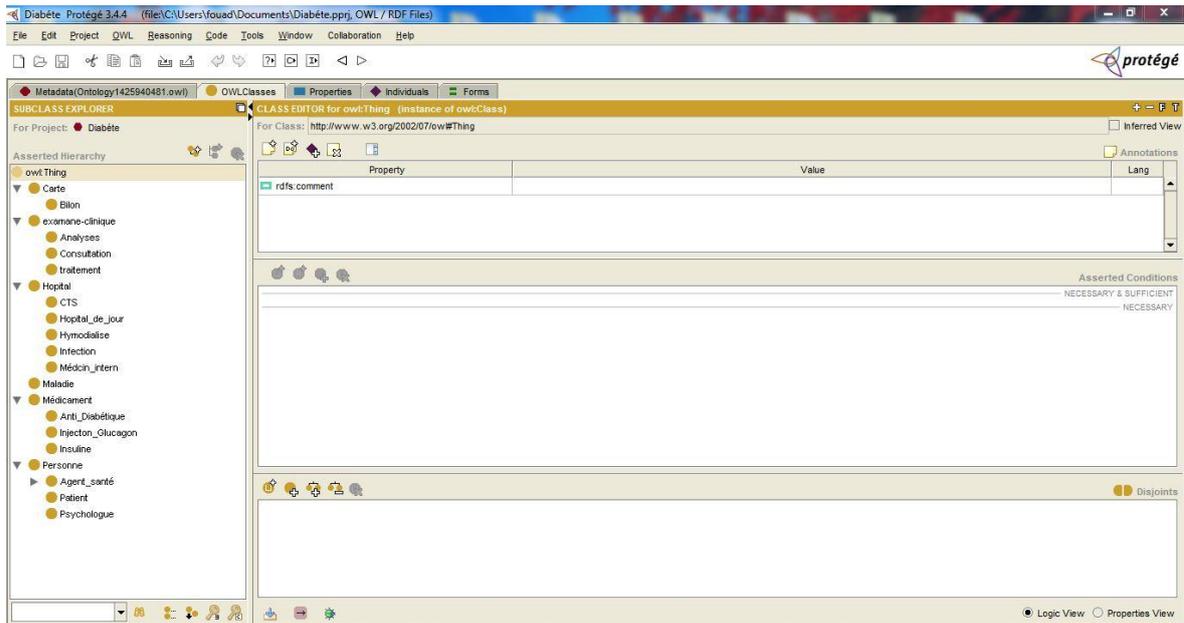


Figure4. 4.Hiérarchies des concepts.

2.2.2. Création des propriétés

Après avoir construit les concepts, nous allons maintenant créer les propriétés pour chacun d'eux, les attributs vont être créés sous PROTEGE OWL par 'dataTypeProperty' et les relations par 'objectProperty'. Les propriétés d'une classe sont les propriétés héritées de sa superclasse, plus ses propres propriétés privées.

La figure 4.5 montre les possibilités de PROTEGE OWL pour la création des propriétés.

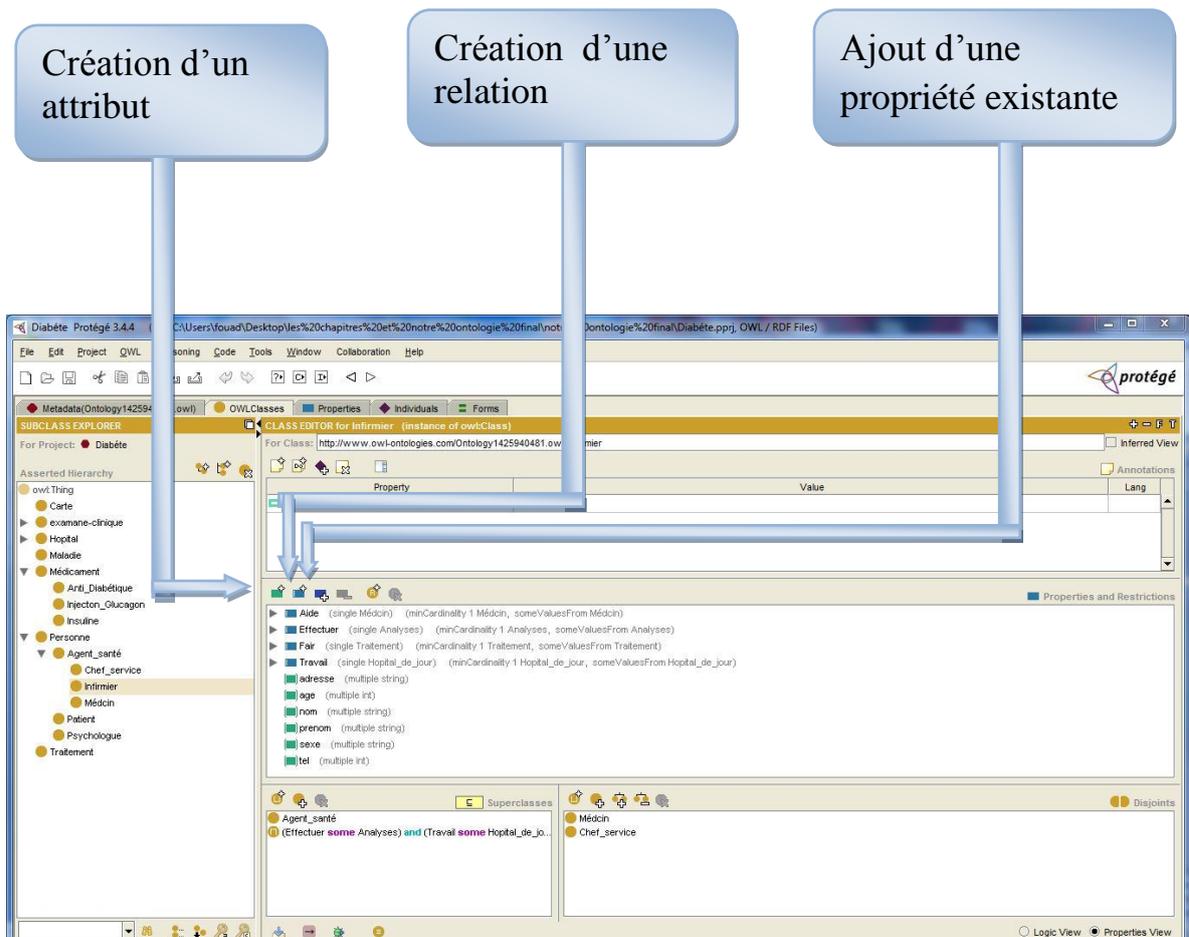


Figure 4.5. Création de propriétés pour une classe.

2.2.3. Création d'attributs (DataTypeProperty)

La création des attributs se fait en cliquant sur le bouton 'data type property', ce qui génère la fenêtre présentée par la figure 4.6. Cette fenêtre permet de fournir les champs pour remplir le nom, le domaine, et le type de l'attribut.

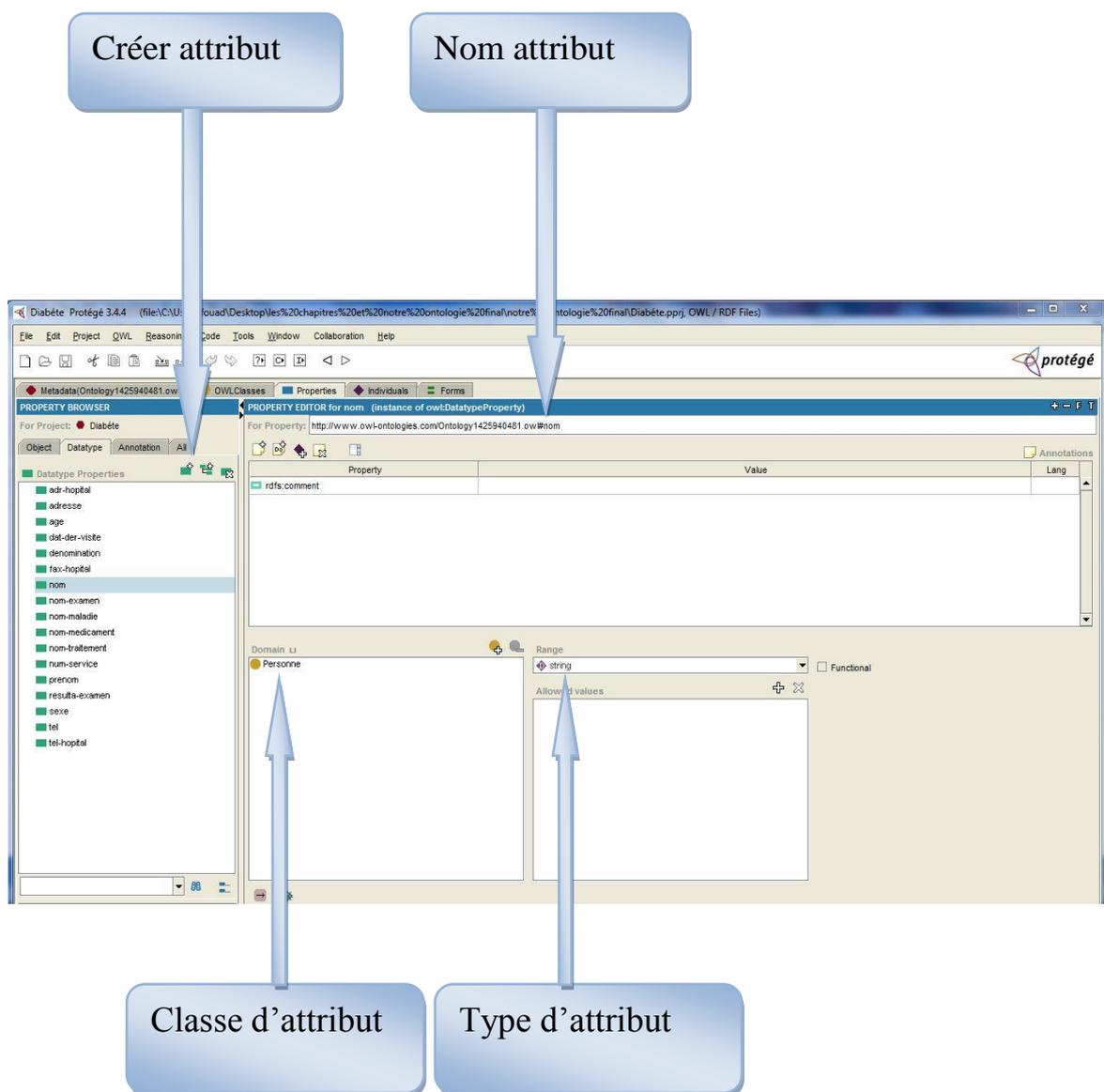


Figure4. 6.Création d'un attribut.

2.2.4. Création des relations (ObjectProperty)

Pour la création de relations, nous devons spécifier le nom de la relation, le domaine, le codomaine et la relation inverse si elle existe. Nous pouvons aussi associer à cette définition le type de la relation (symétrique, transitive,...)

La figure qui suit montre la définition de la relation "faire ".

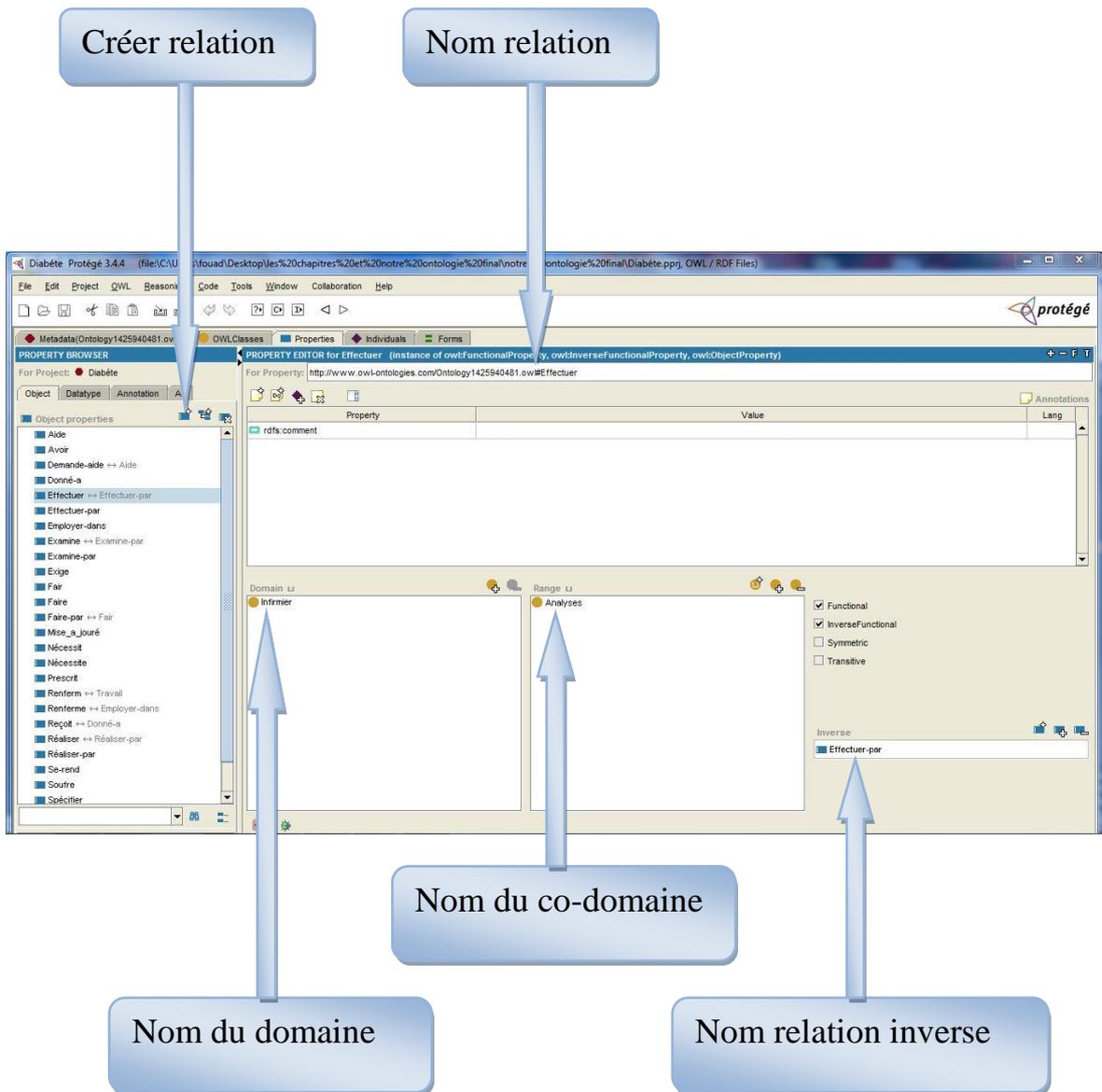


Figure4. 7.Création d'une relation entre deux classes.

2.2.5. Restrictions sur les propriétés

Les restrictions définissent des classes anonymes, les membres de ces classes sont les individus qui vérifient les conditions, les restrictions sont souvent utilisées pour définir les conditions nécessaires pour une classe.

La figure (4.8) ci-dessous montre comment créer une restriction.

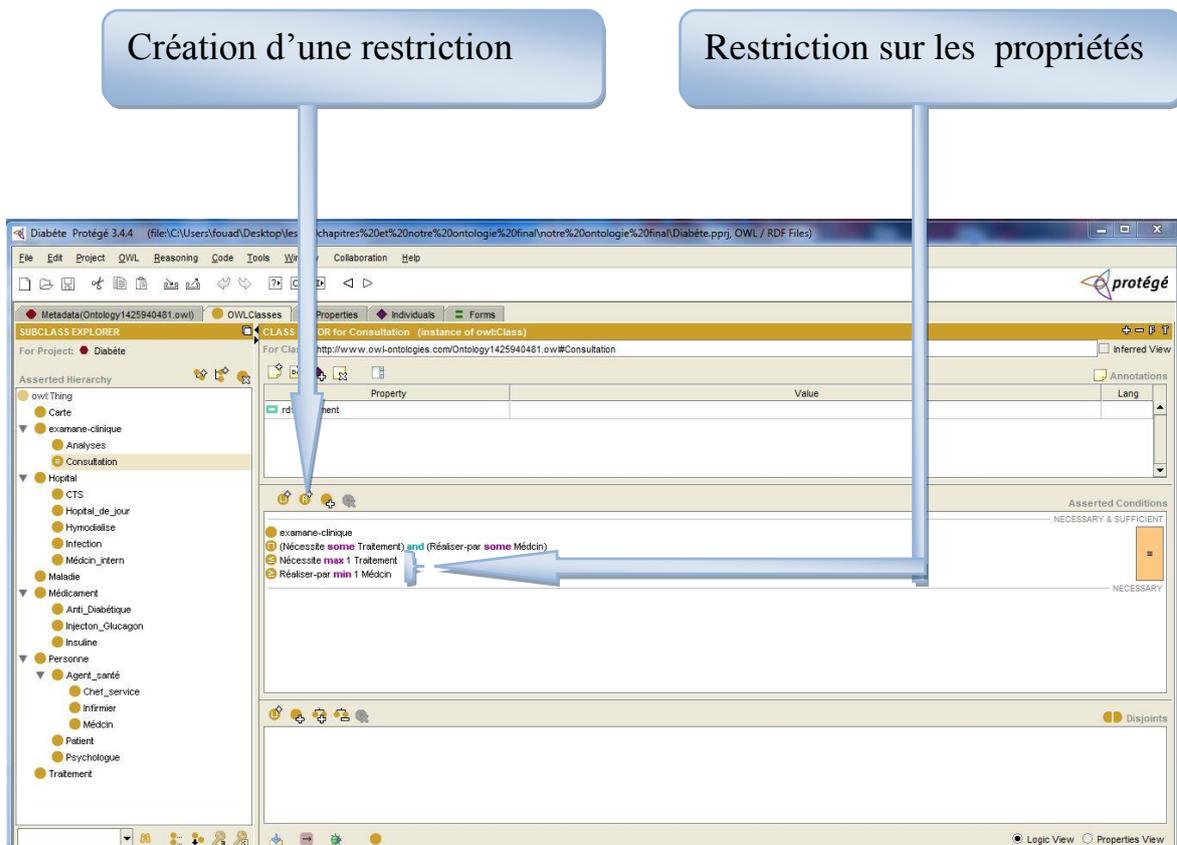


Figure4. 8. Les restrictions créées sur les propriétés d'une classe.

Si vous cliquez au-dessus du bouton (), Protégé OWL vous afficherait la fenêtre suivante :

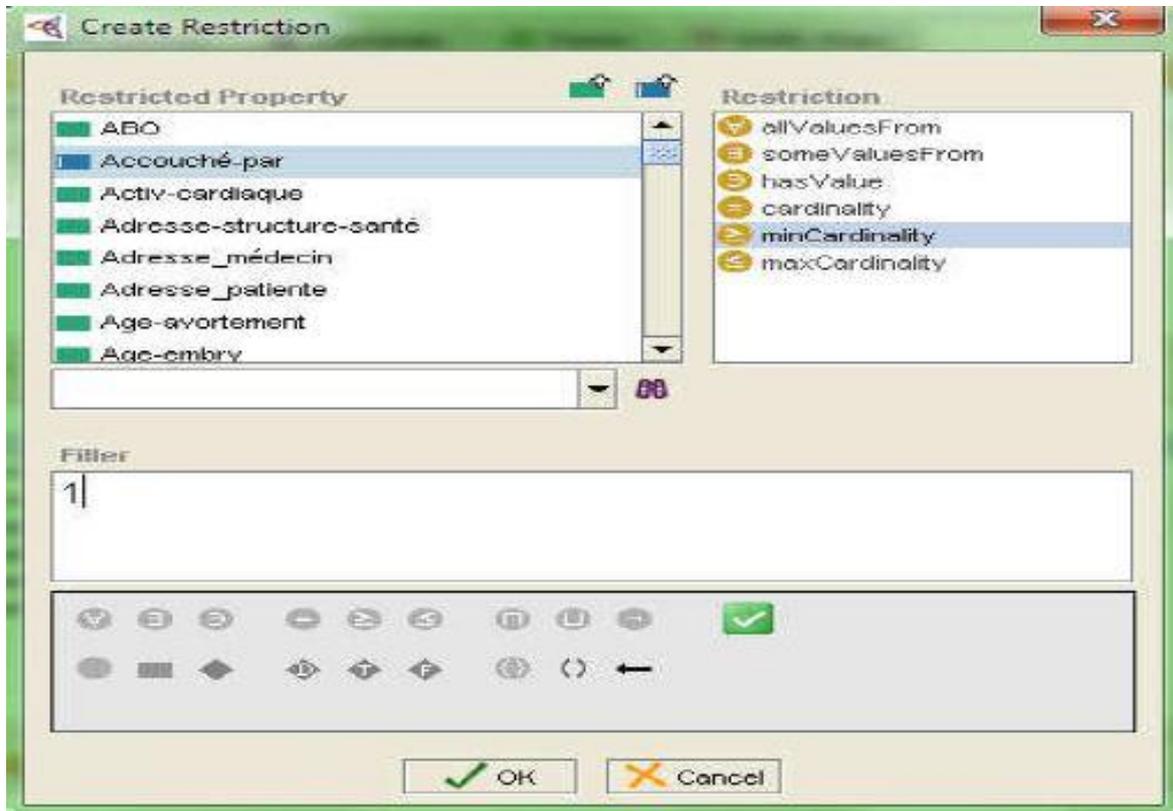


Figure4. 9.Création d'une restriction.

Une fois la fenêtre est affichée, vous sélectionnez la propriété sur laquelle vous voulez faire la restriction, puis choisissez le type de restriction comme Cardinality, minCardinality,maxCardinality, hasValue,...

2.2.6. Création des instances

La dernière étape consiste à créer les instances des classes dans la hiérarchie. Définir une instance individuelle d'une classe exige

- 1) choisir une classe
- 2) créer une instance individuelle de cette classe
- 3) remplir les valeurs d'attributs et les relations.

Par exemple, la figure 4.10 montre la création d'un individu de la classe «Infirmier»

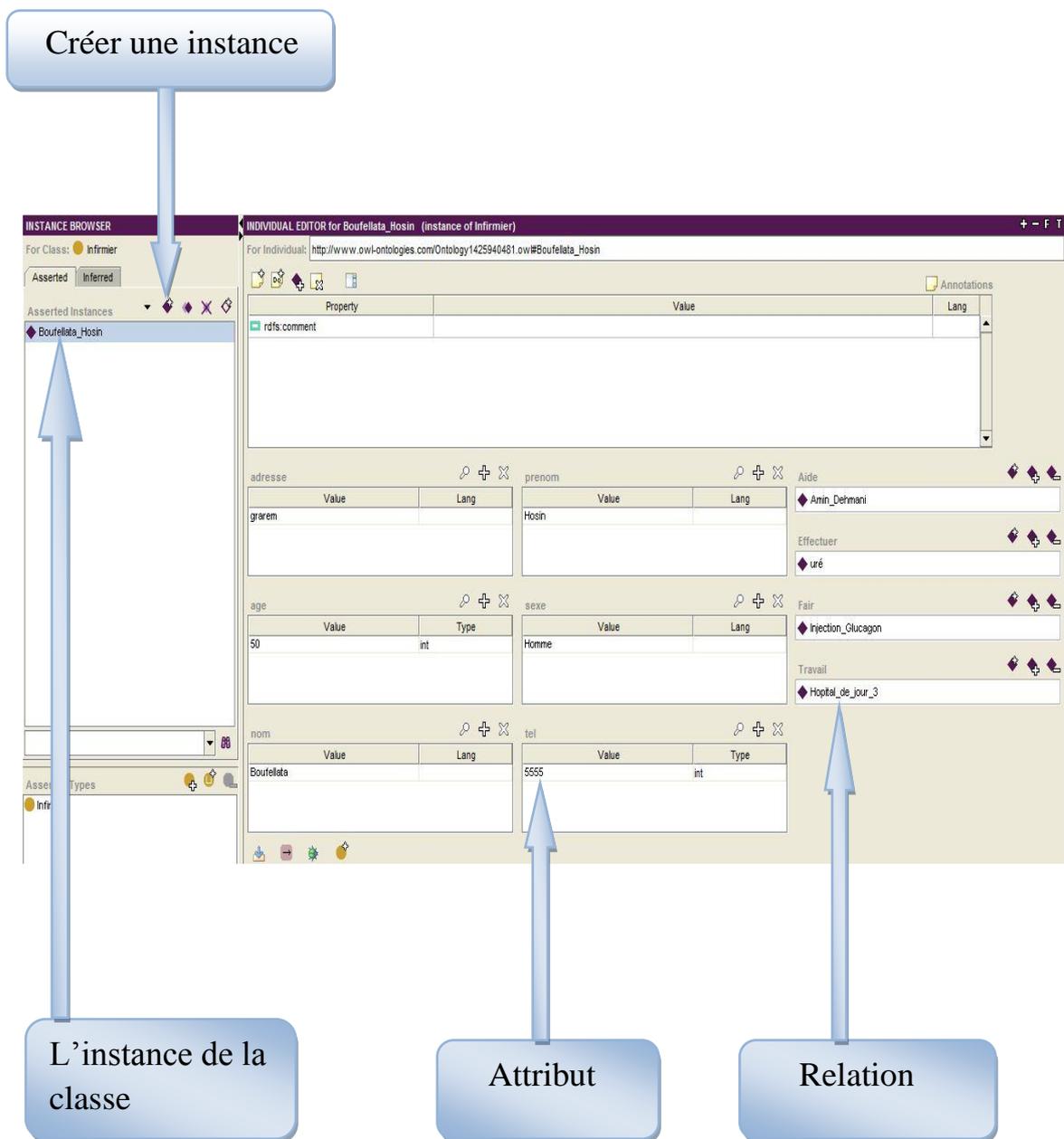
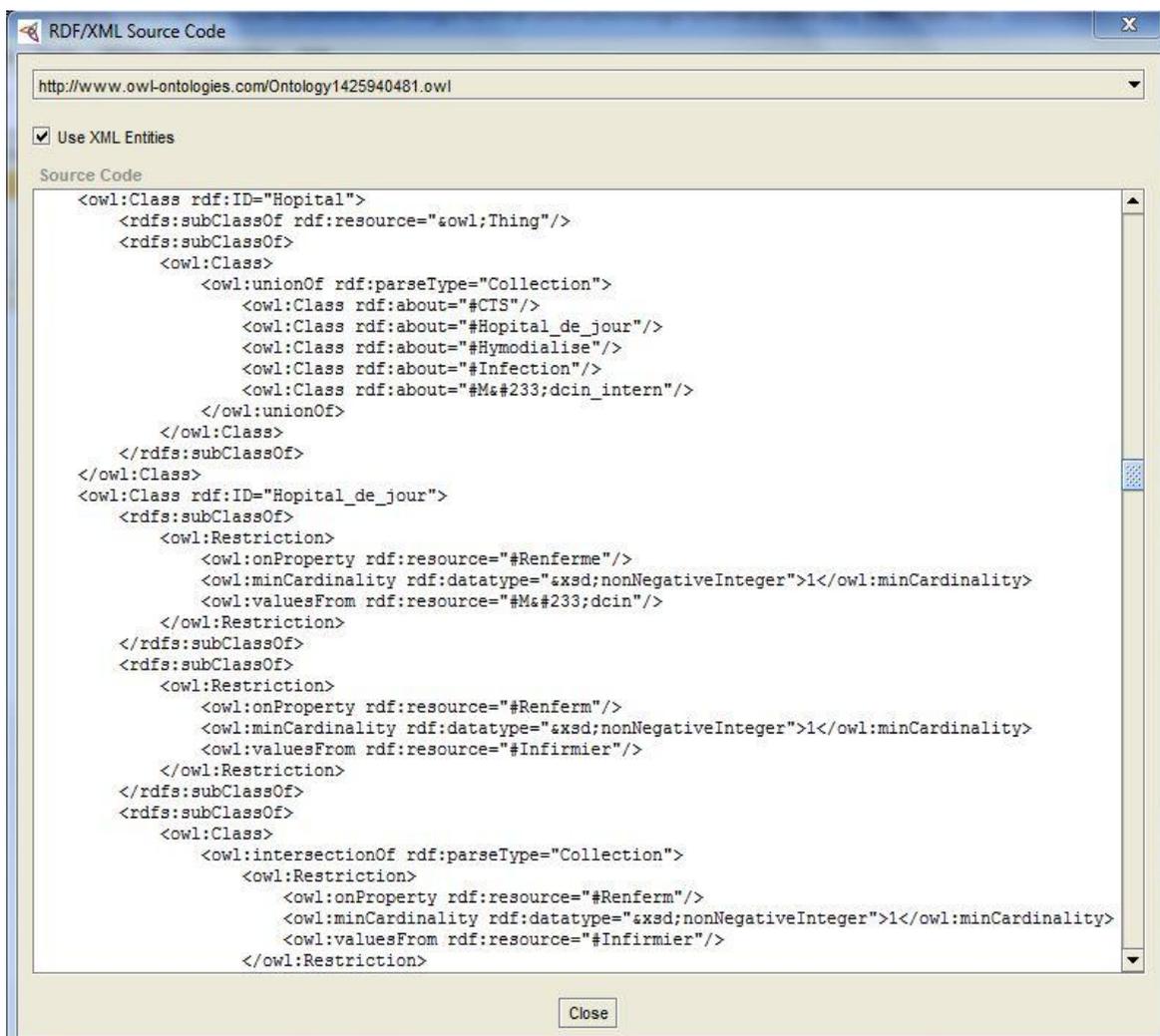


Figure4. 10.Création d'instances.

2.2.7. Génération du code

La construction de l'ontologie est une tâche complexe, même pour implémenter une petite Ontologie, celle-ci va prendre plusieurs lignes de code, et nécessite un grand effort, Pour cela L'outil Protégé OWL a été conçu pour dégager et libérer le développeur de la complexité du codage. Et dans la figure 4.11 on présent un petit code OWL généré par PROTÉGÉ de notre ontologie «diabète ».



```
<owl:Class rdf:ID="Hopital">
  <rdfs:subClassOf rdf:resource="sowl;Thing"/>
  <rdfs:subClassOf>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#CTS"/>
        <owl:Class rdf:about="#Hopital_de_jour"/>
        <owl:Class rdf:about="#Hymodialise"/>
        <owl:Class rdf:about="#Infection"/>
        <owl:Class rdf:about="#M#233;dcin_intern"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Hopital_de_jour">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#Renferme"/>
      <owl:minCardinality rdf:datatype="xsd;nonNegativeInteger">1</owl:minCardinality>
      <owl:valuesFrom rdf:resource="#M#233;dcin"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#Renferm"/>
      <owl:minCardinality rdf:datatype="xsd;nonNegativeInteger">1</owl:minCardinality>
      <owl:valuesFrom rdf:resource="#Infirmier"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Restriction>
          <owl:onProperty rdf:resource="#Renferm"/>
          <owl:minCardinality rdf:datatype="xsd;nonNegativeInteger">1</owl:minCardinality>
          <owl:valuesFrom rdf:resource="#Infirmier"/>
        </owl:Restriction>
      </owl:intersectionOf>
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
```

Figure 4. 11. Fragment du code OWL généré par PROTÉGÉ.

2.2.8. Test & évolution de l'ontologie

Nous avons utilisé le système pellet pour tester l'ontologie d'application, nous distinguons deux types de test: test de consistance et de satisfiabilité, le premier type de test consiste à enlever l'inconsistance entre les concepts et cela en utilisant la test de subsumption, par contre le test de satisfiabilité permet de vérifier pour chaque concept l'existence des instances, un concept C est satisfiable si et seulement si, il existe au moins une interprétation I (instance) pour le concept C.

D'après les tests que nous avons appliqués à l'ontologie d'application, aucune erreur n'est produite lors du test.

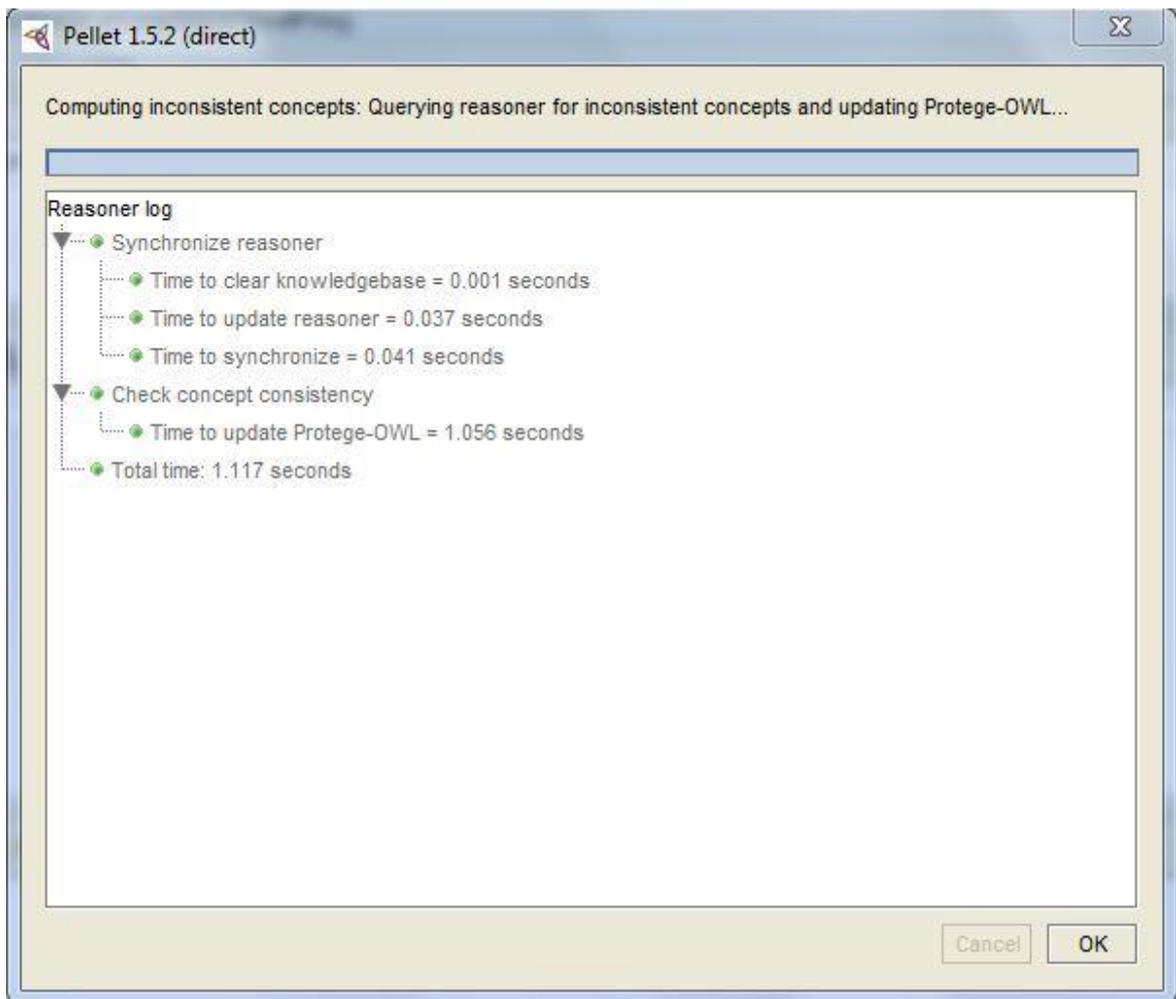


Figure4. 12.Le test de consistance.

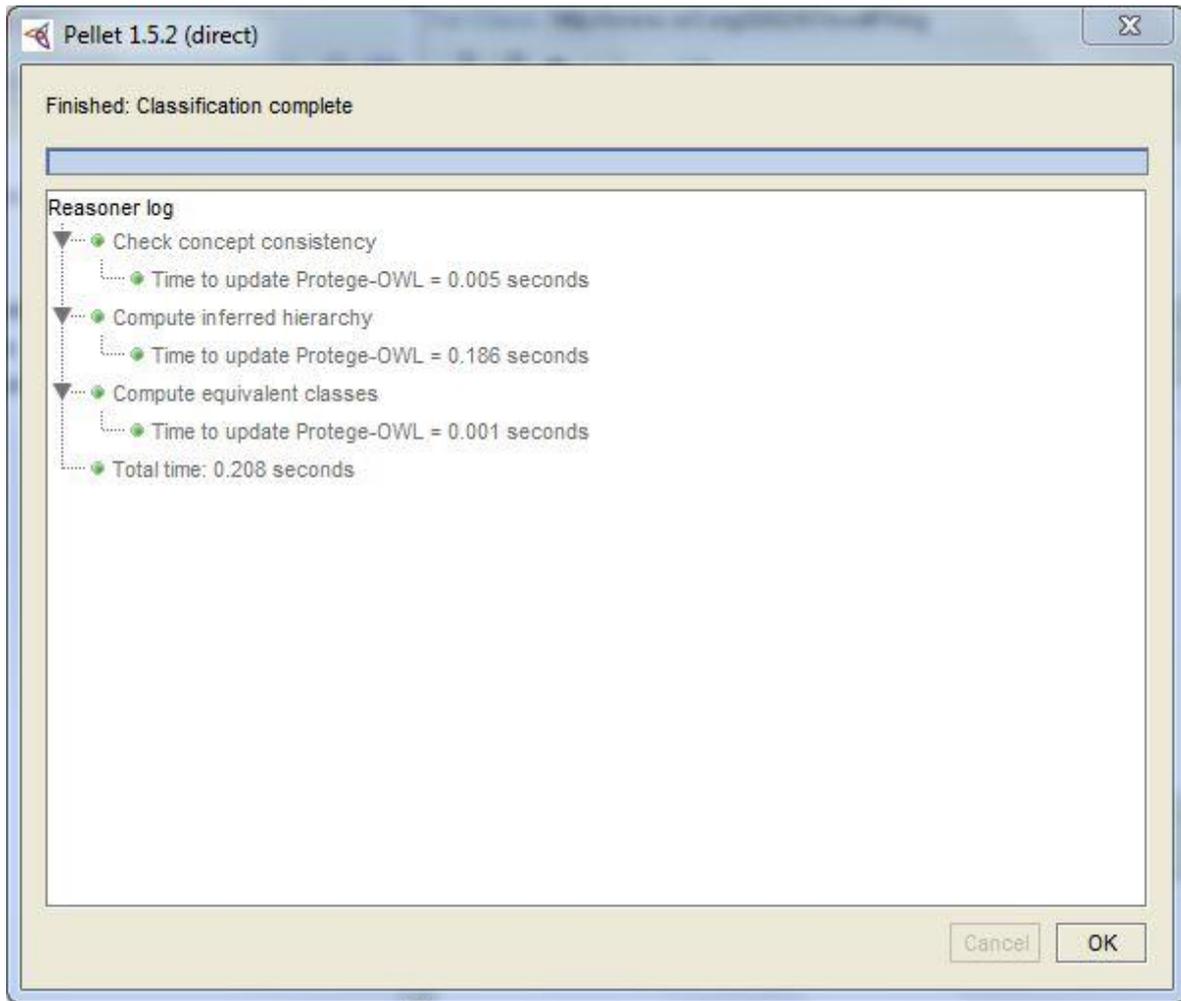


Figure4. 13.Le test de classification.

3. l'implémentation de l'application de gestion de l'ontologie

Il existe plusieurs outils et langages pour la gestion des ontologies médicales. Parmi ces outils nous avons utilisé NetBeans IDE et Jena. Et parmi les langages nous avons utilisés JAVA.

3. 1. Java

Apparu fin 1995 début 1996 et développé par Sun Microsystems Java s'est très rapidement taillé une place importante en particulier dans le domaine de constructions des applications. Les objectifs de java sont d'être multi-plateformes et d'assurer la sécurité aussi bien pendant le développement que pendant l'utilisation d'un programme java.

Java est un langage orienté objet, il permet de construire des applications (bureautiques, graphiques, multimédias, bases de données, environnement de développement, etc...).

Chapitre 4 Implémentation

Son point fort est la portabilité à ses bibliothèques de classes indépendantes de la plateforme, ce qui est le point essentiel de la programmation sur internet ou plusieurs machines différents sont interconnectées.

La réalisation multi-plateformes dépend en fait du système d'exploitation et de sa capacité à posséder des outils de compilation et d'interprétation de la machine virtuelle Java.

3.2. Jena

Jena est une API (application programming interface) Java open source permettant de construire des applications de Web sémantique. Elle contient des classes et des interfaces pour l'interaction avec les modèles RDF et OWL.

Le développement des applications basées sur les ontologies est une tâche très lourde et qui nécessite, préalablement, des infrastructures ou bien des interfaces de programmation déjà construites et qui sont prêtes à l'emploi tel que Jena. De ce fait, nous allons opter la plateforme Jena pour développer notre applications pour visualiser les informations de l'ontologie.

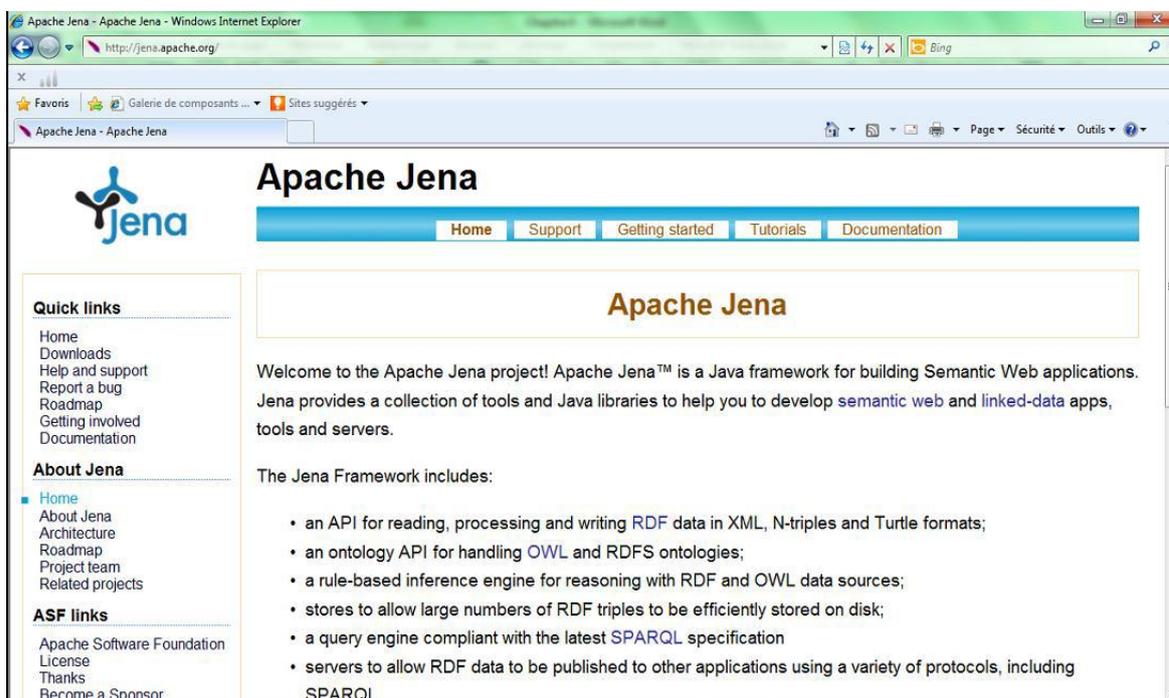


Figure4. 14. Page d'accueil JENA.

3.3. NetBeans

NetBeans IDE (integrated development environment) est un environnement de développement intégré (On peut y écrire, compiler et exécuter les programmes. Un IDE fournit aussi un efficace d'Aide qui décrit tous les éléments du langage et permet facilement de trouver les erreurs dans tes programmes.)

NetBeans n'est pas uniquement un EDI Java. C'est également une plateforme, vous permettant d'écrire vos propres applications Swing.

La licence de NetBeans permet de l'utiliser gratuitement à des fins commerciales ou non, Elle permet de développer tous types d'applications basées sur la plateforme NetBeans.

Aussi nous pouvons facilement ajouter de la bibliothèque pour construire des applications, dans notre cas en ajoute la bibliothèque Jena.

La figure suivante montre notre projet avec la bibliothèque Jena.

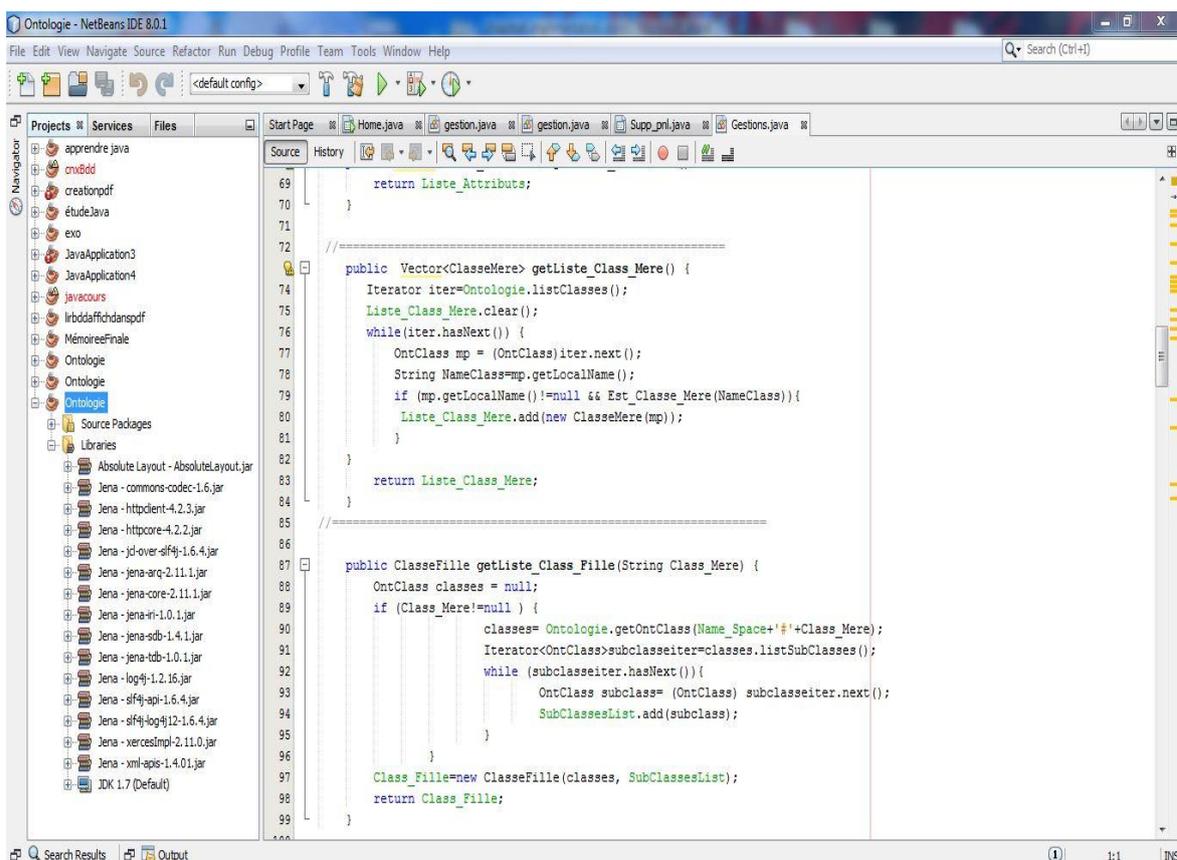


Figure4. 15.Le projet Ontologie avec la bibliothèque JENA.

3.4. Interfaces

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

L'écran ci-dessous présente l'interface principale de notre application.

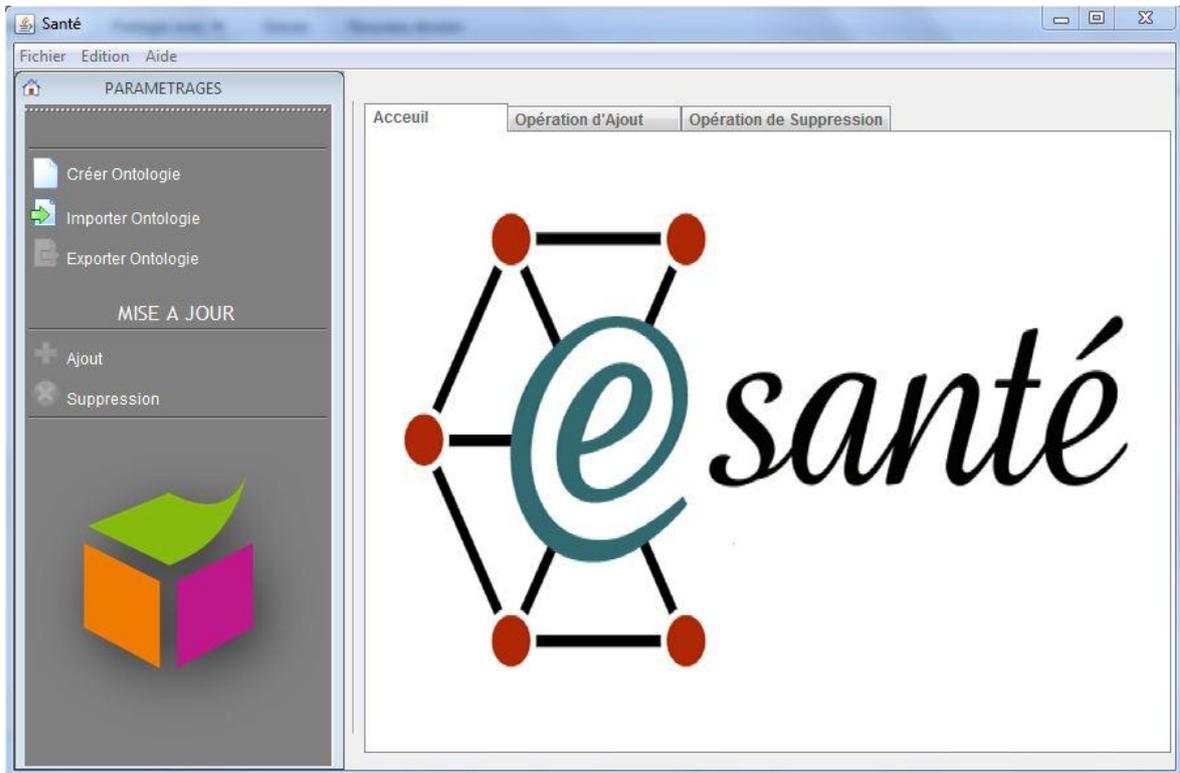


Figure4. 16.Copie d'écran de l'interface principale.

Chapitre 4 Implémentation

L'écran suivant (Figure 4.17) permet d'importer notre ontologie.

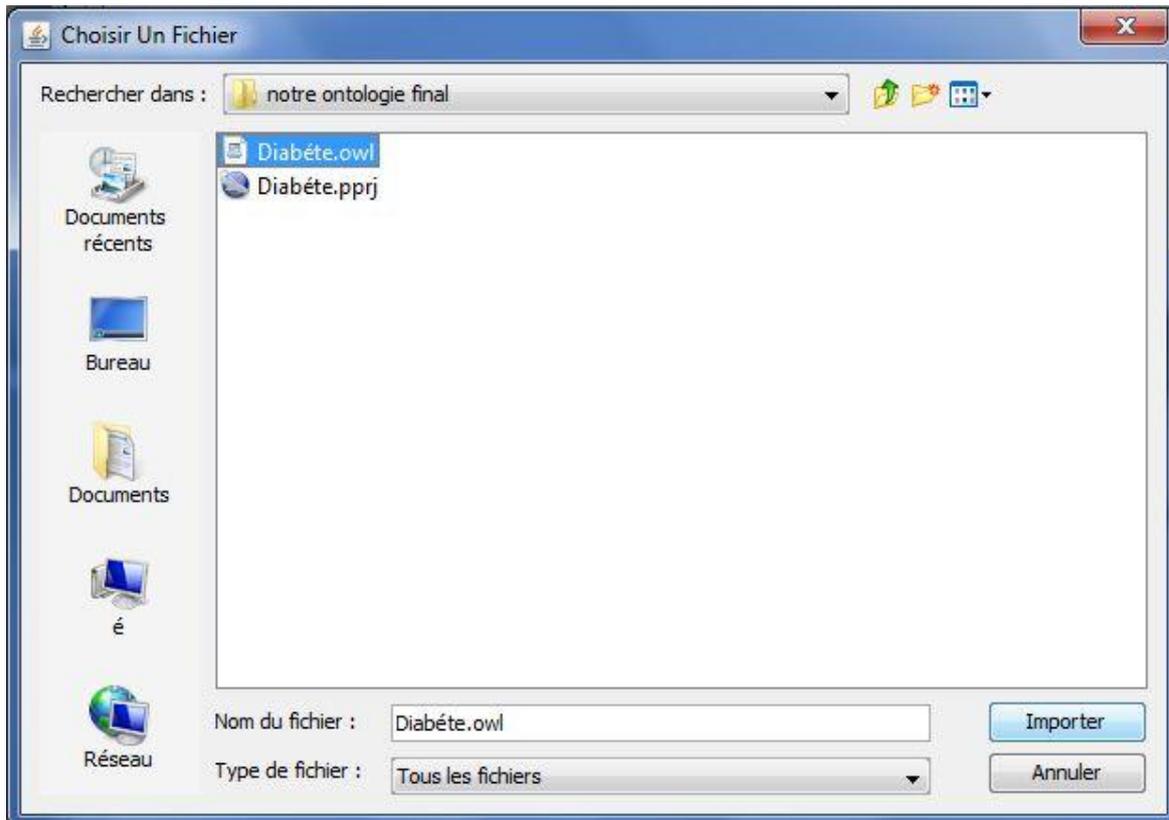
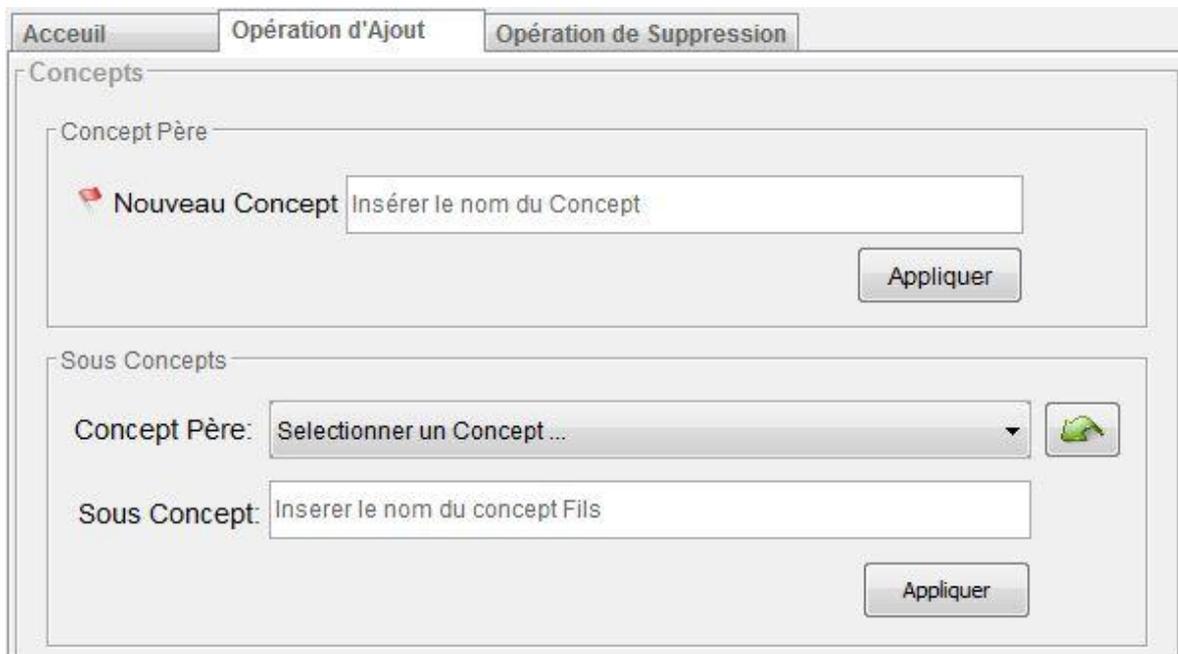


Figure4. 17.Copie d'écran de l'importation de l'ontologie.

La figure (Figure4. 18) représente l'ajout d'un concept et sous concept dans l'ontologie.



The screenshot shows a software interface with three tabs: 'Accueil', 'Opération d'Ajout', and 'Opération de Suppression'. The 'Opération d'Ajout' tab is active. Under the 'Concepts' section, there are two main areas. The first is 'Concept Père', which contains a red flag icon, the text 'Nouveau Concept', a text input field with the placeholder 'Insérer le nom du Concept', and an 'Appliquer' button. The second area is 'Sous Concepts', which contains a 'Concept Père' dropdown menu with the text 'Selectionner un Concept ...' and a green arrow icon, a 'Sous Concept' text input field with the placeholder 'Insérer le nom du concept Fils', and another 'Appliquer' button.

Figure4.18. Copie d'écran d'ajout d'un concept et sous concept.

La figure (Figure4.19) représente l'ajout d'une relation <<faire>> entre le concept patient et le concept examen clinique dans l'ontologie.



The screenshot shows the same software interface with the 'Opération d'Ajout' tab active. Under the 'Relation' section, there are three main fields. The first is 'Domain', a dropdown menu with 'Patient' selected and a green arrow icon. The second is 'Range', a dropdown menu with 'examane-clinique' selected and a green arrow icon. The third is 'Nom de Relation', a text input field with 'faire' entered. An 'Appliquer' button is located at the bottom right of the section.

Figure4.19. Copie d'écran d'ajout d'une relation entre deux concepts.

La figure (Figure 4.20) représente l'ajout d'individu Belkacemi Romeisa.

The screenshot shows a web application interface with three tabs: 'Accueil', 'Opération d'Ajout', and 'Opération de Suppression'. The 'Opération d'Ajout' tab is active. Below the tabs, the title 'Individu' is displayed. The form contains the following elements:

- A dropdown menu for 'Concept' set to 'Personne'.
- A text input field for 'Nom d'Individu' containing 'Belkacemi Romeisa'.
- A button labeled 'Ajouter Attributs'.
- A table with three columns: 'Attribut', 'Type Attribut', and 'Valeur Attribut'. The table contains the following data:

Attribut	Type Attribut	Valeur Attribut
prenom	string	Belkacemi
nom	string	Romeisa
adresse	string	Cité 160 Mila
age	int	26
tel	int	066666666
sexe	string	Femme

At the bottom right of the form is an 'Appliquer' button.

Figure4.19.Copie d'écran d'ajout d'un individu.

La figure (Figure 4.21) représente l'ajout d'un attribut <<Nom>> dans le Concept complication.

The screenshot shows a web application interface with three tabs: 'Accueil', 'Opération d'Ajout', and 'Opération de Suppression'. The 'Opération d'Ajout' tab is active. Below the tabs, the title 'Attribut' is displayed. The form contains the following elements:

- A dropdown menu for 'Concept' set to 'Complication'.
- A text input field for 'Attribut' containing 'nom'.
- A dropdown menu for 'Types' set to 'string'.
- A green message 'Ajout Avec Succès'.
- An 'Appliquer' button.

Figure4.20.Copie d'écran d'ajout d'un attribut.

La figure (Figure 4.22) représente la suppression du concept psychologue.

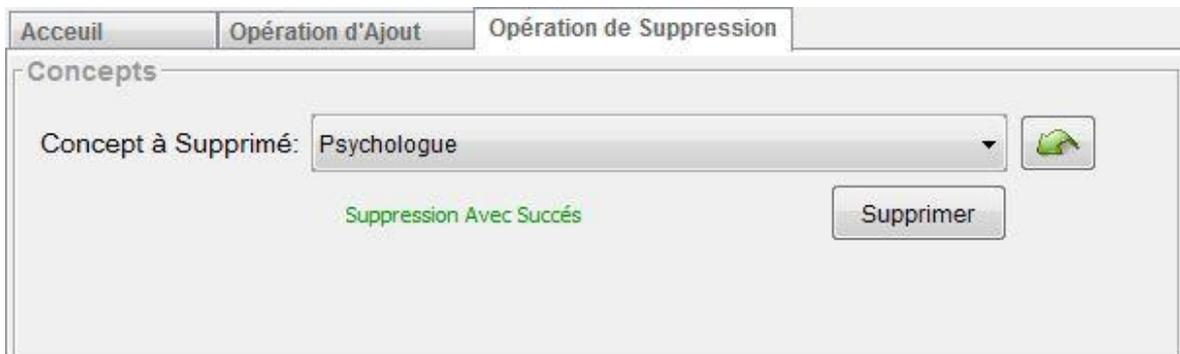


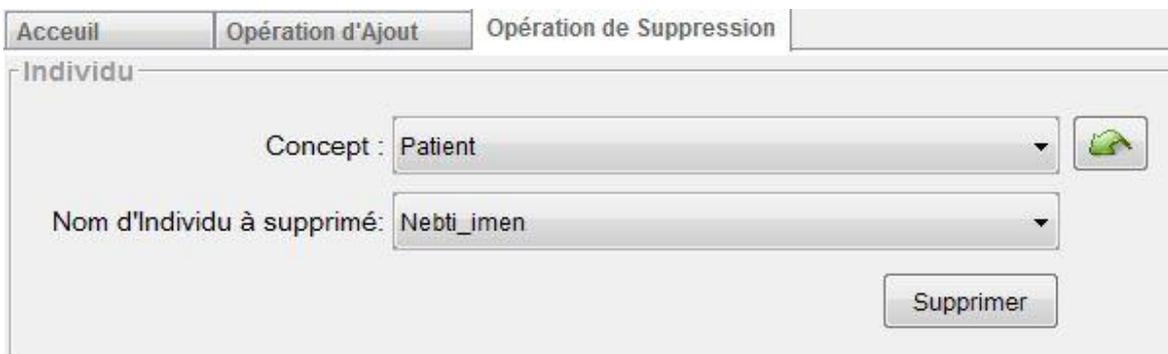
Figure4.21. Copie d'écran suppression du concept.

La figure (Figure4.23) représente la suppression de la relation Réaliser.



Figure4. 23. Copie d'écran suppression d'une relation.

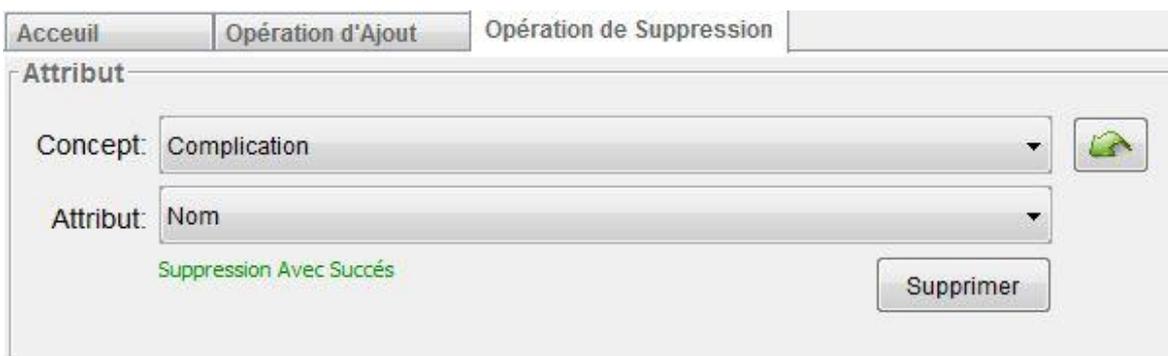
La figure (Figure4. 24) représente la suppression d'un individu Nebti imen.



The screenshot shows a software interface with a tabbed menu at the top containing 'Accueil', 'Opération d'AJout', and 'Opération de Suppression'. The 'Opération de Suppression' tab is active. Below the menu, the title 'Individu' is displayed. The interface contains two dropdown menus: 'Concept' with 'Patient' selected and 'Nom d'Individu à supprimer' with 'Nebti_imen' selected. To the right of the 'Concept' dropdown is a green arrow icon. At the bottom right, there is a 'Supprimer' button.

Figure4. 24. Copie d'écran suppression d'un individu.

La figure (Figure4. 25) représente la suppression l'attribut Nom du concept Complication.



The screenshot shows the same software interface as Figure 4.24, but with the 'Attribut' tab selected. The 'Concept' dropdown menu is set to 'Complication' and the 'Attribut' dropdown menu is set to 'Nom'. A green arrow icon is visible to the right of the 'Concept' dropdown. Below the dropdowns, the text 'Suppression Avec Succès' is displayed in green. A 'Supprimer' button is located at the bottom right.

Figure4. 25. Copie d'écran suppression d'un attribut.

4. Conclusion

Dans ce chapitre, nous avons implémenté l'ontologie médicale «diabète».Par le langage OWL qui facilité beaucoup cette phase d'implémentation, grâce à son pouvoir d'expression, et pour sa décidabilité, aussi l'utilisation de protégé OWL simplifié cette tâche a moyennant son interface graphique sophistiquée. Et finalement, nous avons présenté les outils utilisés dans le développment de notre application « sante » nous utilisons les outils NetBeans IDE et Jena, et en utilisant le langage JAVA, nous avons développé une application de gestion d'ontologie médicale. Les différentes opérations de gestion de l'ontologie ont été développées dans ce chapitre. Nous avons montre comment on peut

Chapitre 4 Implémentation

ajouter ou supprimé un concept, relation, individu ou attribut Après l'importation d'une ontologie.

Conclusion général

Les ontologies médicales servent de plate-forme aux praticiens pour une prise de décision adéquate dans leur diagnostic. La prise en compte des différents aspects liés aux patients, aux maladies, aux remèdes et qui sont très nombreux est facilité par la création de telles ontologies médicales. Elle peut regrouper de nombreuses informations ainsi que les liens entre elles que ce soit du type hiérarchique (classes et sous classes), relations entre les concepts, propriétés des individus, etc....

Après avoir présenté la notion d'ontologie ses composants et son utilisation, les types d'ontologie, leurs processus de construction et leurs différentes méthodologies de construction ainsi le cycle de vie d'ontologie et le formalisme de représentation d'ontologie. Nous avons choisi le type et la méthode pour la construction de notre ontologie diabète dans le domaine médical passe par tous les étapes de création.

Nous avons utilisé le langage de modélisation UML pour développer notre application de gestion de l'ontologie.

Le choix du langage d'implémentation joue un rôle important dans la construction de l'application. En utilisant les outils NetBeans IDE et Jena, et en utilisant le langage JAVA, nous avons développé une application de gestion d'ontologie médicale. Les différentes étapes de la gestion de l'ontologie ont été développées sont l'ajout et la suppression d'un concept, relation, individu et attribut.

References bibliographies

- [1] T.R.Gruber, The Role of Common Ontology in Achieving Sharable, Reusable Knowledge Bases, In J.A.Allen, R.Fikes, and E.Sandewall (Eds), Principles of Knowledge Representation and Reasoning, Proceedings of the Second International Conference,Cambridge, MA, 1991, Morgan Kaufmann, pp. 601-602, 1991.
- [2] Gómez-Pérez, A., Fernández-López, M., and Corcho, O. Ontological Engineering with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web. Springer, 2004.
- [3] N. Aussenac-Gilles & A. Busnel. “Méthode de construction à partir du texte d’une ontologie du domaine de l’industrie de la fibre de verre”. Rapport Interne IRIT/2002-11-R. Avril 2002.
- [4] Neeches, Finin T ,Fikes R.E, Gruber T.R, Senator T et Swartou W.R. « Enabling technology for knowledge sharing » AI Magazine. Vol.12, no 3, 1993.
- [5] Gruber T. (1993).A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition, 5(2), 199-220.
- [6] Guarino N. (1997b).Understanding, building and using ontologies. International J. Human-Computer Studies, 46, 293-310.
- [7] Borst W. N. (1997).Construction of Engineering Ontologies .Center for Telematica and Information Technology, University of Tweenty, Enschede, NL.
- [8] M.Uschold & M.Grüninger, “ONTOLOGIES: Principles, Methods and Applications”. Knowledge Engineering Review . 1996.

- [9] Gómez-Pérez, A., Fernández-López, M., et Corcho, O. (2004). *Ontological Engineering (with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web)*. Springer.
- [10] Mizoguchi, R., Vanwelkenhuysen, J., & Ikeda, M. (1995). Task Ontology for reuse of problem solving knowledge . In N. Mars (Ed.), *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing (KBKS'95)* (pp. 46-57). IOS Press.
- [11] Van Heijst, G., Schreiber, A. T., & Wielinga, B. J. (1997). Using explicit ontologies in KBS development. *International Journal of Human-Computer Studies*, 45, 183–292.
- [12] Guarino, N. (1998). *Formal Ontology in Information Systems* .In N. Guarino (Ed.), *1st International Conference on Formal Ontology in Information Systems (FOIS'98)* (pp. 3-15). IOS Press.
- [13] Gruber, T. R. (1993a). A translation approach to portable ontology specification . *Knowledge Acquisition*, 5 (2), 199–220.
- [15] B. Bachimont, A[14] Hernandez, N., *Ontologies de domaine pour la modélisation du contexte en recherche d'information*. Thèse de doctorat, Université de Toulouse, 2005.
- rts et sciences du numérique : Ingénierie des connaissances et critique de la raison Computationnelle, Mémoire d'Habilitation à Diriger des Recherches, Université de Technologie de Compiègne, 2004.
- [16] M. Uschold, M. King. “Towards a methodology for building ontology, in *Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing*”, IJCAI'95, 1995.
- [17] G. Meditskos, N. Bassiliades, “ Combining a DL Reasoner and a Rule Engine for Improving Entailment-based OWL Reasoning”, *Proc. 7th International Semantic Web Conference (ISWC-2008)*, 26-30 Oct 2008, Karlsruhe, Germany, Springer, 2008 .
- [18] T-R. Gruber. “Towards Principles for the Design of Ontologies Used for Knowledge

Sharing ». International Journal Human-Computer Studies, 1995 .

[19] M.Uschold&M.Grüniger, “ONTOLOGIES: Principles, Methods and Applications”. Knowledge Engineering Review. 1996.

[20] Sowa, J, F, <<Conceptual structures – Information Processing in Mind andMachin>>. The systems Programing Series, Addison-Wesley,(1984)

[21] R,Quillian. <<Semantic Memory>>. In Marvin Minsky, editor Semantic Information Processing.MITPress, 1969.

[22] V. Psyché, O.Mendes and J.Bourdeau, <<Apport de l'ingénierie ontologique aux environnements de formation a distances>>, Centre de recherche LICEF, Télé-université,2003

[23] Valéry Psyché, V. Psyché, <<Rôle des ontologies on ingénierie des EIAH : cas d'un système d'assistance au design pédagogique>>, thèse de doctorat. Université de Québec Montréal, 2007

[24] Abdel Kader Keitak, <<Conception coopérative d'ontologies Prés Consensuelles : Application au domaine de l'Urbanisme>>, thèse de doctorat. Université de LYON, le 06 Juin 2007.

[25] F.First, <<L'ingénierie ontologique>>, Rapport de recherche N°02-07, (2002)

[26] Catarina Eufémia, << Découvert de correspondances sémantiques entre ressources Hétérogènes dans un environnement coopératif>>, thèse de doctorat, Université Claude Bernard- Lyon 1, le 7 décembre 2007.

[27] Anne Tireau, << Une application Smartphone pour un système de recommandation Alimentaires personnalisées>>, mémoire de master2, Université Montpellier II, le 13 septembre 2010.

- [28] F.Frédéric, Travail de diplôme 2007, “Filière informatique de gestion, Web 3.0 : Interrogation intelligente”, 2007.
- [29] S. Bechhofer, I. Horrocks, and C. Goble “ OilEd: a Reason-able Ontology Editor for the Semantic Web”. In Proceedings of KI2001, Joint German/Austrian conference on ArtificialIntelligence, 2001.
- [30] V. Haarslev, R. Moller and M. Wessel, RACER User’s Guide and Reference Manual,version 1.6.Technical report, University of Hamburg, Computer Science Department
2001.
- [31] M. Hemam. Un processus de developpement d’onntologie dans le cadre du web sémantique, Centre Universitaire Larbi Ben M’hidi -Oum El Bouaghi- Institut des sciences exactes (2005).
- [32] Marisol Touraine diabitique.Les différents type de diabète - Doctissimo.htm.
- [33] Laurent AUDIBERT, UML 2, éditions Ayrolles, édition 2007/2008.
- [34] Pascal Roques, UML - Modéliser une application web.
- [35] Raphaël Troncy « Formalisation des connaissances documentaires et des connaissances conceptuelles à l’aide d’ontologies : application à la description de documents audiovisuels »THÈSE pour l’obtention du Doctorat de l’université Joseph Fourier – Gren.

Glossaire

DL	Description Logic
DAML	DARPA Agent Markup Language
DIG	Domain Information Groper
HTTP	HyperText Transfer Protocol
IHM	Interface Homme Machine
OIL	Ontology Inference Layer
OML	Ontology Markup Language
OWL	Ontology Web Langage
RACER	Renamed Abox and Concept Expression Reasoner
TCP	Transmission Control Protocol
W3C	World Wide Web Consortium
XML	Xtensible Markup Langage

Liste des acronymes

ADA	Association Américaine de Diabétologie
API	Application Programming Interface
ANAES	Agence Nationale d'Accréditation et d'Evaluation en Santé
CTS	Centre de Transfert de Sang
DC	Dictionnaire de Concepts
DID	Diabète Insulino Dépendant
DIND	Diabète Non Insulino Dépendant
GL	Génie Logiciel
IA	Intelligence Artificielle
IC	Ingénierie des Connaissances
IDE	Integrated Development Environment
nRQL	New Racer proquery Language
RDF	Resource Description Framework
RDF(S)	Resource Description Framework Schema
RICE	Racer Interactive Client Environment
TOVE	Toronto Virtual Enterprise
UML	Unified Modeling Language
UP	Unified Process
XP	Xtreme Programming