

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



N° Ref :.....

Centre Universitaire de Mila

Institut des sciences et de la technologie

Département de Mathématiques et Informatique

Raisonnement sur une ontologie hybride pour la recherche d'informations médicales

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

Préparé par : Bouhennache Nouredine
Fekraoui Abdennour

Encadré par : Afrai Faiza

Filière : Informatique

Spécialité : STIC

Année universitaire : 2012/2013

REMERCIEMENTS

En tout premier lieu, nous remercions DIEU, le Tout Puissant, qui nous a éclairé le bon chemin et nous a aidé à réaliser ce modeste travail dans les meilleures conditions.

Nous tenons à remercier notre encadreur, Mme F. Afri, pour nous avoir donné l'opportunité de travailler sur ce projet, pour son grand soutien scientifique et moral, pour les conseils, les suggestions et les encouragements qu'elle nous a apportés tout au long de la réalisation de ce mémoire.

Nous adressons également nos remerciements à tous nos enseignants, qui nous ont donné les bases de la science. Nous remercions très sincèrement les membres de jury Khalfi Souheila, Zekiouk Mounira pour nous avoir fait l'honneur d'évaluer notre travail.

DEDICACES

A mes très chers parents qui m'ont beaucoup

Aidé, Soutenu, et encouragé ;

A mes très chères sœurs

A ma petite belle-sœur Roumaïssa

A mon beau-frère Karim ;

A tous mes amis de l'UCM ;

A toute la famille Fekraoui;

A toute la famille Merabet;

A mon binôme et ami Noureddine avec qui j'ai

Partagé ce modeste travail;

A tous ceux que j'aime tant et que je n'ai

Pas cités.

Je dédie ce mémoire ...

Fekraoui Abdennour

DEDICACES

A mes très chers parents qui m'ont beaucoup aidé, soutenu, encouragé et qui ont fait de moi ce que je suis aujourd'hui ;

A mes chères sœurs

Hanane et Lamia ;

A mes beaux frères

Abdelhamid, Abdessalam, SalahEddine ;

A mes beaux neveux

Abdallah et Mohamed El Amine

A mes oncles et tantes, ainsi que mes cousins et cousines ;

A tous mes amis, en particulier

Abdenmour et Zaki ;

A toutes mes amies, en particulier

Besma, Imane, Hayate et Soumia ;

A toutes les familles

Bouhenache et Baouta ;

A toute la famille

Boufendassa en particulier Salim ;

A tous ceux que j'aime tant et que je n'ai pas cités ;

Je dédie ce mémoire ...

Bouhennache Nouredine

Tables des matières

Introduction générale

1. Contexte général.....	1
2. Contribution.....	1
3. Plan du mémoire.....	2

Chapitre 1 Web sémantique et les ontologies

1. Introduction.....	4
2. Le Web Sémantique.....	4
2.1. Architecture et langages du web sémantique.....	5
2.1.1. Architecture du web sémantique.....	5
2.1.2. Langages de représentation de l'ontologie.....	10
3. Ingénierie ontologique.....	13
3.1. Notion d'ontologie :.....	14
3.2. Définitions :.....	14
3.3. Les éléments constituant une ontologie :.....	15
3.4. Utilisation d'une ontologie.....	16
3.5. Les typologies d'ontologies.....	18
3.6. Construction d'une ontologie.....	19
3.6.1. Un squelette de méthodologie pour construire des ontologies.....	19
3.6.2. Quelques méthodologies de construction d'ontologies.....	22
3.6.2.1 Tove.....	22
3.6.2.2 Enterprise.....	22
3.6.2.3. Methontology.....	23
4. Outils pour le Web Sémantique.....	24
4.1. Editeurs d'ontologie.....	25
4.1.1 Protégé.....	25
4.1.2. OILED.....	25
4.1.3 WebODE (2003).....	25

4.2. Moteurs d'inférences.....	26
4.2.1 Racer	27
4.2.2. Pellet	27
5. Conclusion.....	28

Chapitre 2 Raisonnement sur les ontologies hybrides

1. Introduction	29
2. Les paradigmes de modélisation.....	29
2.1 Paradigme classique	29
2.2la programmation logique.....	30
3. Les tendances actuelles des langages du Web sémantique	31
4. Intérêt de combinaison des règles et ontologies	31
5. Approches de combinaison	33
5.1. L'approche hybride	33
5.2. Approche homogène	34
5.2.1. Les règles : SWRL.....	34
6. Travaux Connexes	37
6.1. Les approches homogènes réduisant les deux composantes en un langage logique.....	38
6.1.1 SWRL dans Jess	38
6.1.2. KAON2	39
6.1.3 SWRL dans Sesame	39
6.2. Les approches hybrides utilisant des interfaces	40
6.3. Discussion.....	40
7. Conclusion.....	40

Chapitre 3 Conception d'une ontologie hybride dans le domaine médical

1. Introduction	42
2. Objectif du travail	42

3. Architecture du système de raisonnement sémantique	43
3.1. Description de l'architecture.....	43
3.2.Fonctionnement du système de raisonnement	44
4. Processus construction d'une ontologie OWL	45
4.1. Spécification des besoins.....	46
4.2. Conceptualisation.....	46
4.3. Formalisation	47
4.4. Implémentation	48
4.5. Evolution et vérification	49
5. Construction de l'ontologie hybride GynécOntologie dans le domaine médical.....	49
5.1. La gynécologie : domaine d'application	49
5.2. Etape de spécification.....	52
5.3. Etape de conceptualisation	52
5.4. Etape de formalisation.....	76
5.4.1. Construction de TBox	76
5.4.2. Construction de la ABox.....	79
5.5. Représentation sémantique des règles avec le langage SWRL	81
6. Conclusion.....	90

Chapitre 4 Implémentation et mise en Œuvre « GynécOntologie »

1. Introduction.....	92
2. Les outils d'implémentation	92
2.1 Plateforme Protégé.....	92
2.2. Plugins utilisés	93
2.2.1. SWRL Tab.....	93
2.2.2. SWRL Jess Tab	94
2.3. Le classifieur Pellet	95
2.4. Le Moteur Jess	95
3. Création de l'ontologie avec Protégé OWL	96
3.1. Création des classes et hiérarchie de classes	96
3.2. Création des propriétés.....	97

3.2.1. Création d'attributs (DataTypeProperty)	98
3.2.2. Création des relations (ObjectProperty).....	99
3.2.3. Restrictions sur les propriétés.....	100
3.3. Création des instances	102
3.4. Génération du code	102
3.5. Test & évolution de l'ontologie	103
4. La représentation sémantique des règles SWRL	104
4.1. Les règles SWRL	104
4.2. Les built-ins	105
4.3. Le langage de requête SQWRL	106
4.4. Exécution des règles.....	107
4.5. Assertion des nouveaux faits	108
5. l'interface Gynomed	110
5.1. Gynomed	110
5.2. Outils pour construire l'interface « Gynomed »	110
5.2.1. Java	110
5.2.2. Jena	111
5.2.3. NetBeans	111
6. Interrogation de la base de connaissance	112
7. Conclusion.....	115

Conclusion générale et perspectives

1. Conclusion générale.....	116
2. Perspectives	117
Bibliographies.....	118
Glossaire.....	122

Liste des figures

Figure 1.1 L'architecture en couche de la vision du Web sémantique.....	6
Figure 1.2 La pyramide des langages basés Web.....	11
Figure 1.3 Processus de construction d'ontologie.....	20
Figure 1.4 Architecture abstraite d'un raisonneur OWL.....	26
Figure 2.1 Intérêt de combinaison règles et ontologie.....	32
Figure 2.2 Les degrés d'intégrations	33
Figure 2.3 Principe d'exécution d'un moteur de règles	36
Figure 2.4 Modèle de SWRL dans Jess	38
Figure 3.1 Architecture du système de raisonnement	44
Figure 3.2 Fonctionnement du système de raisonnement	45
Figure 3.3 Démarche d'une consultation médicale.	51
Figure 3.4 Hiérarchies de concepts.	57
Figure 3.5 Classification des concepts.....	58
Figure 3.6 Le diagramme des relations binaires.	60
Figure 4.1 L'interface principale de Protégé.....	93
Figure 4.2 L'éditeur de règles SWRL.....	94
Figure 4.3 Le plugin SWRL Jess Tab.....	94
Figure 4.4 Le classifieur Pellet.....	95
Figure 4.5 Installation de Jess.	95
Figure 4.6 Création de classes.....	96
Figure 4.7 Hiérarchies des concepts	97
Figure 4.8 Création de propriétés pour une classe.	98
Figure 4.9 Création d'un attribut.....	99
Figure 4.10 Création d'une relation entre deux classes.....	100
Figure 4.11 Les restrictions créées sur les propriétés d'une classe.	101
Figure 4.12 Création d'une restriction.....	101
Figure 4.13 Création d'instances.....	102
Figure 4.14 Fragment du code OWL généré par PROTÉGÉ	103
Figure 4.15 Le test de consistance.....	104
Figure 4.16 Le test de classification	104

Figure 4.17 Edition des règles SWRL	104
Figure 4.18 Présentation du panneau des icones.....	105
Figure 4.19 Editer une règle SWRL contenant une buit-ins	106
Figure 4.20 SQWRL Query Tab	107
Figure 4.21 Éditer une requête SQWRL.....	107
Figure 4.22 Exécution des règles.....	108
Figure 4.23 Une instance de la classe Patiente avant l'exécution de la règle.	109
Figure 4.24 Une instance de la classe Patiente après l'exécution de la règle.	110
Figure 4.25 Le projet Ontologie avec la bibliothèque Jena.	112
Figure 4.26 Interface authentification.....	112
Figure 4.27 Interface de l'application.....	113
Figure 4.28 Boîte de dialogue pour ouvrir fichier OWL.	113
Figure 4.29 Affiche les informations.....	114
Figure 4.30 Affiche la hiérarchie d'ontologie.....	115

Liste des tableaux

Tableau 1.1 Syntaxe et sémantique des descriptions de concepts	10
Tableau 2.1 paradigme classique et programmation logique	30
Tableau 3.1 Syntaxe du langage SHIQ	48
Tableau 3.2 Glossaire des termes	55
Tableau 3.3 Dictionnaire des concepts	64
Tableau 3.4 Table des relations binaires	67
Tableau 3.5 Table des attributs	69
Tableau 3.6 Table des axiomes logiques	72
Tableau 3.7 Table des instances	76
Tableau 3.8 Définition des concepts et subsomption	78
Tableau 3.9 Les définitions des différents rôles	79
Tableau 3.10 Description des assertions de concepts.....	80
Tableau 3.11 Description des assertions de relations	81
Tableau 3.12 Règles SWRL pour la sélection des examens.....	83
Tableau 3.13 Règles SWRL pour le diagnostic des maladies.	85
Tableau 3.14 Règles SWRL pour la sélection des traitements.	87
Tableau 3.15 Règles SWRL pour le choix de l'accouchement.....	88
Tableau 3.16 Règles SWRL pour interroger notreontologie.....	90

Liste des acronymes

API	Application Programming Interface
DL	Description Logic
DLP	Description Logic Programs
DAML	DARPA Agent Markup Language
JDBC	Java Data Base Connectivity
JSP	Java ServerPage
HTML	HyperText Markup Language
OIL	Ontology Inference Layer
OML	Ontology Markup Language
OWL	Ontology Web Langage
RACER	Renamed Abox and Concept Expression Reasoner
RDF	Resource Description Framework
RDF(S)	Resource Description Framework Schema
RuleML	Rule Markup Language
SHOE	Simple HTML Ontology Extension
SWRL	Semantic Web Rule Language
TOVE	Toronto Virtual Enterprise
URI	Uniform Resource Identifiers
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
XSLT	Extended Style Sheet Langage Transformation
XML	eXtensible Markup Langage
XMLS	eXtensible Markup Langage Schema
XOL	Ontology Exchange Language

Résumé

Les moteurs de recherche à base d'ontologies utilisent un raisonnement sémantique, afin d'apporter à l'utilisateur les informations pertinentes dont il a besoin. En général, les ontologies utilisées dans la phase de raisonnement de ces moteurs sont représentées en OWL, mais vu les insuffisances de ce langage, l'ontologie est étendue par des règles SWRL qui permettent d'augmenter l'expressivité d'OWL. Ces règles capturent les dépendances entre les propriétés définies dans l'ontologie. Les moteurs d'inférence sur ces règles ne sont pas intégrés dans les classifieurs OWL. Le problème qui se pose alors, est l'absence de raisonneurs effectifs capables de faire interopérer les ontologies OWL et les règles SWRL.

Dans ce travail, nous proposons une approche pour combiner le raisonnement sur le langage OWL et celui sur les règles SWRL dans notre ontologie médicale, en nous basant sur les standards et les outils actuels du web sémantique : protégé OWL et ses plugins, Pellet pour raisonner sur les fragments OWL, et Jess pour raisonner sur les règles SWRL, dans le but d'augmenter les performances du système de recherche développé.

Mots clés: recherche sémantique, raisonnement hybride, ontologie, OWL, règles SWRL.

Introduction générale

1. Contexte général

Les ontologies ont un rôle très important à jouer pour la réalisation du Web sémantique. Elles sont utilisées pour fournir le vocabulaire et la structure des métadonnées associées aux sources annotées. La nouvelle vision du web sémantique accorde une grande importance à la couche règles, car leur puissance va enrichir le raisonnement qui est souvent incomplet.

Différents langages de représentation des connaissances sont apparus : ceux basés sur le paradigme classique, et ceux sur la programmation logique. Avec le temps, la communauté des chercheurs ont tenté de combiner ces deux familles offrant différents niveaux d'expressivité.

Ces dernières années, les travaux sur les ontologies ont connu une certaine modération, dans l'architecture en couches du web sémantique, l'intérêt actuel se concentre sur la couche de règles et son intégration avec la couche d'ontologie en vue d'enrichir le raisonnement souvent insuffisant en utilisant OWL seul [3] [4].

L'intégration d'un langage de règles avec un langage d'ontologies exige la définition d'un nouveau langage, sa syntaxe et sa sémantique, et le développement des algorithmes de raisonnement pour le nouveau langage [38]. SWRL(Semantic Web RuleLanguage) [39] est un langage de règles pour le web sémantique, combinant d'une façon homogène le langage OWL-DL (Ontology Web Language) et le langage Rule-ML (RuleMarkupLanguage),et assure l'expression des entités qu'OWL seul ne permet pas d'exprimer.

Pour le raisonnement sur les connaissances, il existe de nombreux moteurs d'inférence, Racer et Pellet par exemple, raisonnent sur les logiques de descriptions et acceptent en entrée les fichiers OWL. Les moteurs d'inférence tels que Jess et Bossam raisonnent quant à eux sur les règles SWRL.

2. Contribution

La composition des modes de raisonnement (mode de raisonnement sur les règles SWRL avec le raisonnement sur les logiques de descriptions) permet d'augmenter les performances

des systèmes de recherche, en donnant une représentation plus riche des connaissances contenues dans l'ontologie. C'est dans ce cadre que s'inscrit notre réflexion.

La grande quantité des connaissances que les médecins sont tenus de posséder, et l'évolution rapide des informations de cette discipline et le besoin du partage, et la réutilisation, ont conduisaient au développement des ontologies médicales.

Dans notre travail nous proposons une base de connaissances médicale étendue avec les connaissances du service de gynécologie, implémentée avec les technologies du web sémantique. Ces dernières offrent non seulement une bonne structuration, organisation, et modélisation des connaissances, mais aussi des supports de raisonnement fiables sur ces connaissances. Cela nous a conduits à la conception d'un système qui représente toutes les connaissances relatives au domaine, et aide à la décision, quand il s'agit de problèmes complexes.

Nous nous basons sur les standards et les outils actuels du web sémantique : OWL, SWRL et sur les outils actuellement disponibles: Protégé OWL pour l'édition notre ontologie et les règles SWRL, Pellet, Jess pour le raisonnement. Pour augmenter les performances du système de recherche développé. Il s'agit de la performance en termes de qualité des résultats.

3. Plan du mémoire

Pour présenter le travail, nous avons élaboré le plan de lecture suivant:

Chapitre 1 : Le Web Sémantique et les ontologies

Ce chapitre introduit le Web sémantique et spécialement les ontologies, nous commençons par un aperçu sur la notion du web sémantique, puis nous présentons ensuite les principaux formalismes de représentation de connaissances, à savoir, les frames, les graphes conceptuels et les logiques de descriptions. Ensuite nous parlons de la notion d'ontologie en ingénierie des connaissances. Nous découvrirons après les méthodologies les plus représentatives de construction d'ontologies et quelques domaines de leur utilisation. A la fin, nous présenterons les outils nécessaires de leur développement, à savoir, les langages de représentation, les outils d'éditeurs, etc.

Chapitre 2 :Raisonnement sur les ontologies hybrides

Dans ce chapitre nous avons montré les paradigmes de modélisations qui supportent des services de raisonnement spécifiques. Puis nous avons exposé la notion d'hybridation, qui est jugée la meilleure solution pour surmonter les insuffisances du langage OWL. La

combinaison du paradigme classique et la logique de programmation sera abordé. Nous terminons ce chapitre avec une présentation des différents supports de raisonnements pour les ontologies OWL enrichies par des règles SWRL.

Chapitre 3 : Conception d'une ontologie hybride dans le domaine médical

Dans ce chapitre, nous allons présenter l'objectif de notre travail, tout au début nous allons décrire l'architecture globale du système, puis nous présentons le fonctionnement du système de raisonnement sur les ontologies hybrides, basé sur l'idée d'avoir deux moteurs d'inférence pour chaque langage. Un raisonneur pour l'ontologie OWL et un moteur de règles pour les règles SWRL. Ces deux moteurs s'exécuteront en parallèles, et échangeront leurs conclusions.

Par la suite nous allons construire notre ontologie GynécOntologie, en suivant les étapes de la méthodologie «Methondology ». La logique de description est adaptée pour la formalisation de cette dernière. Et l'enrichissement de celle-ci par les règles afin d'augmenter son expressivité.

Chapitre 4 : Implémentation et mise en Œuvre « GynécOntologie »

Dans ce chapitre, nous mettrons en œuvre notre ontologie hybride GynécOntologie. Nous allons implémenter les résultats obtenus précédemment, en utilisant le langage OWL qui est considéré comme un standard pour les ontologies, et le langage de règles SWRL reconnu pour son expressivité de représentation des connaissances. Pour assurer le raisonnement hybride, il est préférable d'utiliser des outils standards du web sémantique : protégé OWL, les plugins SWRL Tab, SWRL Jess Tab, le classifieur Pellet pour raisonner sur l'ontologie OWL, ainsi que le moteur de règles Jess pour raisonner sur les règles et inférer de nouvelles connaissances. Puis nous présentons les outils pour construire notre application, nous choisissons Java comme langage de programmation, et nous utilisons aussi la plateforme Jena pour faciliter notre application, et NetBeans comme un environnement de développement.

Chapitre 1



Web sémantique et les ontologies

1. Introduction

La vision du web sémantique a pour objectif d'échanger, partager, et réutiliser des informations, qui seront interprétable par les machines. Ces dernières années, la communauté de recherche du web sémantique a connu des progrès significatifs vers la réalisation de la vision architecturale de ce Web. Les progrès ont été sous forme de normalisation des langages et des technologies Web qui permettent cette vision.

Dans la section 2 de ce chapitre, nous introduisons des généralités sur le Web sémantique et parlons sur son architecture et ces couches ainsi que les langages de représentation des ontologies.

Nous parlerons dans la section 3 sur les ontologies, qui sont devenues la première solution pour modéliser les connaissances d'un domaine donné, aussi nous exposerons quelques méthodes de construction d'ontologie.

Dans la section 4, nous allons présenter les déferents outils du web sémantique, et le mécanisme de raisonnement sur les ontologies.

2. Le Web Sémantique

A sa création par Tim Berners Lee, au début des années 1990, le web était exclusivement destiné à partager des informations sous forme de pages HTML, affichables par un logiciel «navigateur web », et généralement destinées à être lues par un utilisateur humain.

Très rapidement, on s'est rendu compte que cette conception du web était bien trop limitée, et ne permettait pas un réel partage du savoir. Les limites du Web actuel sont nombreuses, nous mentionnons:

- L'hétérogénéité des formats d'information (Word, pdf ...).
- la nature dynamique et évolutive de l'information.
- L'absence de sémantique ou structure de l'information elle-même, etc.

Une des solutions existantes correspondà la vision du Tim Berners-Lee qui a défini le Web sémantique en [5]comme suit :"**Le Web sémantique est une extension du Web actuel (prolongation du Web actuel), dans laquelle l'information reçoit une signification bien définie, améliorant les possibilités de travail collaboratif entre les ordinateurs et les personnes.**"

Le Web sémantique[5] désigne un ensemble de technologies visant à rendre le contenu des ressources du World Wide Web accessible et utilisable par les programmes et agents logiciels grâce à un système de métadonnées formelles utilisant notamment la famille de langages développés par le W3C.

Il s'agit d'arriver à un Web intelligent, où les informations ne seraient plus stockées mais comprises par les ordinateurs afin d'apporter à l'utilisateur ce qu'il cherche vraiment.

D'après la définition de Tim Berners-Lee, le Web sémantique permettra (contrairement au Web actuel vu comme un Web syntaxique) de rendre le contenu sémantique des ressources Web interprétables, non seulement par l'homme mais aussi par la machine.

Le web sémantique se veut un web de demain dont le contenu peut être appréhendé et exploité par des machines. Résumons les objectifs du web sémantique dans les points suivants:

- ✓ Créer un espace d'échange de ressource entre machines permettant l'exploitation de grand volume d'informations et de services variés.
- ✓ Libérer l'utilisateur d'une bonne partie de son travail de recherche et d'exploitation des résultats.
- ✓ Rendre explicite les relations sémantiques qui existent entre unités d'information.
- ✓ Obtenir des annotations sémantiques plus riches, compatibles avec les standards du Web, tel que le XML, qui fournit un format universel de stockage et d'échange.

2.1. Architecture et langages du web sémantique

2.1.1. Architecture du web sémantique

La vision du Web sémantique est présentée dans la figure 1.1. L'architecture de cette vision est organisée en couches dont seulement les couches basses sont aujourd'hui relativement stabilisées. Chaque niveau repose sur les résultats définis au niveau inférieur, c'est-à-dire que chaque niveau est progressivement plus spécialisé et plus complexe que le niveau précédent.

Les travaux visant la réalisation du web sémantique se situent à des niveaux de complexité très différents. Les plus simples utilisent des jeux plus ou moins réduits de métadonnées dans un contexte de recherche d'information ou pour adapter la présentation des informations aux utilisateurs. Dans ce cas, des langages de représentations simples sont suffisants. Dans les travaux plus complexes mettant en œuvre des architectures sophistiquées, pour permettre par

exemple l'exploitation de ressources hétérogènes, des langages plus expressifs et plus formels issus des travaux de représentations et en ingénierie des connaissances, sont nécessaires.

Dernièrement, le Langage SWRL pour le niveau logique a été soumis au W3C pour devenir une recommandation au même titre que les autres langages.

Grâce aux efforts établis pour réaliser la vision du Web sémantique et au grand consensus dans la communauté d'inclure les règles avec l'ontologie dans l'architecture du Web sémantique, un diagramme de couche plus raffiné a résulté [1], ce diagramme est montré dans la partie(b) de la figure1.1

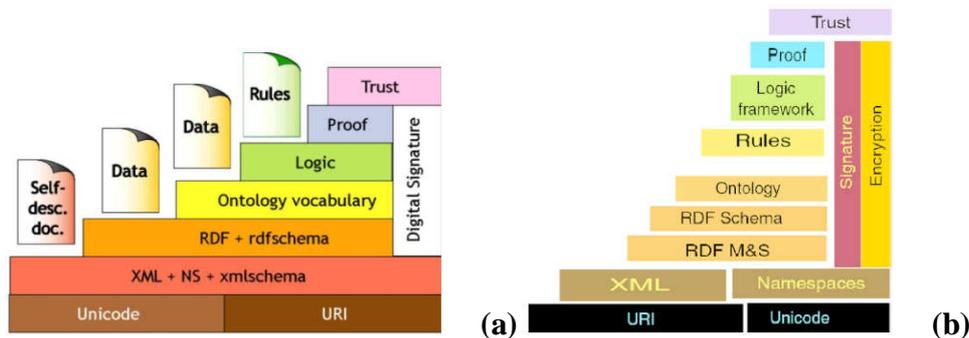


Figure 1.1. L'architecture en couche de la vision du Web sémantique [1].

La couche qui a actuellement atteint une maturité suffisante est la couche d'ontologie sous la forme du langage OWL. La prochaine étape dans le développement du web sémantique est la réalisation des couches de règles, logique et preuve qui seront développées en dessus de la couche d'ontologie. Pour mieux comprendre l'utilité de la couche de règles qui présente un formalisme fondé sur la programmation logique dans l'infrastructure du web sémantique et plus particulièrement dans les applications de recherches d'informations, nous exposons en détail les différentes couches de cette architecture.

➤ La couche URI et Unicode

L'URI est un protocole simple et extensible pour identifier, d'une manière unique et uniforme, toute ressource sur le web. Il s'agit d'un aspect central de l'infrastructure, c'est pour cette raison que cet élément se trouve à la base de l'architecture en couches proposée.

Il existe plusieurs types d'identification de ressource. En effet, un URI peut être classé, en 3 catégories, selon qu'il soit destiné à la localisation, au nommage ou au deux. Le terme URL désigne un sous-ensemble d'URI qui identifie les ressources via une représentation de leur mécanisme d'accès, plutôt que par le nom ou autre attribut de cette dernière.

Par ailleurs, il est à noter que les données sont toujours encodées avec le jeu de caractères Unicode pour un maximum d'interopérabilité. C'est pourquoi cet élément figure dans cette couche de bas niveau, au même titre que l'URI.

➤ **La couche XML**

À ce niveau d'architecture, nous ne sommes toujours pas au point d'affecter une sémantique à l'information, c'est-à-dire de la décrire et lui donner un sens. Il s'agit seulement d'une couche syntaxique de bas niveau qui permet de structurer les données et organiser selon un format de message standard, et ce, grâce au langage de balisage extensible XML.

XML fournit une syntaxe pour décrire la structure du document, créer et manipuler des instances des documents. Il utilise l'espace de nommage (namespace) afin d'identifier les noms des balises (tags) utilisées dans les documents XML. Le schéma XML permet de définir les vocabulaires pour des documents XML valides. Cependant, XML n'impose aucune contrainte sémantique à la signification de ces documents, l'interopérabilité syntaxique n'est pas suffisante pour qu'un logiciel puisse « comprendre » le contenu des données et les manipuler d'une manière significative.

Jusqu'à ce niveau de l'architecture, le problème de l'interprétation de la sémantique de l'information par la machine n'est toujours pas résolu, XML et XML Schéma sont largement répandus dans les applications orientées Web mais ils restent limités car ils ne disposent pas d'une sémantique formelle [3]. Dans ce qui suit, nous présentons les standards de représentation des connaissances qui enrichissent les informations apportées par XML, à savoir RDF/RDFS et OWL.

➤ **La couche RDF**

Après avoir référencé les ressources avec le protocole URI et structuré les informations avec le XML, l'étape suivante consiste à les annoter, afin de les doter d'un sens interprétable par la machine. C'est justement le rôle de la couche RDF et RDF-S dans l'architecture du Web sémantique.

Les couches RDF et RDF Schéma sont considérées comme les premières fondations de l'interopérabilité sémantique. Elles permettent de décrire les taxonomies des concepts et des propriétés. RDF fournit un moyen d'insérer de la sémantique dans un document, où l'information est conservée principalement sous forme de déclarations RDF. Le schéma RDFS décrit les hiérarchies des concepts et des relations entre les concepts, les propriétés et les restrictions domaine/co-domaine pour les propriétés. Les trois langages XML, RDF et RDFS sont présentés plus en détail dans la partie suivante.

➤ La couche ontologique

La couche Ontologie décrit des sources d'information hétérogènes, distribuées et semi-structurées en définissant le consensus du domaine commun et partagé par plusieurs personnes et communautés. Les ontologies aident la machine et l'humain à communiquer avec concision en utilisant l'échange de sémantique plutôt que de syntaxe seulement.

➤ La couche de règles

Les règles sont aussi un élément clé de la vision du Web sémantique. La couche de règles offre la possibilité et les moyens de l'intégration, de la dérivation, et de la transformation de données provenant de sources multiples, etc. Les règles dans cette couche permettent un raisonnement automatisé sur des données structurées afin de déduire toutes les inférences possible.

➤ La couche Logique

La couche Logique se trouve au-dessus de la couche ontologie. Certains considèrent ces deux couches comme étant au même niveau, c.-à-dire des ontologies basées sur la logique et permettant des axiomes logiques. En appliquant la déduction logique, on peut inférer de nouvelles connaissances à partir d'une information explicitement représentée.

Une large variété de logiques a été conçue jusqu'à présent. Étant le formalisme le mieux apprécié dans la représentation de la connaissance, la logique de description est celle qui est généralement la plus adoptée pour la représentation des règles d'inférences.

Historiquement, les logiques de descriptions ne sont pas les premiers outils pour représenter les ontologies. D'abord il y a eu les Frames, des Graphes conceptuels, et enfin les logiques de description.

➤ Frames

Le modèle des Frames est un classique de l'Intelligence Artificielle, et a été initialement proposé comme langage de représentation d'ontologies par T. GRUBER. Le principe de ce modèle est de décomposer les connaissances en classes (ou frames) qui représentent les concepts du domaine. À un frame est rattaché un certain nombre d'attributs (slots), chaque attribut pouvant prendre ses valeurs parmi un ensemble de facettes (facets). Une autre façon de présenter ces attributs est de les considérer comme des relations binaires entre classes dont le premier argument est appelé domaine (domain) et la deuxième portée (range) [12]. Des instances des classes, correspondant à l'extension de chaque concept, peuvent être ajoutées, ainsi que des fonctions qui sont des types particuliers de relations liant un ensemble de classes

à une valeur calculée à partir des valeurs des attributs des classes. La spécification de propriétés conceptuelles des attributs (ou relations) recourt à des formules de la logique du premier ordre.

➤ Graphes conceptuels

Introduit par J. SOWA au début des années 80, le modèle des Graphes Conceptuels (GC) appartient à la famille des réseaux sémantiques. Les réseaux sémantiques modélisent les connaissances sous forme de graphes, les nœuds étant associés à des concepts et les arêtes à des relations. Le modèle des GCs se décompose en deux parties [17]:

- ✓ Une partie terminologique dédiée au vocabulaire conceptuel des connaissances à représenter, c'est-à-dire les types de concepts, les types de relations et les instances des types de concepts.

Cette partie correspond à la représentation du modèle conceptuel mais intègre également des connaissances sur la hiérarchisation des types de concepts et de relations.

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

➤ Logiques de descriptions

Les logiques de description (LDs) découlent directement des travaux fondateurs de Bachmann et de son système KL-ONE. Depuis le début des années 90, la recherche en logique de description s'est considérablement développée.

Les logiques de description peuvent être considérées comme un fragment de la logique du premier ordre, dans lequel les formules ont une variable libre pour les descriptions de concept et deux variables libres pour les descriptions de relations [18].

Une LD est composée de deux parties : un langage terminologique TBOX et un langage assertionnelle ABOX. Le langage assertionnelle est dédié à la description de faits et le langage terminologique à la description de concepts et de rôles. La principale tâche de raisonnement au niveau terminologique est de calculer les relations de subsomption entre concepts [19].

1) Les constructeurs des LDs

Les entités de base qui sont définies et manipulées dans une logique de descriptions sont les concepts et les rôles. Un concept permet de représenter un ensemble d'individus, et un rôle représente une relation binaire entre concepts.

Un concept et un rôle possèdent une description structurée, élaborée à partir d'un certain nombre de constructeurs.

Il existe de nombreux constructeurs permettant de former toute une famille de logiques de description. Par ailleurs, la logique de description SHIQ [20] regroupe un ensemble plus riche de constructeurs. Le tableau (Tableau 1.1) décrit les plus importants.

2) Mécanisme de raisonnement

Le mécanisme de raisonnement de base pour les LDs est la classification de concepts dans la hiérarchie des concepts, réalisée par un algorithme de classification appelé classifieur [21]. Généralement, les LDs sont hautement expressives, disposent d'une sémantique claire et procurent des mécanismes d'inférences efficaces. Elles constituent, ainsi, le formalisme retenu par le Web sémantique pour la représentation des ontologies [22].

La sémantique d'un concept défini par les constructeurs des logiques de description est donnée par le tableau ci-dessous:

Constructeur	Syntaxe	Sémantique (I: une interprétation)
Universel	\top	ΔI (ΔI : ensemble de tous les objets)
Absurde	\perp	\emptyset
Négation	$\neg C$	$\Delta I \setminus CI$
Conjonction	$C \cap D$	$CI \cap DI$
Disjonction	$C \cup D$	$CI \cup DI$
Restriction universelle	$\forall r.C$	$\{x \in \Delta I / \forall y, (x,y) \in r \Rightarrow y \in CI\}$
Restriction existentielle	$\exists r.C$	$\{x \in \Delta I / \exists y, (x,y) \in r \text{ et } y \in CI\}$
Cardinalité minimum	$(\geq n \ r)$	$\{x \in \Delta I / \{y (x,y) \in r\} \geq n\}$
Cardinalité maximum	$(\leq n \ r)$	$\{x \in \Delta I / \{y (x,y) \in r\} \leq n\}$
Conjonction des rôles	$r \cap s$	$\{(x,y) \in \Delta I * \Delta I / (x,y) \in r \wedge (x,y) \in s\}$
Disjonction des rôles	$r \cup s$	$\{(x,y) \in \Delta I * \Delta I / (x,y) \in r \vee (x,y) \in s\}$

Table 1.1. Syntaxe et sémantique des descriptions de concepts [2].

2.1.2. Langages de représentation de l'ontologie

Plusieurs langages de spécification d'ontologies (ou langages d'ontologies) ont été développés pendant les dernières années, et ils deviendront sûrement des langages d'ontologie dans le contexte du Web sémantique. Certains d'entre eux sont basés sur la syntaxe de XML, tels que XOL (Ontology Exchange Language), SHOE (Simple HTML Ontology Extension -

qui a été précédemment basé sur le HTML), OML (OntologyMarkupLanguage), RDF (Resource Description Framework), RDF Schéma. Les deux derniers sont des langages créés par des groupes de travail du World Wide Web Consortium (W3C).

En conclusion, trois langages additionnels sont établis sur RDF(S) pour améliorer ses caractéristiques: OIL (OntologyInference Layer), DAML+OIL et OWL (Web OntologyLanguage). La figure ci-dessous représente les rapports principaux entre tous ces langages sous la forme d'une pyramide des langages du Web sémantique.

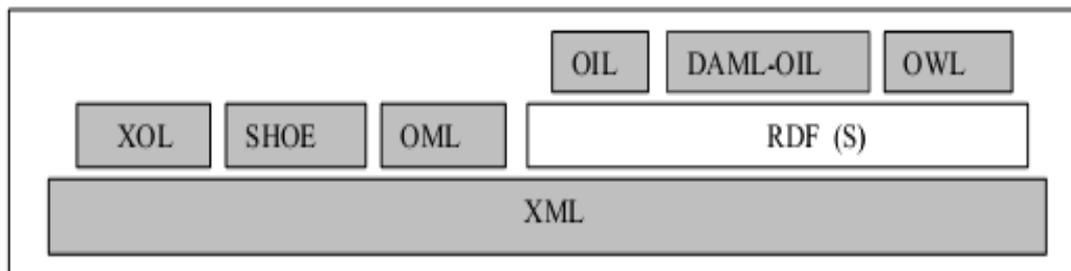


Figure 1.2. La pyramide des langages basés Web.

➤ RDF

Est un langage d'assertion et d'annotations. Les assertions affirment l'existence de relations entre les objets. Elles sont donc adaptées à l'expression des annotations que l'on veut associer aux ressources du Web. RDF est un langage formel qui permet d'affirmer des relations entre des « ressources ». [23]

Le modèle RDF définit trois types d'objets:

- **Ressources** : Cela peut être n'importe quel objet référencé par une URI, qu'il concerne le web (Page HTML, document PDF, fichier multimédia...), ou non (Personne, Région, Etc.).
- **Propriétés** : une propriété est un attribut, un aspect, une caractéristique qui s'applique à une ressource (titre, couleur, taille, auteur, etc.).
- **Valeurs** : C'est la valeur qui sera affectée à la propriété de la ressource. Cette affectation peut être soumise à certaines restrictions.

➤ RDFS (RDF Schéma)

Ce langage offre un niveau de structuration supérieur à RDF. Il a ajouté la possibilité de définir des hiérarchies de classes et de propriétés dont l'applicabilité et le domaine de valeurs peuvent être contraintes. A chaque domaine applicatif peut être ainsi associé un schéma identifié par un préfixe particulier et correspondant à une URI. Les ressources instances sont

ensuite décrites en utilisant le vocabulaire donné par les classes définies dans ce schéma [23] et les applications peuvent alors leur donner une interprétation opérationnelle. Seulement, ce langage n'intègre pas de capacités de raisonnement mais offre des primitives de représentation de structures ou primitives ontologiques.

➤ DAML-OIL

DAML est un langage qui a comme but de fournir les fondations pour la génération suivante du Web sémantique. Le langage a adopté d'abord RDFS comme langage d'ontologie pour l'interopérabilité sémantique entre projets. Comme RDFS n'est pas assez expressif relativement aux exigences du Web sémantique, un nouveau langage nommé DAML-ONT a été développé en tant qu'extension de RDF avec les capacités d'un langage de représentation du savoir : orienté objet et basé sur cadre.

En même temps, un groupe des chercheurs (la plupart d'entre eux sont européens) au sein d'IST d'Union européenne, avec le même but, développe un autre langage d'ontologie appelé OIL. Ce langage a une syntaxe basée sur RDF et il est explicitement construit pour que sa sémantique puisse être spécifiée à travers une description logique très expressive, la logique de description de type SHIQ.

DAML+OIL est la combinaison de ces deux langages. Il hérite des avantages de ces deux langages. En conséquence, DAML+OIL est un langage très expressif et lisible par la machine ainsi que par un être humain avec une syntaxe basée sur RDF.

➤ OWL

OWL signifie Web Ontology Language, défini par le W3C dans, le langage OWL est basé sur la recherche effectuée dans le domaine de la logique de description. OWL permet de décrire des ontologies, c'est-à-dire qu'il permet de définir des terminologies pour décrire des domaines concrets. Une terminologie se constitue de concepts et de propriétés (aussi appelés rôles en logiques de description). Un domaine se compose d'instance de concepts.

Le langage OWL se compose de trois sous-langages qui proposent une expressivité croissante, chacun conçu pour des communautés de développeurs et des utilisateurs spécifiques: OWL Lite, OWL DL, OWL Full. Chacun est une extension par rapport à son prédécesseur plus simple.

- **OWL Lite** répond à des besoins de hiérarchie de classification et de fonctionnalités de contraintes simples de cardinalité 0 ou 1. Une cardinalité 0 ou 1 correspond à des relations

fonctionnelles, par exemple, une personne a une adresse. Toutefois, cette personne peut avoir un ou plusieurs prénoms, OWL Lite ne suffit donc pas pour cette situation.

- **OWL DL** concerne les utilisateurs qui souhaitent une expressivité maximum couplée à la complétude du calcul (cela signifie que toutes les inférences seront assurées d'être prises en compte) et la décidabilité du système de raisonnement (c'est-à-dire que tous les calculs seront terminés dans un intervalle de temps fini). Ce langage inclut toutes les structures OWL avec certaines restrictions, comme la séparation des types: une classe ne peut pas aussi être un individu ou une propriété. Il est nommé DL car il correspond à la logique descriptive.
- **OWL Full** se destine aux personnes souhaitant une expressivité maximale. Il a l'avantage de la compatibilité complète avec RDF/RDFS, mais l'inconvénient d'avoir un haut niveau de capacité de description, quitte à ne pas pouvoir garantir la complétude et la décidabilité des calculs liés à l'ontologie. Quant aux outils de construction d'ontologies, plusieurs éditeurs d'ontologie sont aujourd'hui disponibles tel que : Protégé, ONTOEDIT, OILED, WEBODE, Ils sont tous capables de produire des ontologies dans les langages DAML+OIL et RDFS et quelques-uns proposent une méthodologie de conception plus ou moins complète. Aujourd'hui, Protégé-OWL, est allé loin dans la construction et l'intégration des ontologies exprimées en OWL. Cet outil nous a servi davantage dans la construction de l'ontologie développée dans le cadre de ce travail **GynécOntologie**.

3. Ingénierie ontologique

L'ontologie est un terme emprunté à la philosophie. Dans cette discipline, l'ontologie est un ensemble d'objets existants. L'ontologie est identiquement une étude de l'être en tant qu'être, c'est-à-dire, une étude des propriétés générales de ce qui existe. Cependant, en informatique et d'après l'encyclopédie Wikipédia une ontologie est un ensemble structuré de concepts.

Les concepts sont organisés dans un graphe dont les relations peuvent être des relations sémantiques où des relations de composition et d'héritage (au sens objet) [06].

Le terme ontologie est utilisé depuis le début des années 1990 dans le domaine de l'intelligence artificielle, en particulier de l'ingénierie des connaissances et de la représentation des connaissances. Son champ d'application s'élargit considérablement et il fait désormais partie des objets de recherche courants. Une ontologie est un système formel dont l'objectif est de représenter les connaissances d'un domaine spécifique au moyen d'éléments de base, les concepts, définis et organisés les uns par rapport aux autres [07].

3.1. 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 [08].

3.2. Définitions

De nos jours, il n'existe aucun accord sur le sens exact du terme ontologie. Il est difficile de le définir d'une façon définitive. Le mot est en effet employé dans des contextes très différents touchant la philosophie, la linguistique ou l'intelligence artificielle. De nombreuses définitions ont été offertes pour donner un éclaircissement sur ce terme [10].

Du point de vue de l'ingénierie des connaissances, différentes définitions de l'ontologie ont été données. La plus référencée et aussi la plus synthétique est sans doute celle formulée par GRUBER « Une ontologie est une spécification formelle et explicite d'une conceptualisation partagée » [11] et [12].

En 1993, Gruber propose cette définition afin de montrer qu'une ontologie est un ensemble de définitions, de primitives, de représentations de connaissances spécifiques au contenu : classes, relations, fonctions et constantes d'objet. Cette définition est la plus référencée et la plus synthétique. Nous allons la conserver dans la suite de notre travail. Le terme **conceptualisation** réfère dans cette définition à une abstraction d'un phénomène du monde, obtenue en identifiant les concepts appropriés à ce phénomène. « **Formelle** » indique que les ontologies sont interprétables par la machine. Cependant, « **Spécification explicite** » signifie que les concepts de l'ontologie et les contraintes liées à leur usage sont définis de façon déclarative. Et enfin le terme « **Partagé** » signifie que l'ontologie capture la connaissance consensuelle. (Elle n'est pas l'objet d'un individu mais elle est reconnue par un groupe).

3.3. 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 [13] [14].

➤ 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 [15].

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 subsumption : 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 subsumption 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.

➤ Les instances

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

➤ Les relations

Une relation permet de lier des instances de concepts ou des concepts génériques. Elles sont caractérisées par un terme ou plusieurs, et une signature qui précise le nombre d'instances de concepts que la relation lie, leurs types et l'ordre des concepts, c'est-à-dire la façon dont la relation doit être lue.

Tout comme les concepts, les relations peuvent être spécifiées par des propriétés. Les propriétés intrinsèques d'une relation sont :

- ✓ **Les propriétés algébriques** : symétrie, réflexivité, transitivité, antisymétrie, anti réflexivité.
- ✓ **La cardinalité** : il s'agit du nombre possible de relations de ce type pouvant exister entre les mêmes concepts (ou instances de concept). Les relations portant une cardinalité représentent souvent des attributs. Par exemple, une pièce a au moins une porte, un humain a entre zéro et deux jambes.

Les propriétés liant deux relations sont principalement:

- ✓ **L'incompatibilité** : Deux relations sont incompatibles si elles ne peuvent lier les mêmes instances de concepts. Par exemple, les relations fils-de et père-de sont incompatibles.
- ✓ **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 . Par exemple, les relations fils-de et père-de sont inverses l'une de l'autre.
- ✓ **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. Par exemple, l'appartenance et la non appartenance sont exclusifs.

➤ Les axiomes

Les axiomes forment des contraintes sémantiques pour le raisonnement et donnent une provision d'une conceptualisation. Ils prennent la forme d'une théorie logique.

3.4. Utilisation d'une ontologie

Les ontologies peuvent servir à la communication entre êtres humains ou entre humains et machines, à la spécification de systèmes, à l'interopérabilité qui est une spécialisation de la communication entre ordinateurs ; ainsi qu'à l'indexation et la recherche de documents.

Nous pouvons énumérer un certain nombre d'utilités, notamment:

- **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 consistante 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.

- **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.

- **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 [16].

- **L'indexation et la recherche d'information**

Dans le web sémantique, d'une façon générale, les ontologies sont utilisées pour indexer et décrire les ressources utilisées. Cela permet une plus grande précision dans les résultats des recherches ou d'assignation des ressources.

3.5. Les typologies d'ontologies

Pour la typologie des ontologies, on peut distinguer différents niveaux d'ontologies selon le but pour lequel elles sont conçues. La classification de [24] repose sur deux critères : le sujet et la structure d'une conceptualisation et on peut citer les plus utilisées :

➤ **Les ontologies de domaine**

Elles sont construites sur un domaine particulier de la connaissance. Les ontologies de domaine fournissent des vocabulaires au sujet des concepts dans un domaine et leurs relations au sujet des activités qui ont lieu dans ce domaine, et au sujet des théories et des principes élémentaires régissant ce domaine. Plusieurs ontologies de domaines existent déjà, telle que MENELAS dans le domaine médical. Entreprise est un autre exemple décrivant le domaine de l'entreprise.[24]

➤ **Les ontologies d'application**

Aussi appelée ontologie de domaine-tâche, Ce sont les ontologies les plus spécifiques, elles contiennent les connaissances requises pour une application particulière permettant ainsi de modéliser une activité spécifique dans un domaine donné.

➤ **Les ontologies génériques**

Sont aussi appelées Ontologies de haut niveau ou ontologies Top, elles décrivent des concepts généraux, indépendants d'un domaine ou d'un problème particulier. Elles permettent de formaliser les aspects temporels ou spatiaux des objets du monde réel, soit les concepts de haute abstraction tels que: les événements, les états, les processus, les actions, le temps, l'espace, les relations, les propriétés. [25]

➤ **Les ontologies de tâches**

L'ontologie de tâche décrit les connaissances portant sur tâches et/ou des activités particulières. Ces ontologies fournissent un ensemble de termes au moyen desquels on peut décrire au niveau générique comment résoudre un type de problème. Elles incluent des noms génériques (objectif, contrainte...), des verbes génériques (classer, sélectionner,...), des adjectifs génériques (assigné,...) et autres dans les descriptions de tâches [26].

D'autre part, Les ontologies peuvent être distinguées en fonction du degré de formalisme utilisé pour les exprimer. Uschold et Grüninger (Uschold&Grüninger, 1996) proposent une classification contenant les quatre catégories : Très informelles ; Semi-informelles ; Semi-formelles et Rigoureusement formelles.

- a) **Très informelle** : elle est exprimée en langue naturelle (sémantique ouverte).
- b) **Semi-informelle** : l'ontologie est exprimé sous une forme restreinte et structurée de langage naturel pour augmenter la clarté et pour réduire l'ambiguïté.
- c) **Semi-formelle** : elle est exprimée dans un langage artificiel défini formellement.
- d) **Rigoureuse formelle** : l'ontologie est défini avec une sémantique formelle, permettant la théorie et la preuve.

3.6. Construction d'une ontologie

3.6.1. 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.

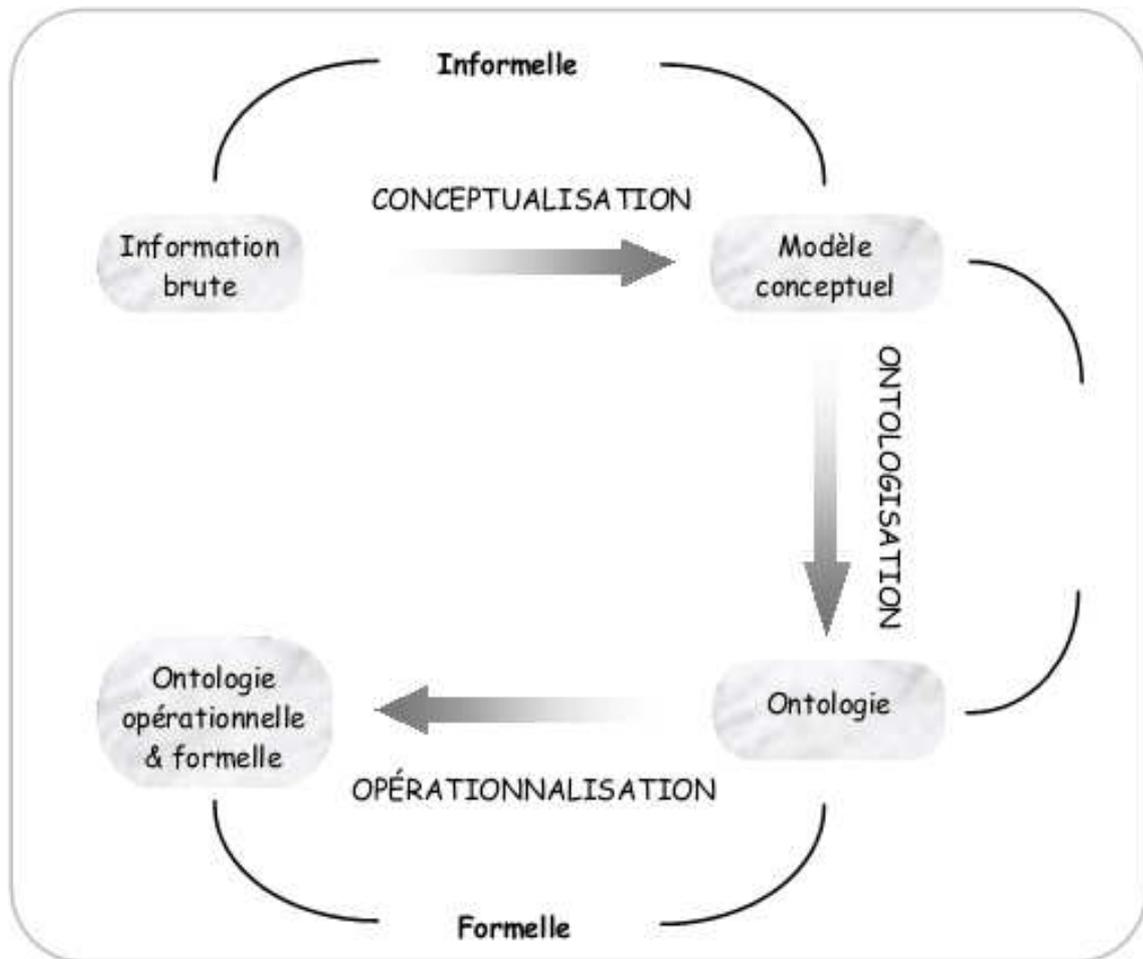


Figure 1.3. Processus de construction d'ontologie [29].

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.

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.

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). Le caractère semi-formel d'une ontologie lui interdit d'être utilisée telle quelle dans un SBC. 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.

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.

3.6.2. 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.

3.6.2.1 Tove

TOVE (Toronto Virtual Enterprise) développé par l'université de Toronto, cette méthodologie repose sur les expériences de développement d'une entreprise [12] [16]. 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 une 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.

3.6.2.2 Enterprise

Uschold et King's [15], proposent le squelette d'une méthode basé sur l'expérience de déconstruction 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 ?

3.6.2.3. 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 ? À quels types de questions les informations de l'ontologie doivent fournir des réponses ? 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 ») [12]. 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 associés aux LDs, et 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.

4. Outils pour le Web Sémantique

Les outils de développement d'ontologies qui existent sur le marché aujourd'hui sont divers et variés à bien des ménagements. Cet état de choses provoque beaucoup d'interrogations lorsque vient le moment d'en choisir un pour construire une nouvelle ontologie (Gómez-Pérez, et al, 2004): L'outil offre-t-il une assistance au développement ? L'outil dispose-t-il d'un moteur d'inférence ? Quels langages d'ontologies l'outil supporte-t-il ? L'outil permet-il d'importer/exporter des ontologies ? L'outil offre-t-il un support à la réutilisation d'ontologies existantes ? L'outil offre-t-il un support graphique à la construction des ontologies ? Les réponses à toutes ces questions pourraient aider à prendre une décision dans le choix de l'un ou l'autre outil.

Dans cette section, nous allons présenter quelques outils d'ingénierie ontologique. Ils permettent à l'utilisateur de créer des ontologies de manière indépendante des langages de représentation et de prendre en charge la phase d'opérationnalisation de l'ontologie en l'exportant dans des langages informatisés standards.

4.1. Editeurs d'ontologie

4.1.1 Protégé [27]

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.

4.1.2.OILEd

L'éditeur OILEd[28] a été développé en 1991 sous la responsabilité de l'université de Manchester pour éditer des ontologies dans les langages de représentation OIL, puis DAML+OIL. Il est orienté vers la représentation en logique de description expressive et, à ce titre, fournit tous les éléments d'interface permettant de spécifier des hiérarchies de concepts et de rôles, les restrictions sur les rôles et les instances.

Il peut être connecté à un raisonneur de logique des descriptions tel que FaCT et RACER, capable de tester la satisfiabilité des ontologies construites ou d'explicitier de nouvelles relations de subsomption entre concepts complexes.

4.1.3 WebODE (2003)

WebODE a été développé par le groupe Ontological Engineering du département d'Intelligence artificielle de la faculté d'Informatique de l'université polytechnique de Madrid. Un éditeur qui assurait le support de METHONTOLOGY, la méthodologie proposée

par ce laboratoire. WebODE est composé de plusieurs modules : un éditeur d'ontologie qui intègre la plupart des services nécessaires à la construction d'ontologies (édition, navigation, comparaison, fusion, raisonnement...), un système de gestion des connaissances à base ontologique, un outil pour annoter les ressources du web et un éditeur de services pour le Web sémantique.

4.2. 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.

Une procédure générale de raisonnement est illustrée dans la figure suivante (figure 1.6) 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.

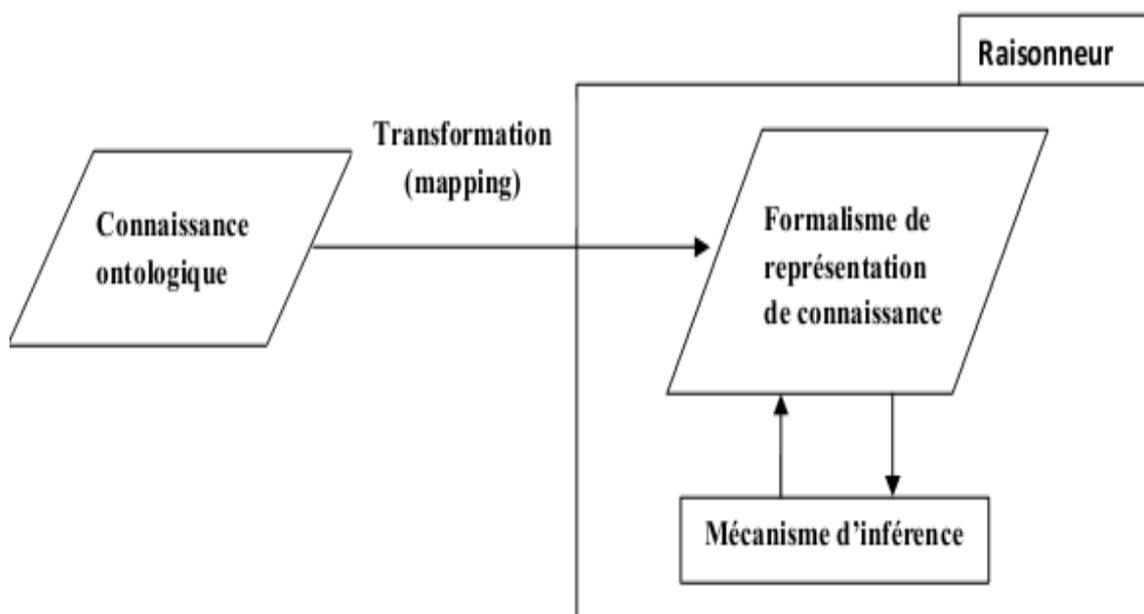


Figure 1.4. Architecture abstraite d'un raisonneur OWL [27].

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.

4.2.1 Racer [30]

Le système Racer (RenamedABox 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 RacerproqueryLanguage) 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.

- ✓ 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, il utilise JRacer ;
- ✓ Racer peut être utilisé dans Protégé OWL.

4.2.2. Pellet [31]

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 OpenSource et offre des évolutions fréquentes. Pellet travaille sur des ontologies décrites en RDF ou OWL et permet les requêtes avec RDQL et SPARQL sur la ABox et la TBox.

5. Conclusion

Le Web sémantique est une vision futuriste du Web dans laquelle l'information est donnée avec une signification explicite afin de faciliter le traitement automatique de cette information.

Nous venons de voir dans ce chapitre un rapide aperçu sur les notions d'ontologies et le Web Sémantique. Nous nous sommes attachés à définir les ontologies en présentant leurs composants, utilisations, typologies, ainsi que leurs constructions et les outils pour l'édition et l'inférence.

Dans le prochain chapitre de ce mémoire, nous présenterons la notion de règles, l'intérêt de combiner ontologie et règles, ainsi que les différentes approches de combinaison et enfin les travaux existants pour le raisonnement sur des ontologies hybrides.

Chapitre 2



Raisonnement sur
les ontologies
hybrides

1. introduction

La sémantique formelle du langage OWL permet l'application des techniques de raisonnement pour effectuer des dérivations logiques. Ces dérivations sont effectuées par des raisonneurs capables de gérer et d'utiliser la sémantique du langage de l'ontologie.

L'ontologie seule ne peut pas modéliser toutes les connaissances relatives à un domaine donné. Ce qui implique l'intégration de règles pour augmenter l'expressivité d'un côté, et enrichir le raisonnement d'un autre.

Dans ce chapitre, nous allons poursuivre les recherches en visant à intégrer les règles et la logique de description tout en maintenant un support logique assez efficace. Nous présentons dans la section 2 de ce chapitre, les paradigmes de modélisation proposés pour le web sémantique. Nous expliquons dans les sections 3 et 4 le besoin de combiner ontologie et règles. Aussi nous exposerons les travaux connexes portant sur le raisonnement sur une ontologie enrichie de règles.

2. Les paradigmes de modélisation

Deux paradigmes différents de modélisation ont été proposés pour le web sémantique [32]. Le paradigme classique fondé sur la logique classique et la programmation logique fondé sur la notion de règles. Chaque paradigme s'adapte mieux à quelques types particuliers de connaissance et supporte des services de raisonnement spécifiques.

2.1 Paradigme classique

Ce premier paradigme est basé sur la notion de la logique classique (standard) comme: la logique propositionnelle, la logique du premier ordre et la logique de descriptions. Le paradigme classique est incorporé dans les langages du web sémantique standardisés par le W3C tels que RDF, RDFS, OWL.

Les LDs sont bien adaptées à la partie terminologique du domaine. Dans ce cas la sémantique des ontologies OWL peut être manipulée par les systèmes de raisonnement sur la LD comme Pellet [31] et Racer [34].

2.2. La programmation logique

La programmation logique est une famille de formalismes de représentation de connaissances centrés autour la notion de règles. Dans ce paradigme, un sous ensemble de la sémantique d'OWL est transformé en règles utilisées par un moteur de règle pour inférer des connaissances implicites. Les règles sont utiles pour représenter la partie déductive du domaine.

Plusieurs études ont été menées pour explorer les points communs, et aussi expliquer les différences entre ces deux paradigmes. Le tableau ci-dessous montre les différences les plus élémentaires tel que : la monotonie, les identifiants, l'environnement, l'hypothèse du monde ouvert, la complexité, ...etc.

Différences	Paradigme Classique	Programmation logique
Monotonie	Raisonnement monotone	Raisonnement non-monotone
Les Identifiants	Identifiant = URI	<u>Un</u> identifiant représente <u>un</u> individu "Unique Name Assumption" Hypothèse du nom unique
Environnement	Environnement ouvert et libre (Web sémantique)	Environnement fermé et hautement structuré (BDD)
Le Monde Ouvert	Les interprétations implicites ne sont pas exclues.	L'hypothèse du monde fermé nie toute information non explicite.
Complexité	Complexité élevée mais fixe.	Complexité en temps polynomial (cas simples et fixes) complexité en temps exponentiel.
Inférences	Supporte les inférences	Ne supporte pas les inférences à cause de son hypothèse du nom unique.
informations incomplètes	<u>Flexible</u> : Informations incomplètes autorisées	informations incomplètes non permises

Tableau 2.1. Paradigme classique et programmation logique [40].

3. Les tendances actuelles des langages du Web sémantique

Le défi actuel pour le Web est d'évoluer vers un Web sémantique, dans lequel les informations ne seraient plus stockées mais "comprises" par les ordinateurs afin d'apporter à l'utilisateur ce qu'il cherche vraiment. D'après Tim Berners-Lee, il s'agit d'arriver à un Web "intelligent".

Le Web sémantique se veut un Web de demain dont le contenu peut être appréhendé et exploité par des machines.

Le langage d'ontologie Web OWL est basé sur la logique de description, une famille de formalisme de représentation de connaissance. OWL a été appliqué avec succès. Mais, la spécification des couches finales du Web sémantique reste encore une question ouverte.

D'après Berner-Lee « **Le défi du Web sémantique est de fournir un langage qui exprime les données et les règles pour le raisonnement sur les données et qui permet aux règles de n'importe quel système de représentation des connaissances à être exportés sur le Web.**»

Les formalismes à base de règles fondés sur la programmation logique ont été proposés comme une solution possible, donc l'ajout d'une couche de règles au-dessus de la couche d'ontologie est considérée comme une tâche centrale dans le développement de l'infrastructure du web sémantique. Ils existent actuellement deux tendances importantes des langages du Web sémantiques. Elles correspondent respectivement aux paradigmes de logique de description et de règles: le langage d'ontologie Web OWL et le langage RuleML.

De plus, bien qu'OWL DL soit plus adapté pour représenter et raisonner sur les ontologies, et RuleMLDatalog pour exprimer les règles, quelques applications nécessitent l'intégration des deux langages.

4. Intérêt de combinaison des règles et ontologies

La combinaison d'un langage de règles avec une ontologie offre plusieurs points positifs. La figure suivante montre que l'ajout des règles à l'ontologie fournit une couche supplémentaire d'expressivité.

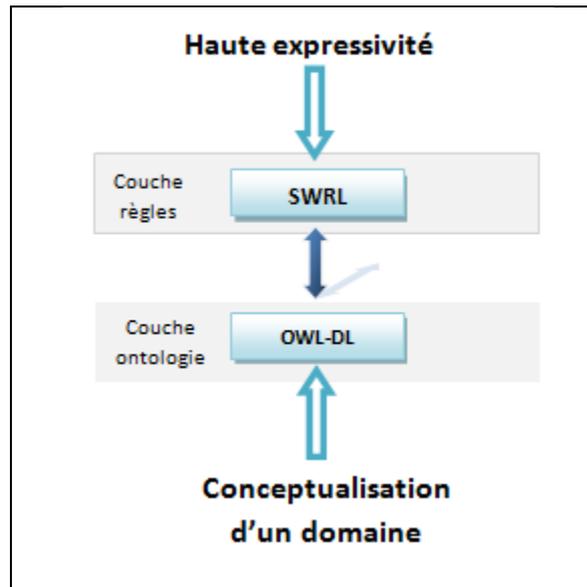


Figure 2.1. Intérêt de combinaison règles et ontologie.

- **Assurer une plus haute expressivité relationnelle**

OWL fournit un ensemble riche de primitives pour exprimer les concepts (axiomes logique), cependant l'ensemble des primitives pour des rôles n'est pas suffisant pour des applications pratiques (OWL ne permet pas la définition des axiomes sur les propriétés). OWL peut seulement modéliser des domaines où les objets sont connectés d'une manière arborescente. Néanmoins, beaucoup d'applications exigent la modélisation des structures relationnelles complexes. Donc les règles peuvent servir comme des extensions des langages d'ontologie basée logique de description, permettant une définition des relations sémantiques plus riches.

- **Exprimer les prédicats d'arité plus grandes que deux**

Les constructeurs de modélisation de base du langage OWL sont les concepts et les rôles, qui correspondent à des prédicats unaires et binaires. Toutefois, de nombreuses relations sont rencontrées dans la pratique d'arité plus grande que deux.

- **Interrogation**

Les moteurs d'inférence sur la logique de description ont une faible performance de raisonnement et d'interrogation sur la ABOX. Donc il est intéressant de considérer la combinaison des LDs avec le paradigme de règle afin d'indiquer des requêtes d'instance expressive [33] [34].

5. Approches de combinaison

Bien que diverses méthodes aient été proposées pour combiner les règles et les ontologies. Jusqu'à présent, il n'existe pas de solution meilleure acceptée par la communauté des chercheurs.

Le principal point de conflit se pose dans l'intégration des ontologies et des règles. Les propositions existantes pour l'intégration des langages de règles avec des langages d'ontologie doivent être classifiées par le degré d'intégration, la figure 2.3 qui suit présente le mode d'intégration (homogène ou hybride) dans l'architecture en couche du Web sémantique :

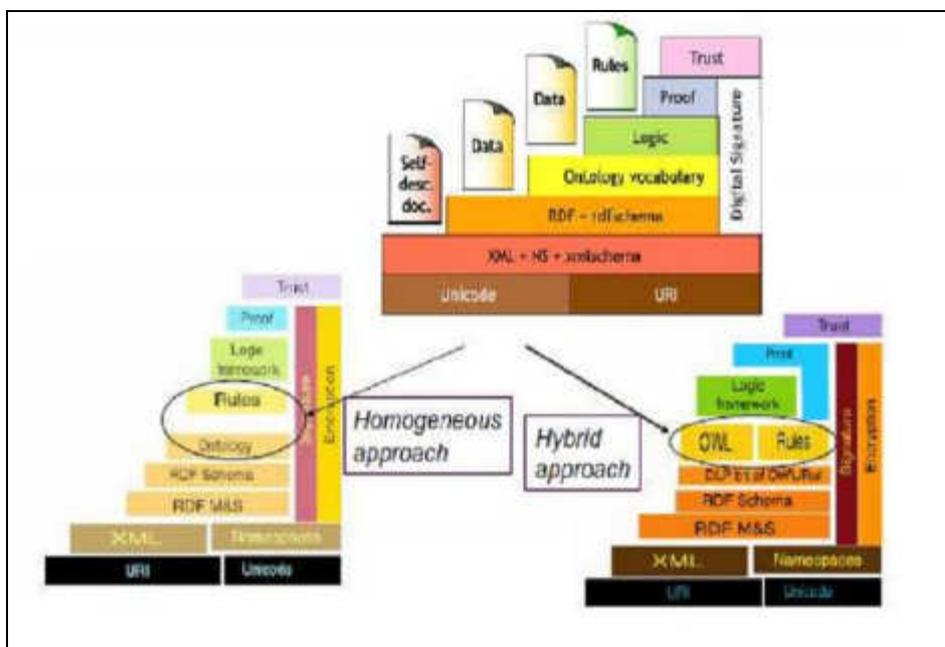


Figure 2.2. Les degrés d'intégrations [1].

5.1. L'approche hybride

Dans cette approche, il existe une séparation stricte entre les prédicats de règles et ceux de l'ontologie, qui sont seulement employés comme des contraintes dans les antécédents des règles. Le raisonnement est établi par la connexion des moteurs de règles avec les moteurs de raisonnement.

La combinaison hybride suit une architecture modulaire de deux sous-systèmes, donc chacun traite une partie distincte de la base de connaissances.

Plusieurs approches ont été proposées avec cette philosophie, telles que Integrating ASP with DL [35], DL+log [36], et CARIN [37], AL-log [36], et dl-programs.

5.2. Approche homogène

Dans cette approche, les règles et les ontologies sont implantées dans le même langage logique (par exemple le langage SWRL ou DLP) sans faire une distinction à priori entre les prédicats de règles et les prédicats d'ontologie. Les ontologies et les règles sont traitées d'une manière homogène, comme un nouveau langage logique.

L'idée générale est que les règles peuvent utiliser des prédicats unaires et binaires de l'ontologie c'est-à-dire les classes et les propriétés ainsi, un nouveau raisonneur est nécessaire, capable de gérer le nouveau langage homogène qui émerge.

5.2.1. Les règles : SWRL

Les règles sont utiles pour diverses raisons : nous pouvons les réutiliser s'ils existent déjà, et ils offrent une très haute expressivité au langage OWL. Dans certain temps, il est plus simple d'écrire des règles avec un langage de règles.

SWRL (Semantic Web RuleLanguage) [38] est un langage de règles pour le web sémantique. C'est un langage qui enrichit la sémantique d'une ontologie définie en OWL. SWRL est la combinaison du langage OWL DL et le langage Rule ML (RuleMarkupLanguage).

Le langage SWRL étend le langage OWL en ajoutant la notion de règles de déduction explicites, interprétées comme des clauses de **Horn**. En effet, SWRL permet contrairement à OWL, de manipuler des instances par des variables ($?x$, $?y$, $?z$). Il ne permet pas de créer des concepts ni des relations, il permet simplement d'ajouter des relations suivant les valeurs des variables et la satisfaction de la règle.

OWL et SWRL sont les langages fondamentaux du Web sémantique. OWL est un langage d'ontologie développé pour construire des ontologies fournissant un haut niveau d'expressivité pour le contenu du Web. SWRL est basée sur OWL, il garde la puissance d'OWL-DL mais au prix de la décidabilité. SWRL permet aux utilisateurs d'écrire des règles pour raisonner sur les individus et pour en déduire de nouvelles connaissances sur ces individus.

➤ La syntaxe

La syntaxe SWRL définit une règle qui est une relation d'implication entre un antécédent (corps) et un conséquent (tête). Si les conditions spécifiées dans l'antécédent sont vérifiées c'est-à-dire si l'on a pu démontrer la partie corps, alors les conditions spécifiées dans le conséquent le sont aussi (antécédent \Rightarrow conséquent).

L'antécédent et le conséquent d'une règle sont des conjonctions d'atomes. Les atomes peuvent avoir les formes:

- ✓ $C(x)$ où C est une description OWL, x est soit une variable, un individu OWL ou encore des valeurs de données (datavalues) de OWL.
- ✓ $P(x,y)$ où P est une propriété OWL (Object_property ou data_type_property), x est soit une variable ou un individu OWL et y est soit une variable, un individu OWL ou encore des valeurs de données de OWL.
- ✓ $\text{sameAs}(x,y)$, $\text{differentFrom}(x,y)$ où x et y sont des variables ou des individus OWL
- ✓ $\text{builtin}(r,x,\dots)$ où r est une relation built-in et x,\dots Sont des valeurs de données OWL.

Le fonctionnement d'une règle est basé sur le principe de satisfiabilité de l'antécédent ou du conséquent. Pour une règle, il existe trois cas de figure :

- L'antécédent et le conséquent sont définis. Si l'antécédent est satisfait alors le conséquent doit l'être.
- L'antécédent est vide, cela équivaut à un antécédent satisfait ce qui permet de définir des faits.
- Le conséquent est vide, cela équivaut à un conséquent insatisfait, l'antécédent ne doit pas être satisfiable.

➤ **Edition des règles SWRL**

Une des étapes clés de l'interopérabilité de règles sur le Web est le langage SWRL, qui vise à être le langage de règles standard du web sémantique. Un des avantages de Protégé est de pouvoir attacher un grand nombre de plugins venant notamment enrichir la représentation de l'ontologie et permettant sa validation.

SWRL Tab est une extension de Protégé OWL, qui prend en charge l'édition et l'exécution des règles SWRL. Il fournit une interface très interactive pour l'édition des règles prenant en charge l'ensemble des fonctionnalités du langage SWRL. Nous expliquerons dans le prochain chapitre le détail du fonctionnement de cet éditeur. Les moteurs de règles tels que Jess peuvent être intégrés avec cet éditeur. Cette intégration permet de fournir un raisonnement à base de règles plus riche pour le Web sémantique.

➤ **Moteur d'Inférences sur SWRL**

L'exécution des règles SWRL nécessite un moteur de règles. Ce dernier applique les faits sur les règles pour déduire des nouveaux faits. De nombreux moteurs d'inférences supportent SWRL: Bossam, Hoolet, KAON2 [42], Pellet, Racer, et Sesame. Ils suivent trois types d'approches:

1. Traduire SWRL en logique du premier ordre (Hoolet) ;
2. Traduire OWL-DL en règles et appliquer un algorithme de chaînage avant (Bossam);
3. Intégrer les règles SWRL dans le moteur d'inférences OWL-DL fondé sur les algorithmes des tableaux sémantiques (Pellet).

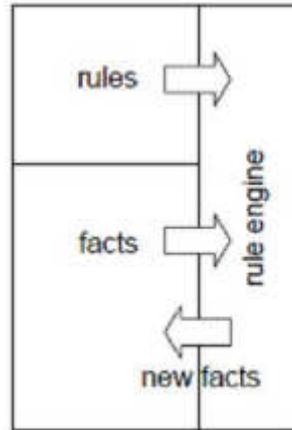


Figure 2.3. Principe d'exécution d'un moteur de règles [39].

Nous présentons les deux moteurs de règles Jess, et Bossam :

a) Jess

Jess est un moteur de règles et environnement de script écrit en Java par Ernest Friedman-Hill au Sandia National Laboratories au Livermore.

Jess est l'un des plus rapide moteurs de règle disponible, malgré sa petite taille et sa légèreté. Son puissant langage de script vous donne accès à toutes les API Java. Jess inclut un environnement de développement complet basé sur la plate-forme primée Eclipse.

Parmi les avantages de ce moteur, nous citons :

1. Expressivité du langage de représentation.
2. Facile à apprendre et à utiliser.
3. Flexibilité, stabilité et puissance.
4. Le mécanisme d'inférence (par défaut le chaînage avant/ chaînage arrière) et aux choix de l'utilisateur.
5. Il existe plusieurs outils supplémentaires pour faciliter la programmation avec Jess :

Plug in avec Eclipse, éditeur de règle pour JESS, et des plug in de protégé pour OWL et Jess (Jess_Tab).

6. Portabilité et rapidité.

b) Bossam

Bossam est un moteur d'inférence pour le web sémantique. Il est développé par l'Institut Coréen de Recherche en Electronique et en Télécommunication (ETRI). Il s'agit

essentiellement d'un moteur de règles RETE-base avec un support natif pour raisonner sur les ontologies OWL, ontologies SWRL, et les règles RuleML. En plus, Bossam comprend plusieurs caractéristiques d'expressivité, notamment:

1. Les références d'URI en tant que symboles,
2. La syntaxe logique d'ordre 2,
3. Les disjonctions dans l'antécédent et conjonctions dans le conséquent,
4. L'attachement procédurale Java basé sur les URI,
5. Supporte à la fois la négation par l'échec, la négation classique.

On peut utiliser Bossam pour charger, inférer, et interroger un ensemble de documents. Ces documents peuvent être une combinaison des documents suivants :

- ✓ RDF(S) documents
- ✓ OWL documents
- ✓ Bossamrule documents
- ✓ SWRL (+OWL) documents

Aussi, du conséquent ou antécédent des règles, on peut appeler des objets Java, donc on aura un mélange de règles, ontologies et objets.

6. Travaux Connexes

Protégé-OWL a été complété par un éditeur SWRL. Il permet d'éditer à la fois des ontologies OWL et des règles SWRL. Une règle SWRL partage les mêmes prédicats unaires et binaires avec la contrepartie OWL. SWRL étend l'expressivité d'OWL mais la décidabilité des réponses n'est pas assurée. Le raisonnement sur des connaissances OWL enrichies par des règles SWRL est une question importante mais difficile à mettre en œuvre. Les supports de raisonnement pour les ontologies OWL enrichies par des règles SWRL présentent un défi pour le web sémantique. Ils sont en cours de construction et ils sont toujours une question ouverte.

Dans certaines applications, il est possible d'utiliser deux langages distincts avec des moteurs d'inférence séparés (un pour la partie structurelle (OWL-DL) et un autre pour la partie de règles (SWRL)). Mais, d'autres applications ont besoin de règles pour étendre l'expressivité d'OWL. Dans ce cas, le raisonnement sur les règles doit être en conjonction avec l'ontologie. Ce cas de système est plus complexe à cause de la décidabilité et de la complexité.

Plusieurs travaux sur le raisonnement sémantique sur les ontologies OWL enrichies par des règles SWRL peuvent être divisés en deux catégories:

6.1. Les approches homogènes réduisant les deux composantes en un langage logique

6.1.1 SWRL dans Jess [41]

Les assertions Jess sont définies comme des triplets (prédicat sujet objet). XSLT (Extended Style Sheet Language Transformation) est un langage permettant de spécifier des règles de transformation sur un document XML dans le but de générer un document ciblé dérivé du document XML. Etant donné un fichier SWRL, son fragment d'ontologie est transformé en faits sur le vocabulaire OWL (assertions Jess) au moyen de feuilles de style XSLT «OWL2JESS.XSL».

Tandis que son fragment de règles (règles SWRL) se transforme en règles Jess via une deuxième feuille de style XSLT «SWRL2JESS.XSL», en plus des règles prédéfinies servant de la traduction de la sémantique OWL. Le moteur Jess est lancé contre la base de règles résultante (faits Jess + règles Jess + règles prédéfinies) pour inférer le résultat. La figure 2.4 décrit les différentes transformations effectuées sur les fragments d'un fichier SWRL pour obtenir des assertions Jess.

Nous concluons que les règles SWRL dans Jess réduisent la partie OWL à des faits Jess, et ajoute ces faits en une base de règles prédéfinies du Jess, où tous les constructeurs OWL primitives sont en vedette que les règles d'inférence de manière à mettre en œuvre la sémantique OWL. Les règles SWRL sont transformées par un XSLT à des règles Jess. Le raisonnement SWRL est fourni par l'exécution du Moteur Jess sur la base de règle qui en résulte.

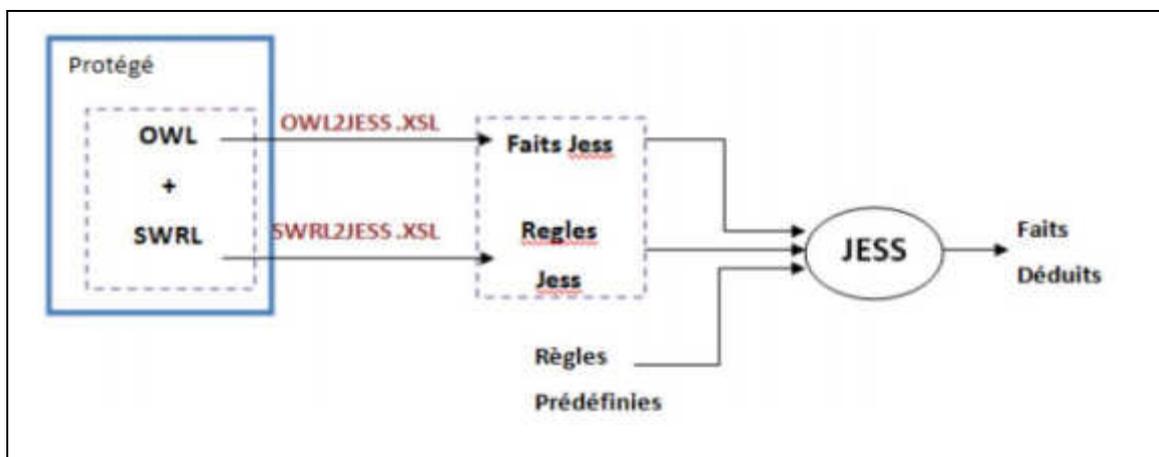


Figure 2.4. Modèle de SWRL dans Jess [41].

6.1.2. KAON2

KAON2 [42] est un raisonneur Java pour SHIQ étendu avec le fragment de DL-safe de SWRL. Il met en œuvre une procédure de décision basée sur la résolution des TBox(subsommation, satisfiabilité, classification) et Abox (recherche, réponse aux requêtes conjonctives). Il est livré avec sa propre interface basée sur Java et prend en charge l'interface DIG.

KAON2 est le successeur du projet KAON (souvent dénommé KAON1). La principale différence à KAON [42] est le langage d'ontologie en charge : KAON utilise une extension propriétaire de RDFS, alors que KAON2 est basé sur OWL-DL et F-Logic.

Caractéristiques

- Une API pour la gestion des programmes d'OWL-DL, SWRL, et les ontologies F-Logic,
- Un serveur autonome donnant accès à des ontologies de manière distribuée en utilisant RMI, Un moteur d'inférence pour répondre à des requêtes conjonctives (exprimée en utilisant la syntaxe SPARQL),
- Une interface DIG, permettant l'accès à partir d'outils tels que Protégé,
- Un module d'extraction des instances de l'ontologie bases de données relationnelles. Le raisonneur de KAON2 est fondé sur la réduction de la base de connaissance SHIQ(D) à un.

6.1.3 SWRL dans Sesame [43]

Sesame est un outil permettant de stocker des ontologies au format RDFS ainsi que des fichiers d'instances des ontologies au format RDF. Constitué de fichiers JSP (Java ServerPage) et de servlets. Sesame nécessite pour fonctionner l'installation de différents outils tels qu'un serveur d'applications, un Système de Gestion de Bases de Données (MySQL) et un driver JDBC permettant à Sesame de dialoguer avec la base de données. Le fichier SWRL représenté par une ontologie OWL étendue par des règles est décomposé en triplets RDF. Une base de données relationnelle appelée Sesame est utilisée pour stocker les triplets RDF des fichiers SWRL, tandis qu'un moteur Datalog est mis au point pour calculer le modèle de Herbrand de la base de connaissances SWRL.

En conséquence, la base de données relationnelle sésame est étendue par une base de données déductive.

6.2. Les approches hybrides utilisant des interfaces

SWRLJessTab [44] est un Plugin de Protégé permettant de calculer les inférences avec le classifieur Racer et le moteur de règle Jess. Afin de raisonner avec les règles et les ontologies, tous les deux doivent être représentés dans la même plateforme de Protégé. Cependant, son inconvénient est la perte d'information, du fait que l'interface des deux composants est incapable d'assurer la complétude du modèle.

6.3. Discussion

L'étude des travaux précédents montre que l'approche homogène suggère une transformation pratique des deux fragments d'un fichier SWRL (fragment d'ontologie+fragment de règles) en une base de règles d'un langage logique constituée des faits et des règles. Le résultat de cette transformation compliquée pose le problème de manque d'expressivité pour la base de règles générée puisque les constructeurs existentiels du langage OWL ne peuvent pas être tous traduits en faits d'un langage logique. En plus, lancer un seul moteur d'inférence contre la base de règles souffrant de tels problèmes engendre un service d'inférence incomplet.

Contrairement à l'approche homogène, l'approche hybride n'effectue pas des transformations explicites des deux fragments d'un fichier SWRL. Elle permet de rapprocher et d'interconnecter des raisonneurs stables et puissants. Elle utilise deux moteurs d'inférences séparés, le premier compile les axiomes de l'ontologie en une base de faits et le deuxième (moteur de règles tel que Jess) exploite ces faits pour déclencher les règles SWRL permettant enfin de trouver toutes les inférences possibles. En conséquence, c'est l'approche sur laquelle nous basons le système développé.

7. Conclusion

Dans ce chapitre, nous avons dressé l'intérêt de combiner des règles et des ontologies pour surmonter les insuffisances d'OWL. Cette combinaison d'OWL et de SWRL apporte une très haute expressivité mais au prix de la décidabilité. Plus le langage est expressif plus il est non décidable. En premier lieu, nous avons présenté les deux paradigmes de modélisation dans le web sémantique : le paradigme classique et le paradigme logique. Ensuite, nous avons montré les deux approches de combinaison d'OWL et les règles SWRL. L'approche Hybride qui se traduit en une séparation stricte entre les prédicats des règles et de l'ontologie, et l'approche homogène, traduite en l'intégration des deux formalismes en résultant un nouveau Langage.

Chapitre 2 Raisonnement sur les ontologies hybrides

Ensuite, nous avons exposé le langage des règles SWRL, et quelques supports d'inférence existants pour le raisonnement dans ontologies OWL enrichies par des règles SWRL. Aussi, nous avons exposé quelques travaux connexes qui combinent les deux composant OWL et SWRL, et raisonne dessus.

Dans le prochain chapitre de ce mémoire, nous présenterons l'architecture d'un système de raisonnement sur une ontologie OWL enrichie par des règles SWRL. Un processus détaillé pour la construction d'une ontologie OWL et son extension par les règles SWRL sera exposé.

Chapitre 3

Conception d'une
ontologie hybride dans
le domaine médical

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. Mais en réalité, obligés de naviguer dans une véritable confusion de pages, il est encore difficile de les obtenir de façon « satisfaisante ». La communauté médicale a alors des besoins réels et concrets d'utilisation d'un futur Web Sémantique. Les besoins majeurs attendus en médecine du Web Sémantique sont de pouvoir trouver facilement des informations médicales sur le Web, partager ces informations grâce au Web et ainsi pouvoir les exploiter pour l'aide à la décision.

Dans ce chapitre nous présentons notre solution au problème posé par ce mémoire, à savoir la conception et le raisonnement sur une ontologie hybride. Pour la modélisation de cette ontologie, nous avons choisi des langages très répandues : OWL étendu par des règles dans le langage SWRL (Semantic Web Rulelanguage).

L'architecture du système de raisonnement sémantique dans les ontologies hybride, repose sur le principe d'avoir deux modules d'inférences, s'exécutant en parallèle, en échangeant leurs conclusions : un classifieur pour raisonner sur l'ontologie OWL, et un moteur de règles, qui admet en entrée les règles SWRL.

Finalement nous donnerons la représentation sémantique des règles avec le langage SWRL qui va augmenter l'expressivité de notre ontologie, et permettre d'inférer de nouvelles connaissances.

Nous devons construire l'ontologie **GynécOntologie**. Une ontologie hybride dans le domaine médical, spécialement dans le service de gynécologie pour structurer les connaissances, les organiser, et plus précisément raisonner dessus. Pour y arriver, nous devons d'abord définir un processus de développement d'ontologie de domaine définie par le langage OWL. Pour ce faire, nous nous sommes basés sur le travail de [45].

2. Objectif du travail

Notre travail consiste principalement à la création d'une ontologie OWL dans un domaine médical spécialisé en gynécologie et l'enrichir avec des règles SWRL pour obtenir une

plus haute expressivité relationnelle, faire des raisonnements et produire de nouveaux faits à partir des inférences qui sont faites par les moteurs d'inférences.

Pour atteindre cet objectif, il faut mettre en œuvre les sous objectifs suivants :

- Suivre une démarche précise qui repose sur une architecture permettant de construire un système de raisonnement et d'inférence.
- Conception de l'ontologie dans le service de gynécologie, et décrire le domaine par les concepts et relations.
- rédiger les règles SWRL qui vont pouvoir ajouter de nouvelles assertions à l'ontologie et les exécuterons en utilisant un moteur de règles, ce qui nous permettra de déduire de nouvelles connaissances dans le domaine dédié.

3. Architecture du système de raisonnement sémantique

Le raisonnement sur une ontologie hybride OWL enrichie par des règles SWRL exige à la fois un raisonnement sur le langage OWL et sur les règles SWRL, ce qui mène soit à l'utilisation d'un raisonneur unique qui prend en charge les deux langages (l'approche homogène) ou de s'appuyer sur deux moteurs d'inférence séparés, le premier moteur compile les axiomes de l'ontologie et le deuxième exploite ses faits pour déclencher les règles SWRL (l'approche hybride).

Dans le cadre de notre travail, nous nous reposons sur la deuxième solution en utilisant un moteur d'inférence sur OWL (pellet) et un moteur de règles sur SWRL (Jess).

3.1. Description de l'architecture

L'architecture du système proposé est constituée des quatre composants suivants :

1) L'ontologie hybride, composée des fragments suivants :

- ✓ Base terminologique fournissant les définitions logiques des concepts (classes), des relations (propriétés), et des axiomes.
- ✓ Base assertionnelle contenant les individus (instances de classes) et les instances des relations qui les relient.
- ✓ Base de règles fournissant des règles étendant l'ontologie pour augmenter son expressivité.

- 2) Le raisonneur, tel que Racer ou Pellet pour effectuer des raisonnements sur la partie OWL de l'ontologie hybride.
- 3) Le moteur de règles, tel que Jess ou Bossam pour raisonner sur les règles modélisées avec SWRL.
- 4) un module qui prend en entrée un fichier d'une extension (.OWL) qui est généré directement à partir de Protégé, qui permet d'afficher les informations de notre ontologie médicale.

L'architecture de ce système de raisonnement est montrée par la figure 3.1.

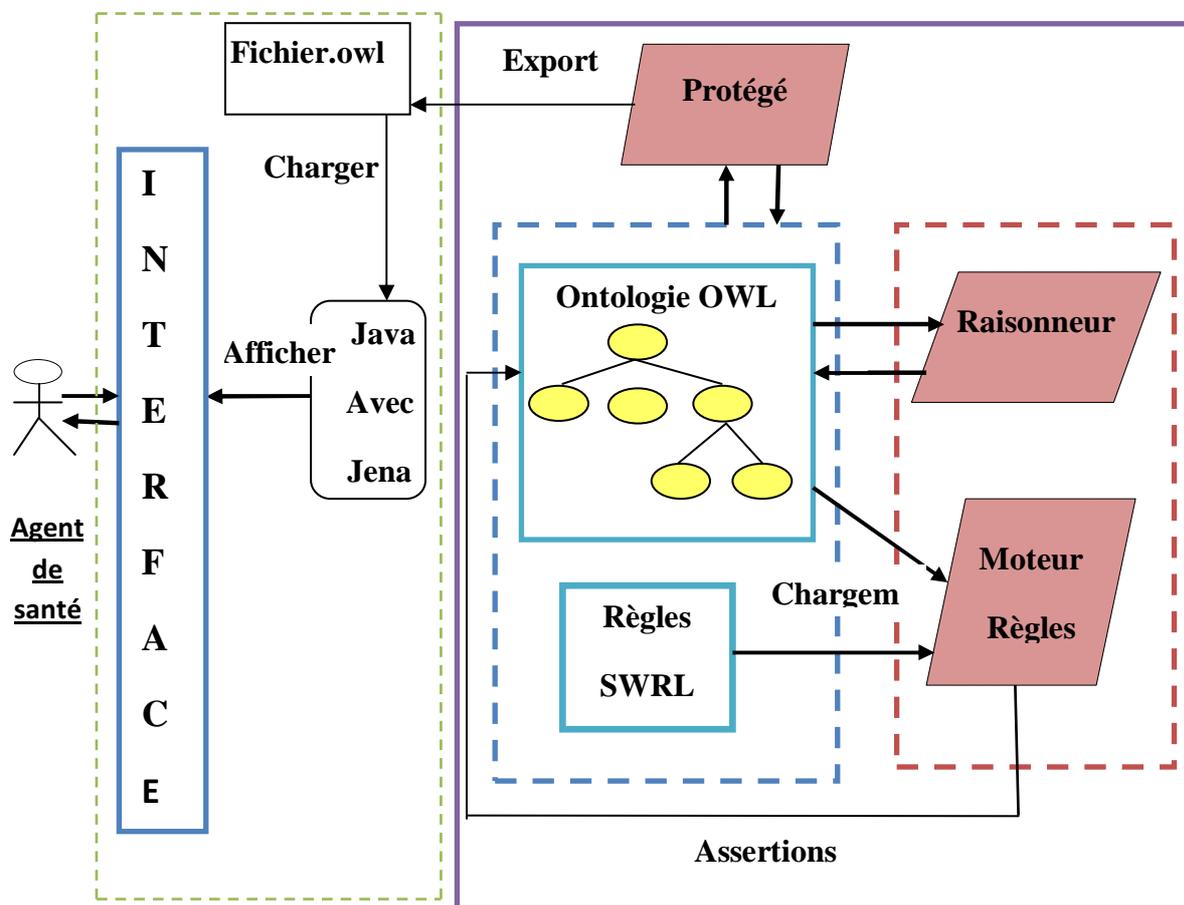


Figure 3.1. Architecture du système de raisonnement [46].

3.2. Fonctionnement du système de raisonnement

Le raisonnement sur une ontologie hybride enrichie par les règles SWRL s'effectue suivant ces étapes :

- (1) Chargement de l'ontologie OWL au niveau du raisonneur (Pellet).

- (2) Exécution du raisonneur, les inférences de base sont établies, et des résultats sont générés.
- (3) Mise à jour de la base assertionnelle de l'ontologie par les résultats obtenus dans (2).
- (4) Chargement des règles SWRL au niveau du moteur de règles (Jess), ces règles se placent dans la base de règles du moteur de règles.
- (5) Chargement des fragments d'OWL de l'ontologie, et leur placement dans la base de faits du moteur d'inférence.
- (6) Exécution du moteur de règles, et génération de nouveaux faits.
- (7) Mise à jour de l'ontologie par les nouvelles assertions obtenues dans (6).

La figure 3.2 suivante explique le fonctionnement du système de raisonnement :

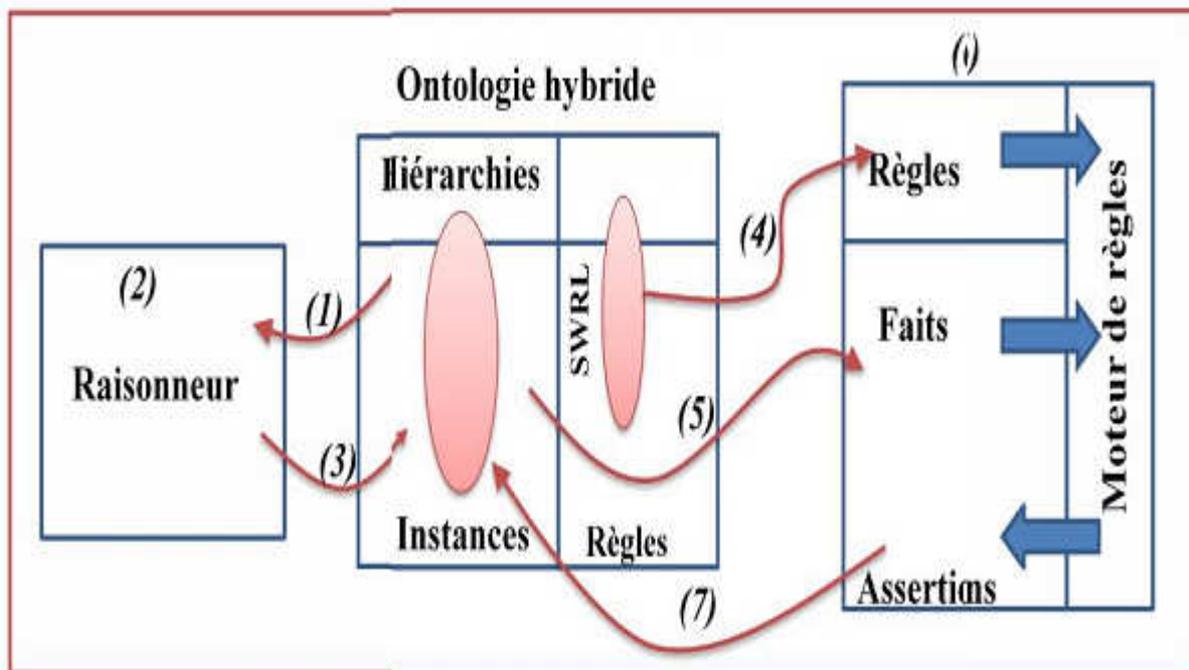


Figure 3.2. Fonctionnement du système de raisonnement [46].

4. Processus 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 s'appuyer 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 [45] :

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

4.1. 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:

- 1) Le domaine de connaissance : déterminer aussi précisément que possible le domaine que va couvrir l'ontologie.
- 2) L'objectif : le but de l'ontologie à créer pour le domaine considéré.
- 3) Les utilisateurs : identifier au maximum les futurs utilisateurs de l'ontologie à créer.
- 4) Les sources d'informations : déterminer les sources d'informations d'où les connaissances seront obtenues, par exemple, les experts du domaine, les documents techniques, etc., sont des sources de connaissance.
- 5) La portée de l'ontologie : déterminer 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.

4.2. 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 :

- 1) 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.
- 2) Classifier les concepts dans des 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.
- 3) Représenter les relations qui existent entre les différents concepts par un diagramme de relations binaires : la relation binaire lie un concept source à un concept cible, qu'ils appartiennent ou non à la même hiérarchie.
- 4) Identifier les concepts par leurs instances, leurs attributs ainsi que leurs concepts synonymes dans un dictionnaire de concepts (DC).
5. Décrire les relations dans une 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.
- 6) Spécifier des contraintes sur les attributs dans une 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.
- 7) Spécifier des axiomes sur les concepts dans une 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.
- 8) Décrire les instances des concepts dans une 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.

4.3. 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 (TOP)
\perp	Le concept le plus spécifique (Bottom)
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

Tableau 3.1. Syntaxe du langage SHIQ [47].

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

4.4. 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.

4.5. 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 Racer.

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.

5. Construction de l'ontologie hybride GynécOntologie dans le domaine médical

Dans cette section, nous construisons notre ontologie médicale qui concerne le service gynécologie. 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 gynécologie.

5.1. La gynécologie : domaine d'application

La gynécologie est une spécialité médico-chirurgicale qui s'occupe de la physiologie et des affections du système génital de la femme, la grossesse et l'accouchement. Il existe de très nombreuses sous-spécialités parmi lesquelles, la procréation médicalement assistée et l'échographie. Le médecin spécialisé pratiquant la gynécologie s'appelle un gynécologue.

Chapitre 3 : Conception d'une ontologie hybride dans le domaine médical

Le gynécologue accompagne la femme de la puberté à la ménopause, prescrit en accord avec sa patiente les moyens de contraception, accompagne la grossesse, décide de mettre en place des procédures d'aide à la procréation, soulage les symptômes de la ménopause, etc.

Le volume considérable d'informations transitant dans le service gynécologie, qui nécessite une structuration des informations et de raisonnement, ainsi que l'importance de cette branche dans la vie de la femme, en matière de sa santé, sont parmi les principales raisons de choisir cette spécialité comme domaine d'application. Une ontologie dans ce cadre donc va être largement utilisée, et partagée.

Le service Gynécologie est séparé en deux sous branches:

- L'obstétrique concerne la prise en charge de la grossesse, de l'accouchement et des suites de couches.
- La gynécologie concerne la prise en charge des pathologies de l'appareil de reproduction (utérus, ovaires, etc.) et du sein chez la femme.

➤ **Motif de consultation en Gynéco-obstétrique**

Une patiente subit un examen gynéco-obstétrique dans trois cas:

(1): A la présence d'une pathologie, il s'agit donc d'une consultation dans le cadre d'une démarche diagnostique.

(2): Pas de pathologie gynécologique, mais pour un examen gynécologique systématique, il s'agit alors d'une démarche de dépistage.

(3): Le suivi des grossesses.

➤ **Une Consultation Gynéco-obstétrique**

L'examen gynéco-obstétrique obéit à des règles simples qui doivent être toujours appliquées, pour élaborer un diagnostic et remettre un traitement si nécessaire. L'examen se déroule comme suit, figure 3.3:

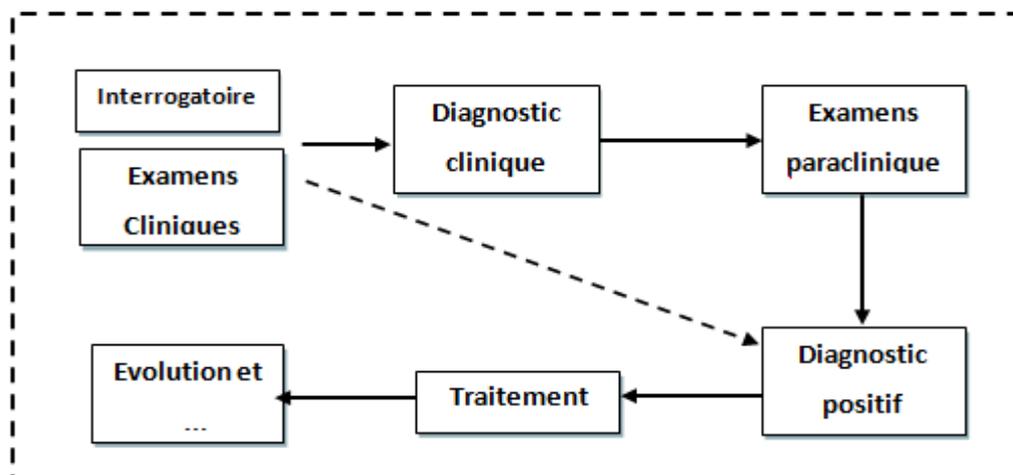


Figure 3.3. Démarche d'une consultation médicale.

- **Interrogatoire** : Il s'agit de recueillir des informations de la patiente. L'interrogatoire a trois objectifs principaux :
 - (1) Déterminer le ou les motifs de la consultation,
 - (2) Répertorier les antécédents personnels et familiaux de la patiente,
 - (3) Faire décrire la symptomatologie fonctionnelle de la patiente.
- **Examen clinique** : permet au médecin ou au clinicien (infirmière, infirmière clinicienne, infirmière praticienne) d'aboutir à un diagnostic. C'est un examen physique à la recherche de signes physiques.
- **Diagnostic clinique** : Le diagnostic clinique apporte une aide précieuse au médecin pour le diagnostic et le suivi de nombreuses pathologies et maladies.

Examen para clinique : Après l'examen clinique, les examens complémentaires permettent d'orienter le diagnostic. Ils ne doivent être demandés qu'après avoir fait un examen clinique minutieux pour confirmer le diagnostic clinique. En fonction de l'hypothèse la plus probable, on doit procéder par des examens : biologiques, radiologiques, autres.

- **Diagnostic positif** : Ce sont des argumentations cliniques et para cliniques permettant d'éliminer les hypothèses incertaines et de confirmer un diagnostic positif.
- **Traitement** : est l'acte de soigner d'un problème de santé.
- **Evolution et surveillance** : la surveillance après le traitement a pour but de déceler les rechutes afin que le traitement ait le plus de chance possible d'obtenir un résultat significatif.

5.2. 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 gynécologie de la médecine.
- ✓ **Les utilisateurs** : médecins et agents de santé du service gynécologie.
- ✓ **Les sources d'informations** : nous nous sommes basés sur le travail [46] pour construire notre ontologie, mais aussi d'après des experts du domaine,
- ✓ **La portée de l'ontologie (liste des termes importants)** : parmi ces termes les plus importants, nous citons : Patiente, Consultation, Maladie, Traitement, Grossesse, Accouchement, Structure santé, Forceps, Echographie ...etc.

5.3. 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 :

➤ Construction du glossaire des termes

Ce glossaire recueille et décrit tous les termes qui sont utiles et potentiellement utilisables dans l'ontologie finale. Le tableau ci-dessous (Tableau 3.2) fournit une liste détaillée des différents termes utilisés dans l'ontologie :

Thème	Description
Personne	Est un humain qui fait partie de la communauté du service gynécologie.
Agent-santé	Est une personne affiliée dans le domaine de la santé. Il peut être : Un docteur, infirmier, clinicien,
Patiente	Est une personne de sexe féminin, qui est caractérisé par : l'âge, le cycle, la parité, la DDR, menstrues,
Fœtus	Est un humain de la fin du 2 ^{ème} mois au terme de la grossesse.
Nouveau-né	Une personne du sexe masculin ou féminin issu d'un accouchement d'une femme.
Grossesse	Un état d'une femme qui est enceinte.

Chapitre 3 : Conception d'une ontologie hybride dans le domaine médical

Thème	Description
Accouchement	C'est l'action de mettre au monde un nouveau-né par une femme enceinte.
Avortement	Interruption prématurée de la grossesse.
Surveillance	Observer une patiente (ou un fœtus) pour la contrôler si elle a une maladie, ou après un traitement, ou une GHR.
Maladie	C'est l'altération de la santé d'un être vivant. C'est une irrégularité par rapport à la norme, découverte par une consultation.
Traitement	Manière de soigner un patient qui a une ou plusieurs anomalies.
Consultation	L'examen de la patiente par un médecin pour découvrir une ou plusieurs anomalies.
Antécédent Obstétricaux	Savoir sur le nombre de grossesses de la patiente, gestité, dates des accouchements, pathologies des grossesses, des accouchements et des suites de couche.
GHR	C'est une grossesse où la femme est atteinte de diabète, ou HTA, ou cardiopathie, qui a une grossesse multiple, est à un âge avancé.
Césarienne	Incision pratiqué dans la paroi abdominale pour extraire le fœtus
Structure-santé	Etablissement où une patiente se rend pour un motif obstétrique ou gynécologique. Ça peut être : un hôpital, une clinique, un bureau de santé.
Biopsie	Prélèvement d'un fragment de tissu ou organe à des fins d'examen microscopique.
Frottis	Prélèvement et étalement sur une lame, en couche mince, de cellules d'un tissu ou un organe en vue d'une observation microscopique.
Sage-femme	Personne ayant pour rôle la surveillance, les soins et le conseil des femmes pendant l'accouchement et dans les suites de couches.
Clinique	Etablissement hospitalier privé ou étatique. Peut-être aussi un centre de radiologie, un laboratoire.
Technique	Méthode employée par un agent de santé lors d'un examen clinique ou para-clinique, un accouchement, ou traitement. Elle peut être

Chapitre 3 : Conception d'une ontologie hybride dans le domaine médical

Thème	Description
	instrumentale ou non-instrumentale
Préventif	Un type de traitement cherche à empêcher l'apparition d'une maladie (Exemple : par une vaccination), ou à supprimer un facteur de risque (tabagisme)
Signes-fcls	C'est des symptômes et des signes qui indiquent une perturbation de l'état d'une patiente. Ils peuvent être subjectifs ou objectifs.
Auscultation	Action d'écouter les bruits internes de l'organisme pour contrôler le fonctionnement d'un organe ou déceler une anomalie.
Hystérocopie	Imagerie permettant de voir la cavité utérine et de réaliser des biopsies dirigées, utile pour le diagnostic de cancer de l'utérus.
Anapath	L'anatomo-pathologie étudie les lésions macroscopiques et microscopiques de tissus prélevés sur des êtres vivants malades par biopsie, frottis ou biopsie extemporanée.
Forceps	Dans ce type d'accouchement, on utilise un forceps pour saisir et protéger la tête du fœtus afin de faciliter son expulsion.
Cryothérapie	Traitement utilisant le froid sous forme de glace, sachets chimique congelés, ou de gaz.
Palpation	Méthode d'examen utilisant les mains et les doigts pour recueillir les informations utiles par le toucher.
Percussion	Méthode d'examen d'organes ou de cavités internes reposant sur l'appréciation de la sonorité ou de la résonance produite par le tapotement de l'extrémité des doigts sur la peau de la région étudiée.
Hémoculture	Technique de laboratoire visant à mettre en évidence la présence ou l'absence de microorganismes pathogènes dans le sang.
F. risque	C'est des éléments qui augmentent le risque d'atteindre une maladie.
E .Speculum	Examen qui s'effectue à l'aide d'un instrument permettant de maintenir large et d'éclairer un conduit ou une cavité du corps.
Episio	Consiste à sectionner la muqueuse vaginale et les muscles afin d'agrandir l'orifice de la vulve et de faciliter l'expulsion du fœtus lors de

Chapitre 3 : Conception d'une ontologie hybride dans le domaine médical

Thème	Description
	l'accouchement.
Hôpital	Etablissement où l'on soigne les malades.
Bureau-santé	Un bureau de santé est un organisme de santé publique établi par un groupe de municipalités urbaines et rurales pour fournir des services de santé publique
Interrogatoire	L'interrogatoire médical, c'est le moment où l'on interroge le patient au début de la consultation pour recueillir certaines informations.
Examen-clinique	L'examen physique, ou examen clinique, fait partie de l'examen médical, qui permet au médecin ou au clinicien (infirmière, infirmière clinicienne, infirmière praticienne) d'aboutir à un diagnostic, ou à une impression clinique.
Examen-para-clinique	Examen destiné à confirmer un diagnostic fait par observation. C'est des examens biologiques, instrumentaux ou d'imagerie
V. basse	Un accouchement par voie basse signifie que le bébé descend par l'utérus, le col et enfin le vagin
Cardiaque	La maladie du cœur
Hématologique	l'hématologie se consacre à l'étude du sang mais également à celle de la moelle osseuse et des ganglions lymphatiques de façon générale.
Médecin	Un médecin est un professionnel de la santé titulaire d'un diplôme de docteur en médecine. Il est chargé de soigner les maladies, pathologies
TSS	Agent de santé qui peut être un radiologue, un biologiste, ...
Infirmière	L'infirmier soigne les malades et veille à leur bien-être
ATCD	C'est l'interroger la patiente en détail sur son passé.
CDD	l'histoire du malade (Profession, Toxicomanie, alcoolisme, tabagisme, Habitudes alimentaires, ... etc.)
...	...

Tableau 3.2. Glossaire des termes.

➤ Construction du diagramme de relations binaires et de classification de concepts

Dans cette étape, nous construisons le diagramme de relations binaires et de classification de concepts en deux étapes principales. Initialement, nous répertorions les concepts en ensembles organisés, ensuite nous relierons les concepts entre eux, si nécessaire, par des relations. La hiérarchie de classification de concepts démontre l'organisation des concepts de l'ontologie en un ordre hiérarchique qui exprime les relations sous classe – super classe. En utilisant la relation « Sous classe de » entre les classes pour définir leurs classifications, la classe C1 est une sous classe de la classe C2 si et seulement si toute instance de la classe C1 est une instance de la classe C2, par exemple la classe Interrogatoire est une sous classe de la classe Consultation. Par exemple nous pouvons commencer en créant des classes pour les concepts généraux : **personne, Maladie, structure-sanitaire, traitement, grossesse, accouchement...** Les figures 3.4 et 3.5 montrent les hiérarchies des concepts généraux mentionnés ci-dessus et leurs classifications.

Chapitre 3 : Conception d'une ontologie hybride dans le domaine médical

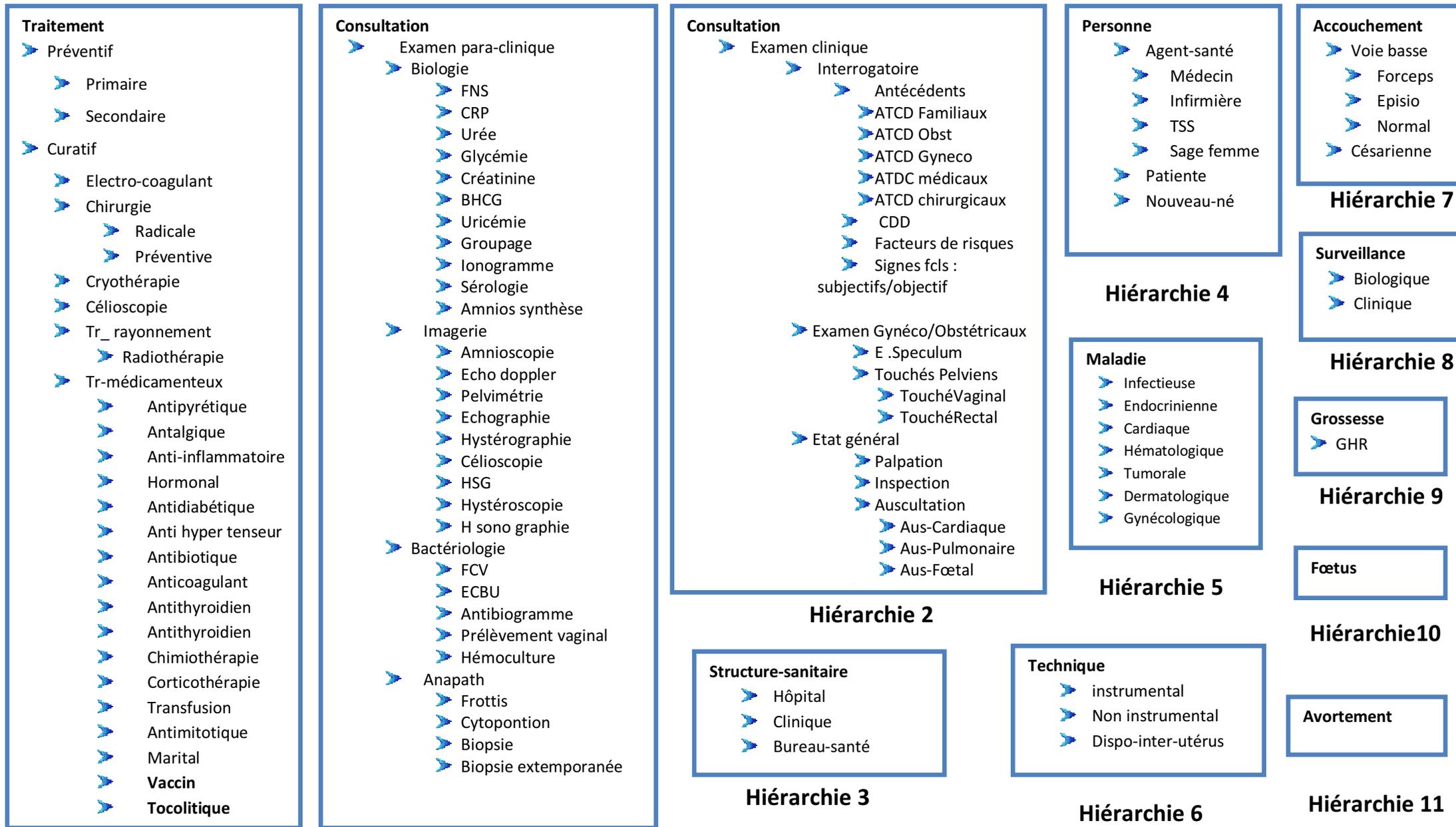


Figure 1.4. Hiérarchies de concepts.

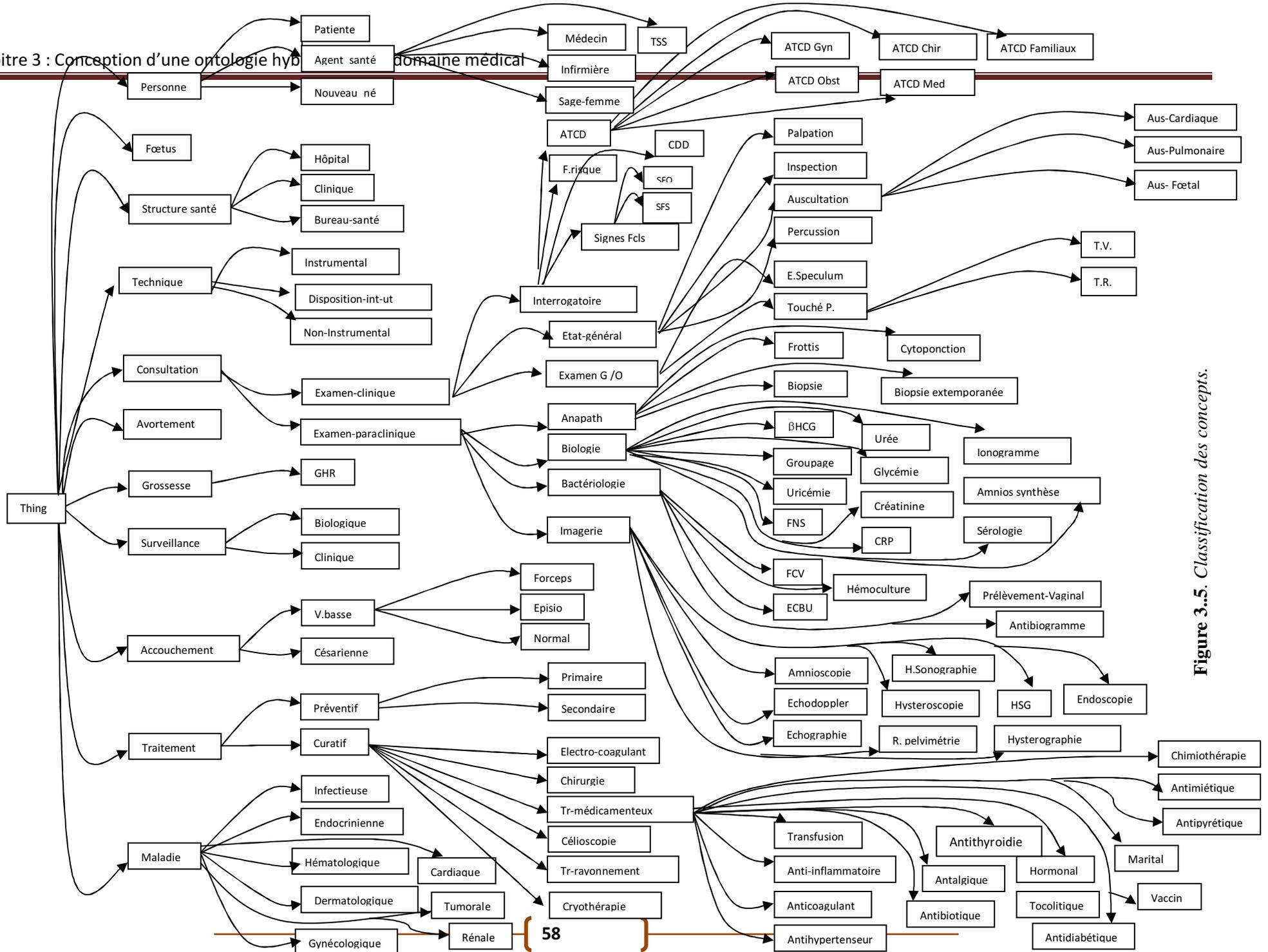


Figure 3..5. Classification des concepts.

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 3.6) représente le diagramme de relations binaires.

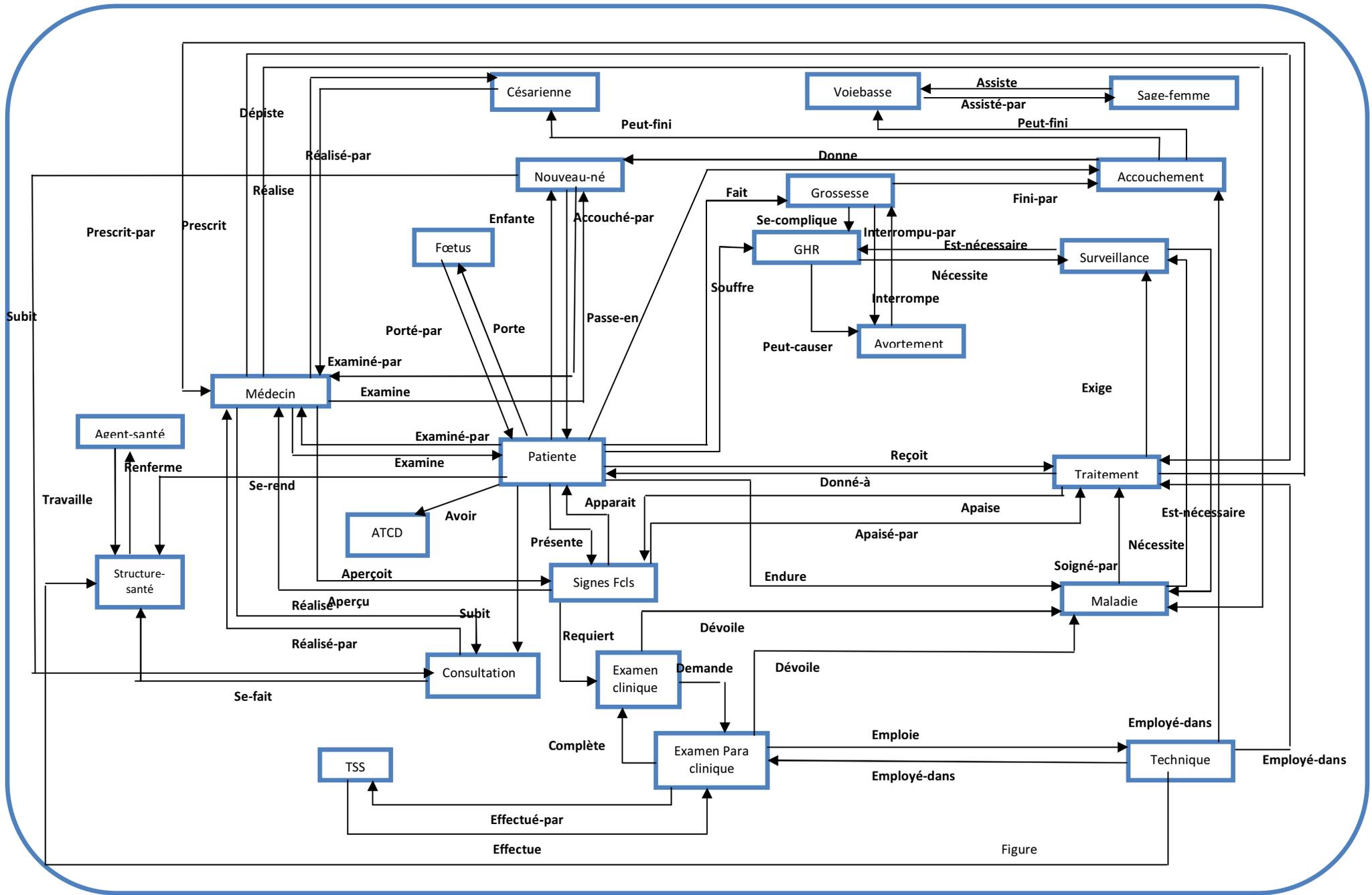


Figure 3.6. Le diagramme des relations binaires.

➤ Dictionnaire de concepts

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 3.3 représente le dictionnaire des concepts pour l'ontologie définie par la hiérarchie précédente.

Nom de concept	Concepts synonymes	Attributs	Instances
Personne	Humain	-	-
Agent_santé	Clinicien, thérapeute	Nom-agent-santé Prénom-agent-santé	-
Patiente	Femme, malade.	Nom-JF Nom-patiente Prénom-patiente Tel-patiente Adresse-patiente Age-patiente DDR Pariété Réglé Cycle	Baouta Samira Boufandassa Rima Loussif Amina Triha Amal
Fœtus	Embryon, germe, œuf.	Age-embry Perim-cranien Activ-cardiaque MAF	Fœtus-1 Fœtus-2
Nouveau-né	Enfant	Date de naissance Poids Heure_naiss Perim-cranien_Nouveau-né	
Grossesse	Terme	Age-grossesse Position- fœtus	Grossesse-1 Grossesse-2

Chapitre 3 : Conception d'une ontologie hybride dans le domaine médical

Nom de concept	Concepts synonymes	Attributs	Instances
		Nbr-fœtus Etat-grossesse	Grossesse-3
Accouchement	Naissance, enfantement.	Date-accouch Délivrance Type-voie	-
Avortement	Echouement	Age-avortement	-
Surveillance	Contrôle, observation	-	-
Maladie	Anomalie, syndrome	Nom-maladie Pronostic Type-maladie Causes Debut-maladie	-
Traitement	Cure, remède, soins	Durée-Traitement	-
Consultation	Examen, visite, réception	Date-consultation	-
Antécédent Obstétricaux	-	Descrip-ATCD-obs	-
GHR	Grossesse a hauts risque	Nom-GHR Type-GHR	GHR_1 GHR_2 GHR_3
Césarienne	Incision, chirurgie	-	-
Structure-santé	Etablissement de santé	Dénomination Adresse-Structure-santé Tel Fax	-
Biopsie	Prélèvement de tissu	descrip-biopsie	Biopsie_1 Biopsie_2

Chapitre 3 : Conception d'une ontologie hybride dans le domaine médical

Nom de concept	Concepts synonymes	Attributs	Instances
Frottis	Prélèvement	Résultat-frottis	Frottis_1
Sage-femme	-	Grade-SF	ZahiaBenhecine
Clinique	Centre hospitalier.	Spécialité Type-clinique	EHS Sidi Mabrouk Imagerie Namous Ibn sina
Technique	Méthode	Descrip-tech	-
Préventif	-	Descrip-prev	-
Signes-fcls	-Symptômes	Descrip-SF	-
Anapath	-	Date-anapath	-
Cryothérapie	-	Date-cryo Result-cryo	-
Palpation	Toucher, tact, contact	Zone-palpée Défense Contracture Présence-Masse Souplesse Douleur	Palpation_seins
Percussion	Tamponnement	Matité Tympanisme	-
Hémoculture	-	Resulta-hemo	-
F.risque	Elément portant au danger	Descrip-frisque	Tabagisme Obésité Diabète
E .Speculum	-	Descrip-speculum	-

Chapitre 3 : Conception d'une ontologie hybride dans le domaine médical

Nom de concept	Concepts synonymes	Attributs	Instances
Episiotomie	-	-	-
Hôpital	Etablissement hospitalier,	Type-hopital	CHU Benbadis Ain naâdja El Bir
Bureau-santé	Siège-santé	Type-Bureau	-
Interrogatoire	Questionnaire	-Descrip-int	-
Examen-clinique	Examen direct, physique	Diagnostic	-
Examen-paraclinique	Examen complémentaire	-	-
S.biologique	-	Observance-bio	-
S.clinique	-	Observance-cli	-
V.basse	-	Dil-col Poche_rempue	-
Curatif	-	Debut-trait	-
Infectieuse	-	Porte-entrée Germe	Salpingite Infection urinaire
Endocrinienne	-	Cause-endocrin	-
Hématologique	-	-	-
Gynécologique	-	-	-
Tumorale	Cancéreuse	Type-tumeur Classification	Cancer du col Cancer endomètre Dysplasie
...

Tableau 3.3. Dictionnaire des concepts.

➤ 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 [48].

Pour chaque relation dont la source est dans l'arbre de classification de concepts, 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 3.4 illustre la spécification des relations binaires entre les différentes hiérarchies pour notre ontologie.

Nom de relation	Concept source	Cardinalité source	Concept cible	Cardinalité cible	Relation inverse
Donne	Accouchement	(0, n)	Nouveau-né	(1, 1)	-
Assiste	Sage-femme	(1, n)	Voie basse	(1, n)	Assisté-par
Fait	Patiente	(0, n)	Grossesse	(1, 1)	-
Fini-par	Grossesse	(0, 1)	Accouchement	(1, 1)	-
Se-complice	Grossesse	(0, n)	GHR	(1, 1)	-
Enfante	Patiente	(1, n)	Nouveau-né	(1, 1)	Accouché-par
Porté-par	Fœtus	(1, 1)	Patiente	(1, n)	Porte
Présente	Patiente	(0, n)	Signes-fcls	(0, n)	Apparaît
Dépiste	Médecin	(0, n)	Maladie	(1, n)	-
Se-rend	Patiente	(0, n)	Structure-santé	(1, n)	-
Réalise	Médecin	(1, n)	Consultation	(1, n)	Réalisé-par
Travaille	Agent-santé	(1, n)	Structure-santé	(1, n)	Renferme
Endure	Patiente	(0, n)	Maladie	(1, n)	-
Reçoit	Patiente	(0, n)	Traitement	(0, n)	Donné-à
Apaise	Traitement	(0, n)	Signes-fcls	(0, n)	Apaisé-par
Exige	Traitement	(0, n)	Surveillance	(0, n)	-
Figure	Technique	(0, n)	Structure-santé	(0, n)	-
Nécessite	Maladie	(0, n)	Surveillance	(1, n)	Est-nécessaire
Employé-dans	Technique	(0, n)	Examen para clinique	(0, n)	Emploie

Chapitre 3 : Conception d'une ontologie hybride dans le domaine médical

Nom de relation	Concept source	Cardinalité source	Concept cible	Cardinalité cible	Relation inverse
Se-fait	Consultation	(1, n)	Structure-santé	(1, n)	-
Renferme	Structure-santé	(1, n)	Agent-santé	(1, n)	Travaille
Examine	Médecin	(0, n)	Patiente	(0, n)	Examiné-par
Est-nécessaire	Surveillance	(0, n)	GHR	(0, n)	-
Dévoile	Examen clinique Examen para clinique	(0, n)	Maladie	(0, n)	-
Porte	Patiente	(0, n)	Fœtus	(1, 1)	Porté-par
Prescrit	Médecin	(0, n)	Traitement	(1, 1)	Prescrit-par
Complete	Examen para clinique	(1, 1)	Examen clinique	(1, n)	Demande
Soigné-par	Maladie	(1, n)	Traitement	(1, n)	-
Souffre	Patiente	(0, 1)	GHR	(0, n)	-
Subit	Patiente	(0, n)	Consultation	(0, n)	-
Peut-causer	GHR	(0, 1)	Avortement	(0, 1)	-
Demande	Examen clinique	(0, n)	Examen para clinique	(1, n)	Complète
Requiert	Signes fcls	(1, n)	Examen clinique	(0, n)	-
Interrompu-par	Grossesse	(0, 1)	Avortement	(0, 1)	Interrompte
Prescrit-par	Traitement	(1, 1)	Médecin	(1, n)	Prescrit
Accouché-par	Nouveau-né	(1, 1)	Patiente	(1, 1)	Enfante
Passe-en	Patiente	(0, n)	Accouchement	(0, n)	-
Interrompte	Avortement	(0, 1)	Grossesse	(0, 1)	Interrompu-par
Apaisé-par	Signes-Fcls	(1, n)	Traitement	(1, n)	Apaise
Donné-a	Traitement	(0, n)	Patiente	(0, n)	Reçoit
Assisté-par	Voie basse	(1, n)	Sage-femme	(0, n)	Assiste

Nom de relation	Concept source	Cardinalité source	Concept cible	Cardinalité cible	Relation inverse
Réalisé-par	Consultation	(1, n)	Médecin	(1, n)	Réalise
Emploie	Examen para clinique	(0, n)	Technique	(0, n)	Employé-dans
Avoir	Patiente	(0, n)	ATCD	(0, n)	-
Aperçoit	Médecin	(0, n)	Signes-Fcls	(1, n)	Aperçu
Examiné-par	Patiente	(1, n)	Médecin	(1, n)	Examine
Aperçu	Signes-Fcls	(1, n)	Médecin	(0, n)	Aperçoit
...

Tableau 3.4. Table des relations binaires.

➤ Tableau des attributs

Les attributs sont des propriétés qui prennent leurs valeurs dans les types prédéfinis (String, Integer, Boolean, Date...). Par exemple le concept femme a comme attributs : Nom, prénom, âge, DDR et parité.

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 3.5 spécifie ces informations pour chaque attribut.

Nom de l'attribut	Type	Cardinalité (min/max)	Valeur par défaut	Domaine des valeurs
Nom-agent-santé	String	(1,1)	-	-
Prénom-agent-santé	String	(1,1)	-	-
Adresse-structure-santé	String	(1,1)	-	-
Dénomination	String	(1,1)	-	-
Dil-col	Integer	(1,1)	-	[0..9]
Grade-med	String	(1,1)	-	{professeur, résident, maitre-assistant, généraliste}
Spécialité-TSS	String	(1,1)	-	{radiologue, biologiste,

Nom de l'attribut	Type	Cardinalité (min/max)	Valeur par défaut	Domaine des valeurs
				anapath}
Age-patiente	Integer	(1,1)	-	[6..80]
DDR	Date	(1,1)	-	-
Pariété	Integer	(0, 1)	-	-
Tel-structure-santé	String	(1, n)	-	-
Nom-JF	String	(1,1)	-	-
Age-embry	Integer	(1,1)	-	[1..42]
Date-accouch	Date	(1,1)	-	-
Position-fœtus	String	(1,1)	-	{Sommet, siège, face, franc, transversal}
Type-GHR	String	(1, n)	-	{HTA, Diabète, anémie, cardiopathie}
Age-grossesse	Integer	(1,1)	-	[1..42]
Mode-admi	String	(1,1)	-	{oral, parentéral, cutané, transmuqueuse, rectal}
Posologie	Integer	(1,1)	-	-
Effet-secondaire	String	(1, n)	-	-
Dosage	Integer	(1, n)	-	-
Forme	String	(1, n)	-	{Comprimé, gélule, suppositoire, pommade, crème, sirop, injectable, poudre, spray}
Type-hôpital	String	(1, n)	-	{Général, Spécialisé, militaire, universitaire}
Type-clinique	String	(1, n)	-	{privé, étatique, radiologie, imagerie, soins, ...}
Descrip-ATCD	String	(1, n)	-	-
Fonction-outil	String	(1, n)	-	-
Technique-outil	String	(1,1)	-	-

Nom de l'attribut	Type	Cardinalité (min/max)	Valeur par défaut	Domaine des valeurs
Pronostic	String	(1,1)	-	{Bon, Mauvais, Sombre}
ABO	String	(1,1)	-	{A, B, O, AB}
Rhésus	String	(1,1)	-	{Positif, Négatif}
Long-col	String	(1,1)	-	[1..9]
Position-col	String	(1,1)	-	{Postérieur, Antérieur, Intermédiaire}
Odeur	String	(1,1)	-	{Fétide, Normal}
Leucorrhée	Boolean	(1,1)	-	-
Saignement	Boolean	(1,1)	-	-
Température	Integer	(1,1)	-	[34..41]
Respiration	String	(1,1)	-	{Apnée, bradypnée, polypnée, tachypnée, pause-respiratoire}
Pouls	String	(1,1)	-	{Long, Normal, Rapide}
Pilosité	String	(1,1)	-	{Hypertrichose, Normal}
Cycle	String	(1,1)	-	{Long, court, normal}
Perim-crânien	Integer	(1,1)	-	-
Activ-cardiaque	Boolean	(1,1)	-	-
MAF	Boolean	(1,1)	-	-
Date de naissance	String	(1,1)	-	-
Poids	Integer	(1,1)	-	-
Heure_naiss	String	(1,1)	-	-
Nbr-fœtus	Integer	(0,1)	-	-
Délivrance	String	(1,1)	-	{Normal, pathologique}
Réglé	Boolean	(1,1)	-	-
...

Tableau 3.5. Table des attributs.

➤ **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

Chapitre 3 : Conception d'une ontologie hybride dans le domaine médical

l'axiome et l'expression logique. Pour notre ontologie nous spécifions quelques axiomes comme il est représenté dans le Tableau 3.6.

Nom du concept	Description	Expression logique
Personne	Chaque personne du service gynécologie est soit une patiente, un agent de santé, nouveau-né.	$\forall (X), \text{Personne}(X) \Rightarrow \text{Patiente}(X) \vee \text{Agent-Santé}(X) \vee \text{Nouveau-né}(X)$
Patiente	Une patiente doit au moins se rendre dans une structure de santé, et être examiné par un médecin.	$\forall (X), \text{Patiente}(X) \Rightarrow \exists Y \text{Structure-Santé}(Y) \wedge \text{Se-Rend}(X, Y) \wedge \exists Z \text{Médecin}(Z) \wedge \text{Examiné-par}(Z, X)$
Médecin	Un médecin travaille dans une structure de santé, examine des patients, prescrit des traitements et dépiste des maladies.	$\forall (X), \text{Médecin}(X) \Rightarrow \exists Y \text{Structure-Santé}(Y) \wedge \text{Travaille}(X, Y) \wedge \exists Z \text{Traitement}(Z) \wedge \text{Prescrit}(X, Z) \wedge \exists W \text{Maladie}(W) \wedge \text{Dépiste}(X, W)$
Grossesse	Etat d'une patiente qui porte un fœtus.	$\forall (X), \text{Grossesse}(X) \Rightarrow \exists Y \text{Patiente}(Y) \wedge \text{Fait}(Y, X) \wedge \exists Z \text{Fœtus}(Z) \wedge \text{Porte}(Y, Z)$
Technique	Peut-être instrumental ou non instrumental	$\forall (X), \text{Technique}(X) \Rightarrow \text{Instrumental}(X) \vee \text{Non-Instrumental}(X)$
Examen clinique	Dans le service gynécologie, un examen clinique est un examen physique de l'état général et un examen gynéco/obstétricaux.	$\forall (X), \text{Examen-Clinique}(X) \Rightarrow \exists Y \text{Etat-Général}(Y) \wedge \exists Z \text{Examen-G/O}(Z)$
Forceps	Dans un accouchement par forceps, on utilise l'instrument forceps.	$\forall (X), \text{Forceps}(X) \Rightarrow \exists Y \text{Instrumental}(Y) \wedge \text{Emploie}(X, Y)$
Examen G/O	Consiste en un examen	$\forall (X), \text{Examen-G/O}(X) \Rightarrow \exists Y \text{Speculum}(Y)$

Nom du concept	Description	Expression logique
	au speculum et les touchers pelviens.	$\wedge \exists Z \text{ Touché-P}(Z)$
Palpation	Examen réalisé par un médecin où on emploie une technique non instrumentale (la main).	$\forall (X), \text{Palpation}(X) \Rightarrow \exists Y \text{ Médecin}(Y) \wedge \text{Réalisé-par}(X, Y) \wedge \exists Z \text{ Non-Instrumental}(Z) \wedge \text{Emploie}(X, Z)$
Accouchement	Naissance d'un nouveau-né, il est assisté par une sage-femme. Peut-être par voie basse ou césarienne.	$\forall (X), \text{Accouchement}(X) \Rightarrow \exists Y \text{ Nouveau-né}(Y) \wedge \text{Donne}(X, Y) \wedge \exists Z \text{ Sage-femme}(Z) \wedge \text{Assisté-par}(X, Z)$
Examen para-clinique	Peut-être des examens biologiques, bactériologiques, radiologiques, ou anatomie-pathologique.	$\forall (X) \text{ Examen-paraclinique}(X) \Rightarrow \text{Biologie}(X) \vee \text{Bactériologie}(X) \vee \text{Anapath}(X) \vee \text{Imagerie}(X)$
Surveillance	Elle peut être biologique ou Clinique	$\forall (X) \text{ Surveillance}(X) \Rightarrow \text{Biologique}(X) \vee \text{Clinique}(X)$
Traitement	Traitement peut être préventif ou curatif.	$\forall (X) \text{ Traitement}(X) \Rightarrow \text{Préventif}(X) \vee \text{Curatif}(X)$
Structure-santé	Peut-être un hôpital, une clinique ou un bureau de santé.	$\forall (X) \text{ structure-santé}(X) \Rightarrow \text{Hôpital}(X) \vee \text{Clinique}(X) \vee \text{bureau-santé}(X)$
Fœtus	un fœtus est porté par une patiente.	$\forall (X), \text{Fœtus}(X) \Rightarrow \exists Y \text{ Patiente}(Y) \wedge \text{Porté-par}(X, Y)$
Maladie	Peut-être infectieuse, endocrinienne, cardiaque, tumorale, gynécologique, dermatologique, ou hématologique.	$\forall (X) \text{ Maladie}(X) \Rightarrow \text{Infectieuse}(X) \wedge \text{Endocrinienne}(X) \wedge \text{Hématologique}(X) \wedge \text{Dermatologique}(X) \wedge \text{Gynécologique}(X) \wedge \text{Cardiaque}(X) \wedge \text{Tumoral}(X)$

Nom du concept	Description	Expression logique
Biopsie	Un examen anatomie-pathologique, qui emploie une technique instrumentale.	$\forall (X), \text{Biopsie}(X) \Rightarrow \exists Y \text{ Instrumental}(Y) \wedge \text{emploie}(X, Y)$
GHR	Une grossesse à hauts risques peut causer un avortement et qui nécessite une surveillance.	$\forall (X), \text{GHR}(X) \Rightarrow \exists Y \text{ Avortement}(Y) \wedge \text{peut-causer}(X, Y) \wedge \exists Z \text{ Surveillance}(Z) \wedge \text{Nécessite}(X, Y)$
Avortement	C'est une interruption de grossesse.	$\forall (X), \text{Avortement}(X) \Rightarrow \exists Y \text{ Grossesse}(Y) \wedge \text{Interrompt}(X, Y)$
Signes-fcls	C'est les symptômes qu'apparaît chez une patiente et qui sont apaisés par un traitement.	$\forall (X), \text{Signes-fcls}(X) \Rightarrow \exists Y \text{ Patiente}(Y) \wedge \text{Apparaît}(X, Y) \wedge \exists Z \text{ Traitement}(Z) \wedge \text{Apaisé-par}(X, Z)$

Tableau 3.6. Table des axiomes logiques.

➤ 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 3.7 illustre quelques instances pour chaque classe.

Concept	Nom de l'instance	Attribut	Valeur
Médecin	X Y	Nom	Bouzidi
		Prénom	Tarek
		Grade-médecin	Professeur
		Spécialité-médecin	
		service	chirurgie générale et digestive
		Tel	0000000000
		Fax	XY@yahoo.fr
		Email	XXX – Constantine

Concept	Nom de l'instance	Attribut	Valeur
		Adresse	
Antimitotique	Cisplatine	Nom Nom-commercial Forme Posologie Dosage Mode-admin Effet-secondaire Type-med	CISPLATINE MYLAN Cisplatine Injectables 50 à 120 mg/m ² Individuelle Cutané Toxicité rénale, auditive, hématologique, neuropathies périphériques, troubles gastro- intestinale. Cancérologie et hématologie
M. Tumoral	Cancer du col	Stade Nom-maladie Debut-maladie Histoire Pronostic Complication Type-maladie	2 Cancer du col utérin 20/06/ 2009 GFHDHGDHGDHGF Bon Non Tumeur
Instrumental	Endoscope	Descrip-tech Nom-outil Technique-outil Fonction	Tube, constitué de fibres optiques conduisant la lumière équipé de petites caméras qui retransmettent l'image sur un écran. Endoscope Introduire l'endoscope dans un conduit ou une cavité de l'organisme. visualisation d'opérations chirurgicales intrusives ou l'observation.

Chapitre 3 : Conception d'une ontologie hybride dans le domaine médical

Concept	Nom de l'instance	Attribut	Valeur
	Trocart	<p>Descrip-tech</p> <p>Nom-outil</p> <p>Technique-outil</p> <p>Fonction</p>	<p>Instrument chirurgical constitué d'une tige métallique contenue dans une canule et terminée par une pointe coupante.</p> <p>Trocart</p> <p>l'aiguille au contact de la lésion, de forer au trocart puis d'animer la pointe du trépan de quelques oscillations et d'aspirer fortement à la seringue pour décrocher le prélèvement.</p> <p>Pour effectuer des ponctions, et des biopsies.</p>
Antimitotique	Adriamycine	<p>Nom</p> <p>Nom-commercial</p> <p>Forme</p> <p>Posologie</p> <p>Dosage</p> <p>Mode-admin</p> <p>Effet-secondaire</p> <p>Type-med</p>	-
M. infectieuse	Salpingite	<p>Nom-maladie</p> <p>Type-maladie</p>	
ATCD-Gyn	Abondance-règles	Descrip-ATCD	Abondance des règles
Hôpital	Ain naâdja	<p>Dénomination</p> <p>Adresse</p> <p>Tel</p>	<p>HSA Ain Naadja</p> <p>Commune kouba Alger</p>

Chapitre 3 : Conception d'une ontologie hybride dans le domaine médical

Concept	Nom de l'instance	Attribut	Valeur
		Fax Email Type-hôpital	(021) 54 54 54 (021) 54 54 53 - Militaire
	CHU Benbadis	Dénomination Adresse Tel Fax Email Type-hôpital	CHU Benbadis Rue Bensghir Abdelouahab constantine 31 64 16 07 / 031 64.29.72 / 031 64.29.73 031 64 17 00 chuconstantine@ibnsina.and.s.dz Universitaire
Clinique	EHS Sidi Mabrouk	Dénomination Adresse Tel Fax Spécialité Type-clinique	EHS Sidi mabrouk mère et enfant. sidi mabrouk, – Constantine 031683700 031 61 43 94 031 62 55 35 Gynécologie/obstétrique Etatique, maternité
	Ibn sina	Dénomination Adresse Tel fax Spécialité Type-clinique	Laboratoire Ibn sina 4 bd Ammar Boughaba- constantine 031.93.71.07 031.93.71.07 Analyses médicales Laboratoire
Groupage	O positif	Rehsus ABO	Positif O
	A Negatif	Rehsus	Negatif

Concept	Nom de l'instance	Attribut	Valeur
		ABO	A
...

Tableau 3.7. Table des instances.

5.4. 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.

5.4.1. 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 Clinique est subsumée par la classe Structure-santé on écrit :

$$\text{Clinique} \sqsubseteq \text{Structure-santé}$$

Les définitions de différents concepts sont illustrées dans le tableau 3.8.

Concept	Définition	Relation de subsomption
Personne	Personne := Patiente \cup Agent-santé \cup Nouveau-né	Personne \sqsubseteq Thing
Agent-santé	Agent-santé := Médecin \cup Sage-femme \cup TSS \cup Infirmier	Agent-santé \sqsubseteq Personne
Médecin	Médecin := Agent-santé \wedge (\exists travaille Structure-santé) \wedge (\exists examine. Patiente) \wedge (\exists réalise. Consultation) \wedge (\exists dépiste. Maladie) \wedge (\exists prescrit.	Médecin \sqsubseteq Agent-santé

Concept	Définition	Relation de subsumption
	Traitement)	
structure-santé	Structure-santé := hôpital \cup Clinique \cup Bureau-santé	structure-santé \subseteq Thing
Hôpital	Hôpital := (\exists 1 Renferme.Agent- santé) \cap (\exists 1 Se-rend.Patiente)	Hôpital \subseteq structure-santé
Maladie	Maladie := Infectieuse \cup Endocrinienne \cup Hématologique \cup Dermatologique \cup Tumoral \cup Gynécologique \cup Cardiaque	Maladie \subseteq Thing
Traitement	Traitement := Curatif \cup Préventif	Traitement \subseteq Thing
Curatif	Curatif := Electro-Coagulant \cup Chirurgie \cup Tr-Médicamenteux \cup Célioscopie \cup Tr- Rayonnement \cup Disposition- Inter-Ut \cup Cryothérapie	Curatif \subseteq traitement
Tr- médicamenteux	Tr-Médicamenteux : = Traitement \cap (\exists Apaise. Signes-fcls) \cap (\exists Prescrit-par. Médecin) \cap (\exists Donnée-a- Patiente)	Tr-médicamenteux \subseteq Curatif
Technique	Technique := Instrumental \cup Non-instrumental	Technique \subseteq Consultation
Instrumental	Instrumental := Technique	Instrumental \subseteq Technique
Voie-basse	Voie-basse := Accouchement \cap	Voie-basse \subseteq Accouchement

Concept	Définition	Relation de subsumption
	(≥ 1 Emploi.Technique)	
Anapath	Anapath := Biopsie \cup Frottis \cup Cytoponction \cup Biopsie- extemporanée	Anapath \subseteq examen paraclinique
Inspection	Inspection := Etat-général	Inspection \subseteq Etat-général
β-HCG	β -HCG := Biologie	β HCG \subseteq Biologie
Groupage	Groupage := Biologie	Groupage \subseteq Biologie
Consultation	Consultation := Interrogatoire \cup Examen-clinique \cup Examen- paraclinique \cup Technique	Consultation \subseteq Thing
Examen clinique	Examen-clinique := Etat-général \cup Examen-G/O	Examen clinique \subseteq Consultation

Tableau 3.8. Définition des concepts et subsumption.

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. Par exemple la relation Avoir-paramètre qui relie une activité avec ses paramètres est spécifiée par :

<p>Présente: (patiente, signes-fcls) Porte : porté-par</p>
--

Le tableau 3.9 représente les définitions des différents rôles de notre ontologie.

Rôle	Couple (domaine, co-domaine)	Rôle inverse
Examine	(médecin, Patiente)	Examiné-par
Porte	(Patiente, Foetus)	Porté-par
Dépiste	(Médecin, Maladie)	-
Se-rend	(Patiente, Structure-santé)	-

Réalise	(Médecin, Consultation)	Réalisé-par
Demande	(Examen-clinique, Examen-paraclinique)	Complète
Emploie	(Examen-paraclinique, Technique)	Employé-dans
Apaise	(traitement, Signes-fcls)	Apaisé-par
Reçoit	(Patiente, traitement)	Donné-à
Présente	(Patiente, Signes-fcls)	Apparaît
Dévoile	(Examen-clinique, Maladie)	-
Exige	(Traitement, Surveillance)	-
Fini-par	(Grossesse, Accouchement)	-
Enfante	(Patiente, Nouveau-né)	Accouché-par
Assiste	(Sage-femme, Voie basse)	Assisté-par
Nécessite	(Maladie, Surveillance)	Est-nécessaire
Souffre	(Patiente, GHR)	-
Se-complice	(Grossesse, GHR)	-
Endure	(Patiente, Maladie)	-
Soigné-par	(Maladie, Traitement)	-
Fait	(Patiente, Grossesse)	-
Interrompu-par	(Grossesse, Avortement)	Interrompte
Travaille	(Agent-santé, structure-santé)	Renferme

Tableau 3.9. Les définitions des différents rôles.

5.4.2. Construction de la 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 : Endoscopie : Imagerie

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

Exemple : (Prinpérent, vomissement) : Apaise

Chapitre 3 : Conception d'une ontologie hybride dans le domaine médical

Le tableau 3.10 définit quelques assertions de concepts, et le tableau 3.11 définit quelques assertions de relations de notre ontologie.

Concept	Description
Médecin	Médecin : Bouzidi-Tarek
Hôpital	Hôpital : CHU Ibn Badiss
	Hôpital : Ben baatouche HSA
Clinique	Clinique : EHS-Sidi-Mabrouk
Sage-femme	Sage-femme : ZahiaBenhecine
Anti-inflammatoire	Anti-inflammatoire: Voltarene
Antalgique	Antalgique : Paracétamol
Antiémétique	Antiémétique : Prinpérent
M. Tumoral	M. Tumoral : Cancer du col
M. Infectieuse	M. Infectieuse : Salpingite
	M. Infectieuse : Infection urinaire
Instrumental	Instrumental : Speculum
	Instrumental : Trocart
Patiente	Patiente : Loussif Amina
	Patiente :Triha Amal
	Patiente : Amara Amal

Tableau 3.10. Description des assertions de concepts.

Relation	Description
Présente	(Triha Amal, Hémorragie) : Présente
Soigné-par	(cancer du col, Cisplatine) : Soigné-par
Endure	(Daas Meryem, hypertension) : Endure
Dépiste	(MokhnacheIssam, fibrome) : Dépiste
Apaise	(Prinperent, vomissement) : Apaise
Travaille	(ZaamoucheNardjess, Laboratoire Ibn Sina) : Travaille
Emploie	(Endoscopie, endoscope) : Emploie
Réalisé-par	(β HCG, ZaamoucheNardjess) : Réalisé-par
Souffre	(Ayachi Farah, GHR-anémie) : Souffre

Nécessite	(Salpingite, biologique) : Nécessite
Figure	(Speculum,EHS Sidi Mabrouk) : Figure
Employé-dans	(trocart, Biopsie) : Employé-dans
Reçoit	(Amara Amal, Antibiotique) : Reçoit
Prescrit	(MokhnacheIssam, Voltarene) : Prescrit
Enfante	(Ayachi Farah, AyachiNahla) : Enfante

Tableau 3.11. Description des assertions de relations.

5.5. Représentation sémantique des règles avec le langage SWRL

L'ontologie est formulée dans le langage de représentation de connaissance OWL .Cette implémentation est réalisée par l'outil Protégé et validée par le raisonneur Pellet. Le langage OWL pose certains problèmes à cause de quelques limitations en particulier dans la définition des propriétés. Pour résoudre ce problème, SWRL, une extension à OWL, a été développé pour rendre le langage encore plus expressif pour la description de ces propriétés.

L'enrichissement de l'ontologie par des règles consistant à injecter un système à base de règles dans la base de connaissance, permet d'ajouter via les règles SWRL des axiomes sur les propriétés décrites dans le fragment d'ontologie.

Dans notre travail, nous avons conçu une série de règles permettant l'enrichissement des modèles ontologiques de la description des connaissances dans le service de gynécologie. Ces règles sont essentiellement inspirées du travail [46], et d'autres acquises par l'intervention des experts du domaine. Ces règles peuvent être divisées en quatre catégories :

❖ Sélection des examens

Ces règles permettent de diriger la patiente vers un examen clinique ou s'il est nécessaire un examen complémentaire pour confirmer le diagnostic résultant du premier examen, ou pour trouver l'anomalie qui n'a pas été observée lors de l'examen clinique.

Le tableau 3.12 montre quelques exemples de ces règles ainsi que leurs interprétations :

N°	Règle SWRL	Interprétation
01	$Patiente (?x) \wedge Palpation (?y) \wedge Présence-masse (?y, true) \rightarrow Demande (?y, Biopsie)$	Si le résultat d'une palpation est la présence d'une masse chez une patiente donc cette dernière doit faire un examende Biopsie.

Chapitre 3 : Conception d'une ontologie hybride dans le domaine médical

02	$\text{Patiente} (?x) \wedge \text{Age-patiente} (?x, ?y) \wedge \text{swrlb:greaterThanOrEqual}(?y, 50) \wedge \text{Pariété}(?x, ?z) \wedge \text{swrlb:equal}(?z, 0) \rightarrow \text{Subit} (?x, \text{Mammographie})$	Une patiente âgée de plus de 50 ans et qui n'a jamais accouché doit subir une mammographie.
03	$\text{Frottis} (?x) \wedge \text{Résultat-frottis}(?x, \text{"anormal"}) \wedge \text{Examen-clinique} (?y) \rightarrow \text{Demande} (?y, \text{Biopsie})$	Si une patiente a subi un examen clinique de Frottis dont le résultat est anormal, elle doit subir une biopsie.
04	$\text{Patiente} (?x) \wedge \text{Fait}(?x, ?y) \wedge \text{Grossesse} (?y) \wedge \text{Palpation}(?z) \wedge \text{Douleur} (?z, \text{"lombaire"}) \rightarrow \text{Demande} (?z, \text{Créatinine})$	Une femme enceinte qui a une douleur lors de la palpation doit être orientée vers un examen para clinique de créatinine.
05	$\text{Patiente} (?x) \wedge \text{Porte}(?x, ?y) \wedge \text{Foetus}(?y) \wedge \text{MAF}(?y, \text{false}) \wedge \text{Subit}(?x, ?z) \wedge \text{Aus-Aus-fœtal}(?z) \wedge \text{Bruit-fœtal}(?z, \text{"aboli"}) \rightarrow \text{Demande} (?z, \text{Echographie})$	Une patiente portant un fœtus dont le bruit fœtal est aboli, est orientée vers une échographie.
06	$\text{Patiente}(?x) \wedge \text{Fait}(?x, ?y) \wedge \text{Grossesse}(?y) \wedge \text{Age_grossesse}(?y, ?z) \wedge \text{swrlb:lessThanOrEqual}(?z, 12) \wedge \text{Subit}(?x, ?a) \wedge \text{Examen_G.O}(?a) \wedge \text{Résultat-EGO}(?a, \text{"hémorragie"}) \rightarrow \text{Demande} (?a, \text{Echographie})$	Une patiente dans les 12 premières semaines de grossesse qui présente une hémorragie est orientée vers une échographie.
07	$\text{Patiente} (?x) \wedge \text{Présente}(?x, ?y) \wedge \text{SF_objectifs}(?y) \wedge \text{Pâleur}(?y, \text{true}) \rightarrow \text{Subit} (?x, \text{FNS})$	Une patiente qui présente une pâleur est orientée vers un examen de type FNS.
08	$\text{Patiente} (?x) \wedge \text{Fait}(?x, ?y) \wedge \text{Grossesse}(?y) \wedge \text{Age_grossesse}(?y, ?z) \wedge \text{swrlb:lessThanOrEqual}(?z, 12) \rightarrow \text{Subit}(?x, \text{Biologie}) \wedge \text{Subit}(?x, \text{Imagerie})$	Une patiente doit subir un examen de biologie et un autre d'imagerie au courant du 1er trimestre de sa grossesse.
09	$\text{Patiente}(?x) \wedge \text{Fait}(?x, ?y) \wedge \text{Grossesse}(?y) \wedge \text{Age_grossesse}(?y, ?z) \wedge \text{swrlb:greaterThan}(?z, 12) \wedge \text{swrlb:lessThanOrEqual}(?z, 24) \rightarrow \text{Subit}(?x, \text{Biologie}) \wedge \text{Subit}(?x, \text{Imagerie})$	Une patiente doit subir un examen de biologie et un autre d'imagerie au courant du 2ème trimestre de sa grossesse.

10	$\text{Patiente}(\text{?x}) \wedge \text{Fait}(\text{?x}, \text{?y}) \wedge \text{Grossesse}(\text{?y})$ $\wedge \text{Age_grossesse}(\text{?y}, \text{?z}) \wedge \text{swrlb:greaterThan}(\text{?z}, 24) \wedge \text{swrlb:lessThanOrEqual}(\text{?z}, 42) \rightarrow$ $\text{Subit}(\text{?x}, \text{Biologie}) \wedge \text{Subit}(\text{?x}, \text{Imagerie})$	Une patiente doit subir un examen de biologie et un autre d'imagerie au courant du 3ème trimestre de sa grossesse.
11	$\text{Patiente}(\text{?x}) \wedge \text{Subit}(\text{?x}, \text{?y}) \wedge \text{Examen_G.O}(\text{?y})$ $\wedge \text{ResultEGO}(\text{?y}, \text{"anomalie"}) \rightarrow$ $\text{Demande}(\text{?y}, \text{Colposcopie})$	Une patiente qui a subi un examen G/O dont le résultat est anormal, est orientée vers une colposcopie.
12	$\text{Patiente}(\text{?x}) \wedge \text{Subit}(\text{?x}, \text{?y}) \wedge \text{Palpation}(\text{?y}) \wedge$ $\text{Présence-Masse}(\text{?y}, \text{true}) \rightarrow$ $\text{Subit}(\text{?x}, \text{Mammographie})$	Une femme qui se rend en consultation et on observe une présence de masse dans les seins doit faire une mammographie.
13	$\text{Patiente}(\text{?x}) \wedge \text{Endure}(\text{?x}, \text{?y}) \wedge \text{Maladie}(\text{?y}) \wedge$ $\text{Nom-maladie}(\text{?y}, \text{"Hépatite"}) \rightarrow$ $\text{Subit}(\text{?x}, \text{Sérologie})$	Une patiente qui souffre d'une hépatite doit subir un examen de sérologie.

Tableau 3.12. Règles SWRL pour la sélection des examens.

❖ Diagnostic des maladies

Ces règles permettent de diagnostiquer les différentes maladies qui peuvent atteindre une Patiente durant sa vie, soit elle est enceinte ou pas. Suivant les signes fonctionnels présentés, Ou les résultats des examens cliniques ou des examens complémentaires, la maladie Correspondante est dégagée.

Des exemples de ces règles avec leurs sens sont illustrés par le tableau suivant :

N°	Règle SWRL	Interprétation
01	$\text{Patiente}(\text{?x}) \wedge \text{Régulé}(\text{?x}, \text{false}) \wedge$ $\text{Age_patiente}(\text{?x}, \text{?y}) \wedge \text{swrlb:lessThan}(\text{?y}, 35)$ $\rightarrow \text{Endure}(\text{?x}, \text{ménopause-précoce})$	Si une patiente âgée de moins de 35 ans et souffre d'abondances des règles alors, elle endure une ménopause précoce
02	$\text{Patiente}(\text{?x}) \wedge \text{Présente}(\text{?x}, \text{?y}) \wedge \text{SF-objectifs}(\text{?y})$ $\wedge \text{TAS}(\text{?y}, \text{?z}) \wedge$ $\text{swrlb:greaterThanOrEqual}(\text{?z}, 160) \wedge \text{TAD}(\text{?y}, \text{?a})$ $\wedge \text{swrlb:greaterThanOrEqual}(\text{?a}, 110)$ $\rightarrow \text{Endure}(\text{?x}, \text{Hypertension})$	Si une patiente présente une tension artérielle systolique supérieure ou égale à 160 et diastolique supérieure ou égale à 110 alors, elle endure une

Chapitre 3 : Conception d'une ontologie hybride dans le domaine médical

		hypertension.
03	$\text{Patiente} (?x) \wedge \text{Présente} (?x, ?y) \wedge \text{SF-objectifs} (?y)$ $\wedge \text{TAS} (?y, ?z) \wedge \text{swrlb:lessThanOrEqual} (?z, 100) \wedge$ $\text{TAD} (?y, ?a) \wedge \text{swrlb:lessThanOrEqual} (?a, 80)$ $\rightarrow \text{Endure} (?x, \text{Hypotension})$	Si une patiente présente une tension artérielle systolique inférieure ou égale à 100 et diastolique inférieure ou égale à 80, alors elle endure une hypertension.
04	$\text{Patiente} (?x) \wedge \text{Régulé} (?x, \text{false}) \wedge$ $\text{Age-patiente} (?x, ?y) \wedge \text{swrlb:lessThan} (?y, 20) \wedge \text{swrlb:greaterThan} (?y, 15)$ $\rightarrow \text{Endure} (?x, \text{aménorrhée-primaire})$	Une patiente âgée entre 15 ans et 20 ans et n'est pas réglée, alors elle souffre d'une aménorrhée primaire.
05	$\text{Patiente} (?z) \wedge \text{Fait} (?z, ?x) \wedge \text{Grossesse} (?x)$ $\wedge \text{Age_grossesse} (?x, ?a) \wedge \text{swrlb:lessThan} (?a, 24) \wedge$ $\text{Subit} (?z, ?y) \wedge \text{Aus_foetal} (?y) \wedge$ $\text{Bruit_foetal} (?y, \text{"aboli"}) \rightarrow$ $\text{Interrompu_par} (?x, \text{Avortement})$	Une grossesse dans les 2 premiers trimestres qui présente un bruit fœtal aboli, est interrompue par un avortement.
06	$\text{Patiente} (?x) \wedge \text{Subit} (?x, ?y) \wedge \text{Créatinine} (?y)$ $\wedge \text{Valeur-Créatinine} (?y, ?z) \wedge \text{swrlb:greaterThan} (?z, 12) \rightarrow \text{Endure} (?x, \text{Rénale})$	Une patiente qui a la valeur de la créatinine au dessous de 12 souffre d'une maladie rénale.
07	$\text{Patiente} (?x) \wedge \text{Fait} (?x, ?y) \wedge \text{Grossesse} (?y)$ $\wedge \text{Age_grossesse} (?y, ?z) \wedge \text{swrlb:greaterThan} (?z, 24) \wedge \text{Subit} (?x, ?a) \wedge \text{Examen_G.O} (?a) \wedge$ $\text{Result-EGO} (?a, \text{"hémorragie"}) \rightarrow$ $\text{Endure} (?x, \text{Hématologique})$	Une patiente qui présente une perte sanglante, endure un cancer de l'endomètre et doit recevoir une biopsie.
08	$\text{Patiente} (?x) \wedge \text{Régulé} (?x, \text{true}) \wedge \text{Présente} (?x, ?y)$ $\wedge \text{SF_subjectifs} (?y) \wedge$ $\text{Descrip_SFS} (?y, \text{"règles_abondantes"}) \rightarrow$ $\text{Endure} (?x, \text{C_endomètre}) \wedge \text{Subit} (?x, \text{Biopsie})$	Une patiente qui présente des règles abondantes, endure un cancer de l'endomètre et doit recevoir une biopsie.
09	$\text{Patiente} (?x) \wedge \text{Régulé} (?x, \text{true}) \wedge \text{Présente} (?x, ?y)$ $\wedge \text{SF_subjectifs} (?y)$ $\wedge \text{Descrip_SFS} (?y, \text{"saignement_irrégulies"})$ $\rightarrow \text{Endure} (?x, \text{C_endomètre}) \wedge \text{Subit} (?x, \text{Biopsie})$	Une patiente qui présente un saignement irrégulier peut endurer un cancer de l'endomètre et doit être orienté vers une biopsie.

Chapitre 3 : Conception d'une ontologie hybride dans le domaine médical

10	$\text{Patiente} (?x) \wedge \text{Subit} (?x, ?a) \wedge \text{FNS} (?a) \wedge$ $\text{hémoglobine} (?a, ?b) \wedge \text{swrlb:lessThan} (?b, 10)$ $\rightarrow \text{Endure} (?x, \text{anémie})$	Une patiente dont le taux d'hémoglobine est inférieur à 10, souffre d'une anémie.
11	$\text{Patiente} (?x) \wedge \text{Présente} (?x, ?y) \wedge \text{SF_objectifs} (?y)$ $\wedge \text{Descrip-SF} (?y, \text{"Leuchonée"}) \wedge \text{Endure} (?x, ?z) \wedge$ $\text{Maladie} (?z) \wedge \text{Nom-maladie} (?z, \text{"Fièvres"}) \wedge$ $\text{Subit} (?x, ?a) \wedge \text{Palpation} (?a) \wedge \text{Douleur} (?a,$ $\text{"Mobilisation"}) \rightarrow \text{Endure} (?x, \text{Salpingite})$	Une femme qui présente des leuchonée et fièvre avec douleur à la mobilisation, elle endure une salpingite

Tableau 3.13. Règles SWRL pour le diagnostic des maladies.

❖ Sélection des traitements

Concernant cette catégorie de règles, un traitement est indiqué suivant la maladie d'une patiente, ou les signes qu'elle présente.

Nous mentionnons quelques exemples explicatifs de ces règles dans le tableau 3.14.

N°	Règle SWRL	Interprétation
01	$\text{Patiente} (?x) \wedge \text{GHR} (?y) \wedge \text{Souffre} (?x, ?y)$ $\wedge \text{Nom-GHR} (?y, \text{"diabétique"})$ $\rightarrow \text{Reçoit} (?x, \text{Antidiabétique})$	Une patiente qui souffre d'une grossesse à haut risque diabétique doit recevoir un traitement antidiabétique.
02	$\text{Patiente} (?x) \wedge \text{Endure} (?x, \text{Salpingite})$ $\rightarrow \text{Reçoit} (?x, \text{Antibiotique})$	Une patiente qui endure une maladie Salpingite, doit recevoir un traitement antibiotique.
03	$\text{Patiente} (?x) \wedge \text{Souffre} (?x, ?y) \wedge \text{GHR} (?y) \wedge$ $\text{Nom-GHR} (?y, \text{"diabétique"})$ $\rightarrow \text{Nécessite} (?y, \text{Surv-clinique})$	Une patiente souffrant d'une grossesse à haut risque diabétique, donc la grossesse nécessite une surveillance clinique
04	$\text{Patiente} (?x) \wedge \text{Biopsie} (?y) \wedge \text{subit} (?x, ?y)$ $\wedge \text{Descrip-biopsie} (?y, \text{"masse-bénigne"})$ $\rightarrow \text{Endure} (?x, \text{Tumorale}) \wedge \text{Reçoit} (?x, \text{Chirurgie})$	Une patiente qui subit un examen de biopsie dont le résultat est la découverte d'une masse bénigne, doit recevoir une chirurgie à cause de la présence d'une tumeur
05	$\text{Patiente} (?x) \wedge \text{Biopsie} (?y) \wedge \text{subit} (?x, ?y)$	Une patiente qui subit un examen

Chapitre 3 : Conception d'une ontologie hybride dans le domaine médical

	\wedge Descrip-biopsie (?y, "masse-maligne") \rightarrow Endure (?x, Tumorale) \wedge Reçoit (?x, Antimitotique)	de biopsie dont le résultat est la découverte d'une masse maligne, doit recevoir un antimitotique à cause de la présence d'une tumeur
06	Patiente(?x) \wedge Souffre(?x, ?z) \wedge GHR(?z) \wedge Nom-GHR(?z, "anémique") \wedge Subit(?x, ?a) \wedge FNS(?a) \wedge Nombre_GR(?a, ?b) \wedge swrlb:lessThan(?b, 10) \rightarrow Reçoit(?x, Transfusion)	Une patiente qui souffre d'une GHR anémique et qui subit un examen FNS donnant le nombre de globines rouges inférieur à 10, doit recevoir une transfusion.
07	Patiente (?x) \wedge Endure (?x, ?y) \wedge Tumorale(?y) \wedge Nom-maladie(?y, "Dysplasie") \rightarrow Reçoit (?x, Chimiothérapie)	Une patiente qui endure une maladie tumorale 'Dysplasie', doit recevoir une chimiothérapie.
08	Patiente (?x) \wedge Endure(?x, ?y) \wedge Tumorale(?y) \wedge Nom-maladie(?y, "C_endomètre") \rightarrow Reçoit (?x, Chirurgie)	Une patiente qui endure un cancer de l'endomètre, doit recevoir une chirurgie.
09	Patiente(?x) \wedge Subit(?x, ?y) \wedge T_vaginal(?y) \wedge Dil_col(?y, ?z) \wedge swrlb:lessThanOrEqual(?z, 3) \wedge Fait(?x, ?a) \wedge Grossesse(?a) \wedge Age_grossesse(?a, ?b) \wedge swrlb:greaterThanOrEqual(?b, 36) \rightarrow Reçoit(?x, Tocolitique)	Une patiente dans le 3eme trimestre de sa grossesse, et dont la dilatation du col est inférieure ou égale à 3 doigts doit prendre du Tocolitique.
10	Patiente(?x) \wedge Subit(?x, ?y) \wedge Facteurs-risques(?y) \wedge Descrip-frisque(?y, "HTA") \rightarrow Reçoit(?x, Anti_hypertenseur)	une patiente qui présente un facteur de risque HTA, doit recevoir un traitement antihypertenseur.
11	Patiente(?x) \wedge Subit(?x, ?y) \wedge Facteurs-risques(?y) \wedge Descrip-frisque(?y, "Diabète") \rightarrow Reçoit(?x, Antidiabétique)	Une patiente qui présente un facteur de risque diabète, doit recevoir un traitement antidiabétique.
12	Patiente(?x) \wedge Endure(?x, ?y) \wedge Endocrinienne(?y) \wedge Nom-maladie(?y, "Tyroïde") \rightarrow Reçoit(?x, Levothyrox)	Une patiente qui souffre d'une maladie endocrinienne thyroïde, doit recevoir un traitement

		antithyroïdien comme levothyrox.
13	$\text{Patiente}(?x) \wedge \text{Endure}(?x, ?y) \wedge \text{Tumorale}(?y) \wedge$ $\text{Nom-maladie}(?y, \text{"Cancer_endomètre"}) \wedge$ $\text{Stade}(?y, ?z) \wedge \text{swrlb:equal}(?z, 0) \rightarrow \text{Reçoit}(?x,$ $\text{Chirurgie})$	Une patiente qui souffre d'un cancer de l'endomètre et le stade de la maladie 0, doit recevoir un traitement chirurgical.

Tableau 3.14. Règles SWRL pour la sélection des traitements.

❖ Le choix du type d'accouchement

Ces règles permettent de décider si une patiente enceinte passe par un accouchement normal ou elle doit passer en césarienne selon la position du fœtus et selon le cas de la patiente.

Le tableau 3.15 montre des exemples de ces règles :

N°	Règle SWRL	Interprétation
01	$\text{Patiente}(?x) \wedge \text{Avoir}(?x, ?y) \wedge$ $\text{Ant-obstétricaux}(?y) \wedge \text{Fait}(?x, ?z) \wedge$ $\text{Grossesse}(?z) \rightarrow \text{Passe-en}(?x, \text{Césarienne})$	Une patiente enceinte qui a des antécédents obstétricaux doit passer en césarienne
02	$\text{Patiente}(?x) \wedge \text{GHR}(?y) \wedge \text{Souffre}(?x, ?y) \wedge$ $\text{Nom-GHR}(?y, \text{"HTA"})$ $\rightarrow \text{Passe-en}(?x, \text{Césarienne})$	Une patiente qui souffre d'une grossesse à haut risque de type HTA doit passer en césarienne
03	$\text{Patiente}(?x) \wedge \text{Fait}(?x, ?y) \wedge \text{Grossesse}(?y)$ $\wedge \text{Position-fœtus}(?y, \text{"Siège"})$ $\rightarrow \text{Passe-en}(?x, \text{Césarienne})$	Une patiente enceinte dont la position du fœtus est en « Siège » doit passer en césarienne
04	$\text{Patiente}(?x) \wedge \text{Fait}(?x, ?y) \wedge \text{Grossesse}(?y)$ $\wedge \text{Position-fœtus}(?y, \text{"Sommet"})$ $\rightarrow \text{Passe-en}(?x, \text{Normal})$	Une patiente enceinte dont la position du fœtus est en sommet passe en accouchement normal.
05	$\text{Patiente}(?x) \wedge \text{Subit}(?x, ?y) \wedge \text{Pelvimétrie}(?y)$ $\wedge \text{Résultat-pelvimétrie}(?y, \text{"Bassin-rétréci"})$ $\rightarrow \text{Passe-en}(?x, \text{Césarienne})$	Une patiente dont le bassin est de type rétréci, doit passer en césarienne.
06	$\text{Patiente}(?x) \wedge \text{Subit}(?x, ?y) \wedge \text{T-vaginal}(?y)$ $\wedge \text{Dil-col}(?y, ?z) \wedge \text{swrlb:lessThanOrEqual}(?z, 4) \wedge$ $\text{Fait}(?x, ?a) \wedge \text{Grossesse}(?a) \wedge$	Une patiente dans la 28ème semaine de la grossesse et dont la dilation du col est inférieure à 4 doigts doit passer en épisiotomie.

Chapitre 3 : Conception d'une ontologie hybride dans le domaine médical

	Age-grossesse (?a, ?b) \wedge swrlb:greaterThan(?b, 28) → Passe-en (?x, Episiotomie)	
07	Patiente(?x) \wedge Fait(?x, ?y) \wedge Grossesse(?y) \wedge Pariété(?x, 0) \wedge Subit(?x, ?z) \wedge Toucher_vaginal(?z) \wedge Dil-col(?z, ?a) \wedge swrlb:lessThan(?a, 8) → Passe-en(?x, Episiotomie)	Une patiente enceinte dont lapariété est 0, et la dilatation du col est inférieure à 8, passe en accouchement épisiotomie.
08	Patiente(?x) \wedge Fait(?x, ?y) \wedge Grossesse(?y) \wedge Age-grossesse(?y, ?z) \wedge swrlb:lessThanOrEqual(?z, 12) \wedge Endure(?x, ?a) \wedge Maladie(?a) \wedge Nom-maladie(?a, "Rubéole") → Interrompu-par(?y, Avortement)	Une patiente dans le 1 ^{er} trimestre de sa grossesse, et elle souffre d'une rubéole, donc sa grossesse est interrompue par un avortement.

Tableau 3.15. Règles SWRL pour le choix de l'accouchement.

Aussi, nous avons conçu une série de règles permettant d'interroger la base de connaissance de notre ontologie « **GynécOntologie** ».

Le tableau suivant montre des exemples de ces règles :

N°	Règle SWRL	Le rôle de la Règle SWRL
1	Patiente(?x) \wedge Passe-en(?x, Césarienne) → sqwrl:count(?x)	Permet de compter le nombre des patientes qui passe en accouchement par césarienne.
2	Patiente(?x) \wedge Passe-en(?x, Episiotomie) → sqwrl:count(?x)	Permet de compter le nombre des patientes qui passe en accouchement par Episiotomie.
3	Patiente(?x) \wedge Passe-en(?x, Voie_basse) → sqwrl:count(?x)	Permet de compter le nombre des patientes qui passe en accouchement par Voie basse.
4	GHR(?x) \wedge Peut-causer(?x, Avortement) → sqwrl:count(?x)	Permet de compter le nombre des grossesses haut risque qui Peut causer un Avortement.
5	Grossesse(?x) \wedge Interrompu-par(?x, Avortement) → sqwrl:count(?x)	Permet de compter le nombre des grossesses peuvent Interrompues par

Chapitre 3 : Conception d'une ontologie hybride dans le domaine médical

		Avortement.
6	$\text{Patiente}(\text{?x}) \wedge \text{Endure}(\text{?x}, \text{Cancer-sein})$ $\rightarrow \text{sqwrl:count}(\text{?x})$	Permet de compter le nombre des patientes attendre par un cancer du sein.
7	$\text{Patiente}(\text{?x}) \wedge \text{Subit}(\text{?x}, \text{Mammographie}) \wedge$ $\text{Age-patiente}(\text{?x}, \text{?y}) \rightarrow \text{sqwrl:avg}(\text{?y})$	Permet de compter la moyenne d'âges des patientes qui font une Mammographie.
8	$\text{Patiente}(\text{?x}) \wedge \text{Endure}(\text{?x}, \text{Cancer-sein}) \wedge$ $\text{Age-patiente}(\text{?x}, \text{?y}) \rightarrow \text{sqwrl:avg}(\text{?y})$	Permet de compter la moyenne d'âges des patientes qui souffrent d'un cancer du sein.
9	$\text{Patiente}(\text{?x}) \wedge \text{Endure}(\text{?x},$ $\text{Dermatologique}) \rightarrow \text{sqwrl:count}(\text{?x})$	Permet de compter le nombre des patientes attendre d'une maladie Dermatologique.
10	$\text{Patiente}(\text{?x}) \wedge \text{Endure}(\text{?x}, \text{Endocrinienne})$ $\rightarrow \text{sqwrl:count}(\text{?x})$	Permet de compter le nombre des patientes qui maladie par Endocrinienne.
11	$\text{Patiente}(\text{?x}) \wedge \text{Endure}(\text{?x}, \text{Gynécologique})$ $\rightarrow \text{sqwrl:count}(\text{?x})$	Permet de compter le nombre des patientes qui maladie par Gynécologique.
12	$\text{Patiente}(\text{?x}) \wedge \text{Endure}(\text{?x}, \text{Hématologique})$ $\rightarrow \text{sqwrl:count}(\text{?x})$	Permet de compter le nombre des patientes qui maladie par Hématologique.
13	$\text{Patiente}(\text{?x}) \wedge \text{Endure}(\text{?x}, \text{Infectieuse})$ $\rightarrow \text{sqwrl:count}(\text{?x})$	Permet de compter le nombre des patientes qui maladie par Infectieuse.
14	$\text{Patiente}(\text{?x}) \wedge \text{Endure}(\text{?x}, \text{Rénale})$ $\rightarrow \text{sqwrl:count}(\text{?x})$	Permet de compter le nombre des patientes qui maladie par Rénale.
15	$\text{Patiente}(\text{?x}) \wedge \text{Endure}(\text{?x}, \text{Tumorale})$ $\rightarrow \text{sqwrl:count}(\text{?x})$	Permet de compter le nombre des patientes qui maladie par Tumorale.
16	$\text{Patiente}(\text{?x}) \wedge \text{Fait}(\text{?x}, \text{?y}) \wedge \text{Grossesse}(\text{?y})$ $\wedge \text{Nbr-fœtus}(\text{?y}, \text{?z})$ $\wedge \text{swrlb:greaterThanOrEqual}(\text{?z}, 2)$ $\rightarrow \text{sqwrl:count}(\text{?x})$	Permet de compter le nombre des patientes qui donne naissance à des jumeaux ou plus.

17	$\text{Patiente}(?x) \wedge \text{Examiné-par}(?x, \text{Bouzidi-Tarek}) \rightarrow \text{sqwrl:count}(?x)$	Permet de compter le nombre des patientes qui Examiné par le médecin BouzidiTarek.
18	$\text{Technique}(?x) \wedge \text{Figure}(?x, \text{CHU_Ibn_Badiss}) \rightarrow \text{sqwrl:select}(?x)$	Permet d'affiche les techniques qui se trouvent dans la structure santé CHU Ibn Badiss.
19	$\text{Patiente}(?x) \wedge \text{Se-rend}(?x, \text{CHU_Ibn_Badiss}) \rightarrow \text{sqwrl:count}(?x)$	Permet de compter le nombre des patientes qui se rend dans la structure santé CHU Ibn Badiss.
...

Tableau 3.16. Règles SWRL pour interroger notre ontologie.

6. Conclusion

Dans ce chapitre, nous avons présenté l'architecture du système de raisonnement sur notre ontologie hybride. Le principe du fonctionnement du système de raisonnement est d'interconnecter un moteur de règles et un raisonneur s'exécutant en parallèles, leurs inférences seront effectuer, et de nouveaux faits vont être insérés.

Ensuite, nous avons décrit le processus de développement de l'ontologie hybride GynécOntologie ; une ontologie dédiée au domaine médical et particulièrement la spécialité gynécologie. Ce processus est basé principalement sur la méthodologie «METHONTOLOGY ». Il est constitué de cinq phases successives. La spécification, première phase de ce processus, a permis l'analyse des différentes sources de données concernant le domaine de l'ontologie à construire, et d'établir un document de spécification pour celle-ci, exprimé en langage naturel. Ce document décrit le domaine, la portée, les utilisateurs et les caractéristiques de l'ontologie.

La phase de conceptualisation a permis d'identifier et de structurer les connaissances du domaine sous forme d'un ensemble de concepts inter-liés. Quant à la formalisation, elle a permis la représentation du modèle conceptuel obtenu par un formalisme de représentation de connaissances qui est la logique de descriptions SHIQ.

Chapitre 3 : Conception d'une ontologie hybride dans le domaine médical

Nous avons abouti à un ensemble de représentations intermédiaires semi-formelles, et nous avons obtenu une ontologie conceptuelle. Les résultats de l'étape de conception vont servir pour l'opérationnalisation de cette ontologie qui aura lieu dans le prochain chapitre.

Afin d'enrichir l'ontologie OWL par des règles SWRL permettant d'assurer une plus haute expressivité relationnelle, nous avons étendu le processus de construction de l'ontologie GynécOntologie par l'édition des règles SWRL pour ajouter des axiomes sur les propriétés.

Chapitre 4

**Implémentation
Et mise en Œuvre
« GynécOntologie »**

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 l'ontologie proposée.

Un langage de définition d'ontologie pour le Web est nécessaire pour implémenter notre ontologie « GynécOntologie ». pour cela nous utilisons le langage OWL qui offre des fonctionnalités sémantique plus riches que le langage RDFS.

Ainsi, nous utiliserons SWRL comme langage de règles, pour définir des règles. Ces dernières sont faciles à écrire et apporteront une plus haute expressivité.

Pour l'implémentation du système de raisonnement, nous avons choisi des outils standards du Web sémantique : Protégé-OWL et ses plugins, le raisonneur Pellet, et le moteur de règles Jess.

Aussi nous parlons des outils utilisés pour construire notre application « Gynomed », 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 « GynécOntologie » avec l'extension (.OWL).

2. Les outils d'implémentation

2.1 Plateforme Protégé

PROTEGE OWL est une interface modulaire, développée au Stanford Medical Informatics 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 [49].

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 [49].

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

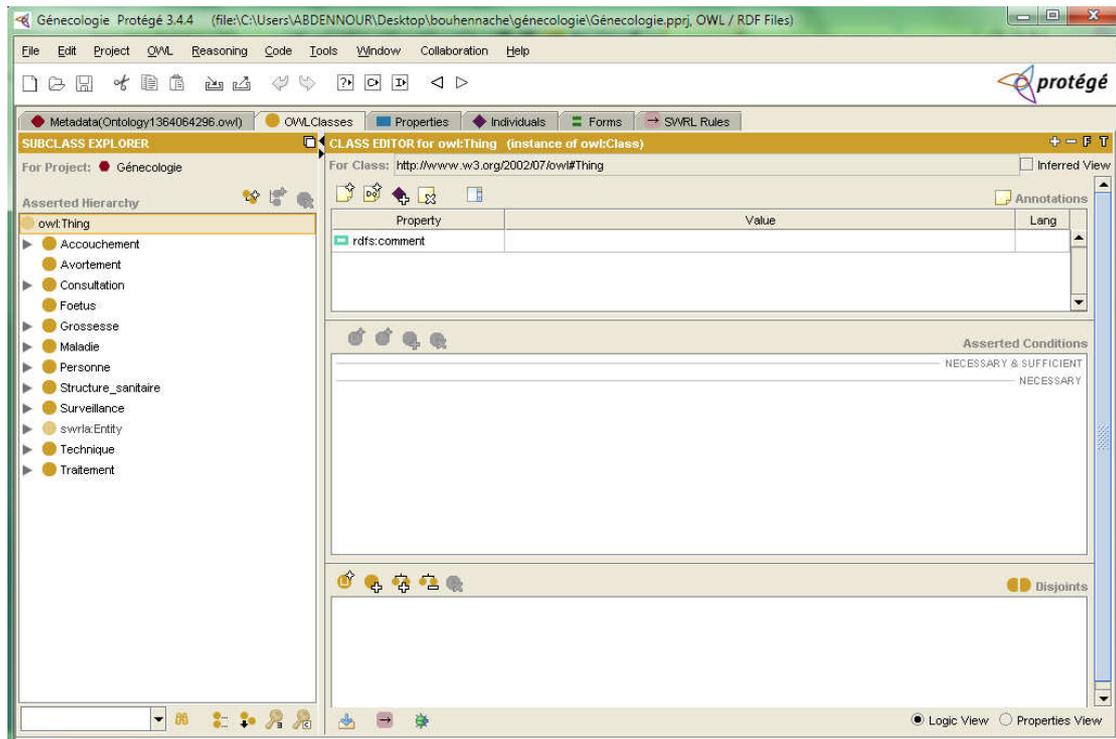


Figure 4.1. L'interface principale de Protégé.

2.2. Plugins utilisés

2.2.1. SWRL Tab

L'éditeur SWRL est un plugin de Protégé OWL permettant l'édition interactive des règles SWRL. Les utilisateurs peuvent créer, modifier, lire et écrire les règles SWRL, il permet aux utilisateurs de sélectionner des classes OWL, propriétés et individus de la base de connaissances (ontologie), les charger et les insérer dans la règle en cours de création.

La figure 4.2 représente l'interface interactive de SWRL TAB.

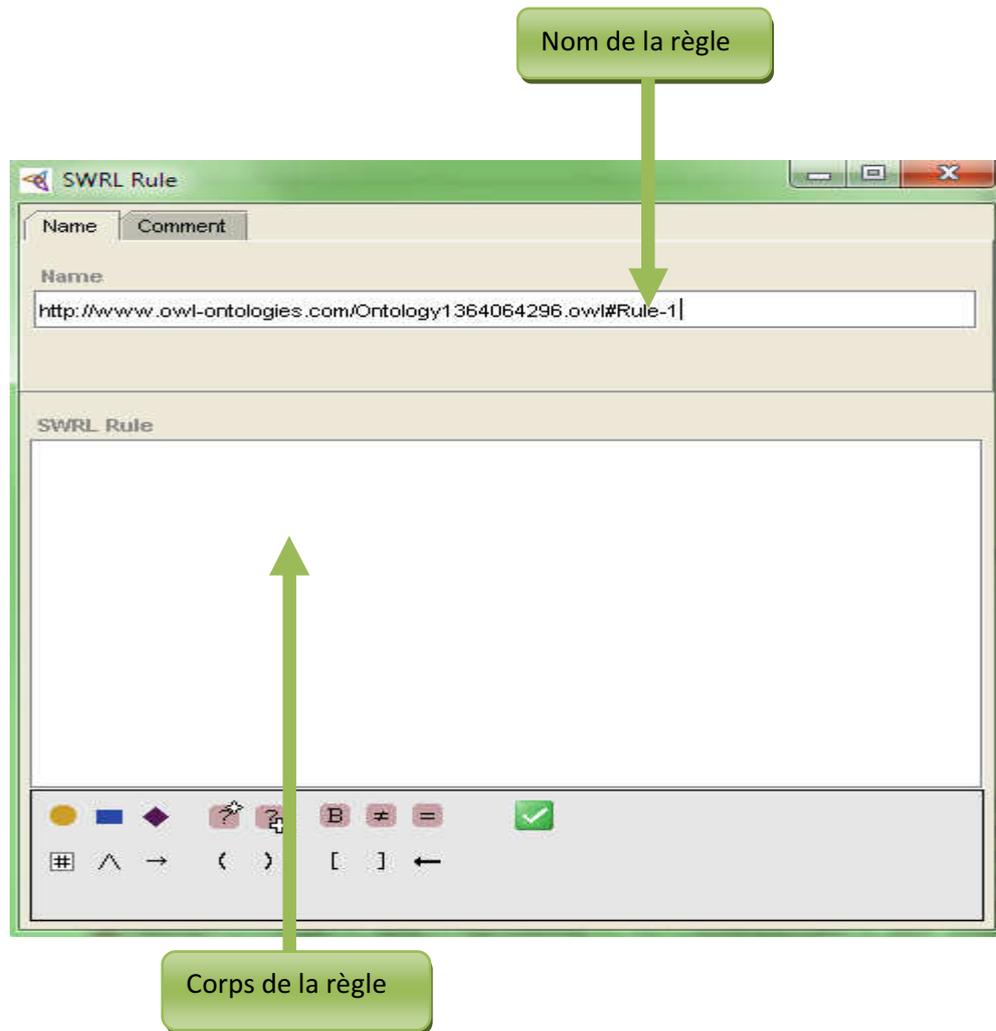


Figure 4.2. L'éditeur de règles SWRL.

2.2.2. SWRL Jess Tab

SWRL Jess Tab est un plugin dans Protégé OWL, qui prend en charge l'exécution des règles SWRL en utilisant le moteur de règles Jess. Il fournit une interface graphique (Figure 4.3).

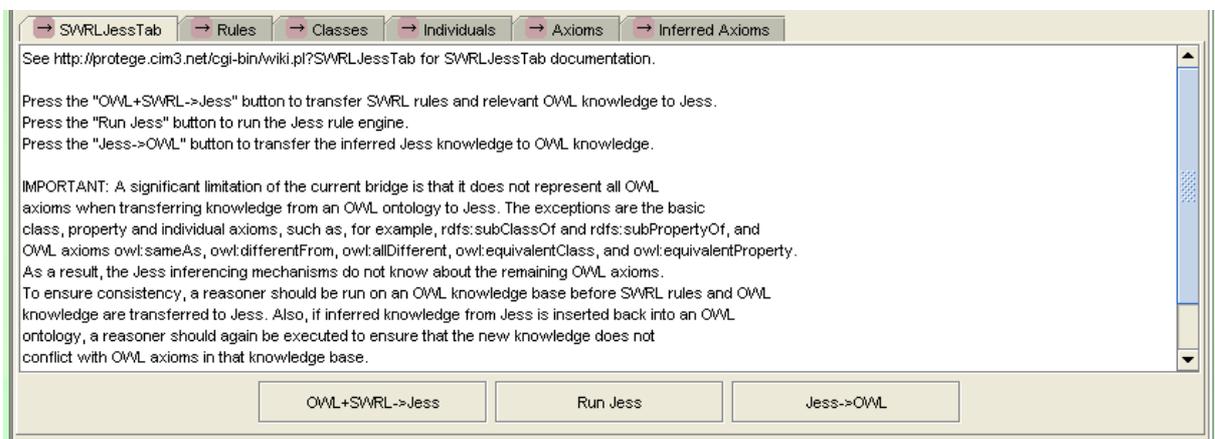


Figure 4.3. Le plugin SWRL Jess Tab.

L'utilisateur peut choisir les règles SWRL pour les exécuter, ou une règle ou plus, comme il peut visualiser ce que la règle peut déduire des nouveau fait ou non.

2.3. 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.

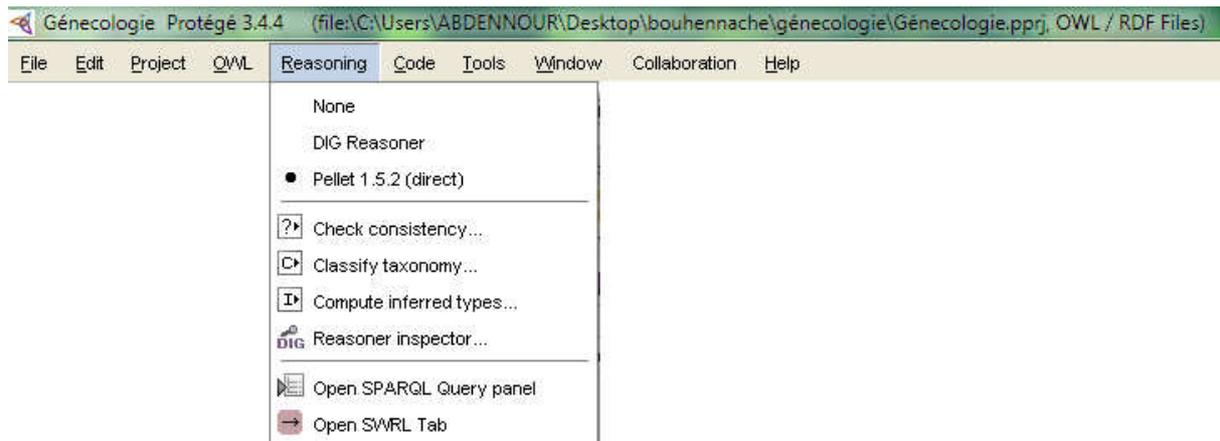


Figure 4.4. Le classifieur Pellet.

2.4. Le Moteur Jess

Jess est un moteur de règles basé sur Java et l'environnement de script que le SWRL Jess Tab dépend. Pour l'installation de Jess, copier le fichier jess.jar contenu dans la distribution de Jess et le coller dans le répertoire plugins/edu.stanford.smi.protege.owl d'installation de Protégé OWL.

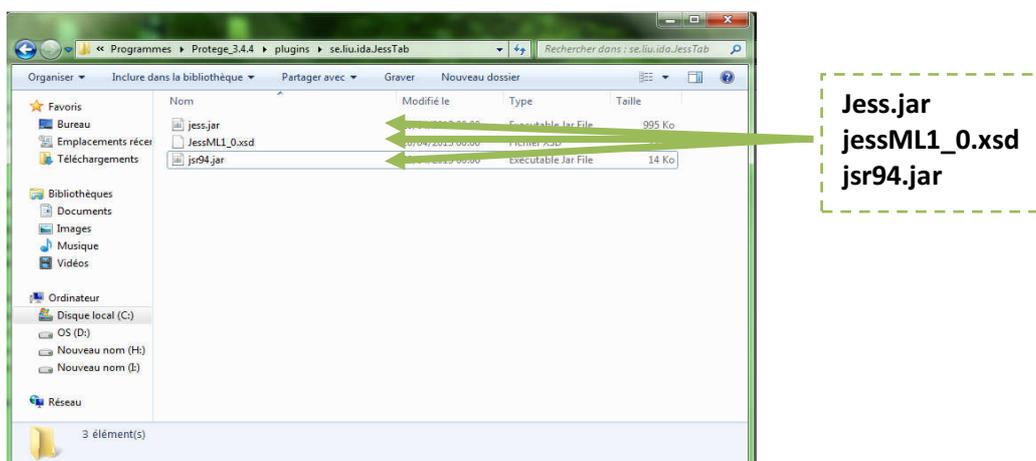


Figure 4.5. Installation de Jess.

3. Création de l'ontologie avec Protégé OWL

3.1. 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 4.6 présente la procédure de création d'un concept sous Protégé OWL.

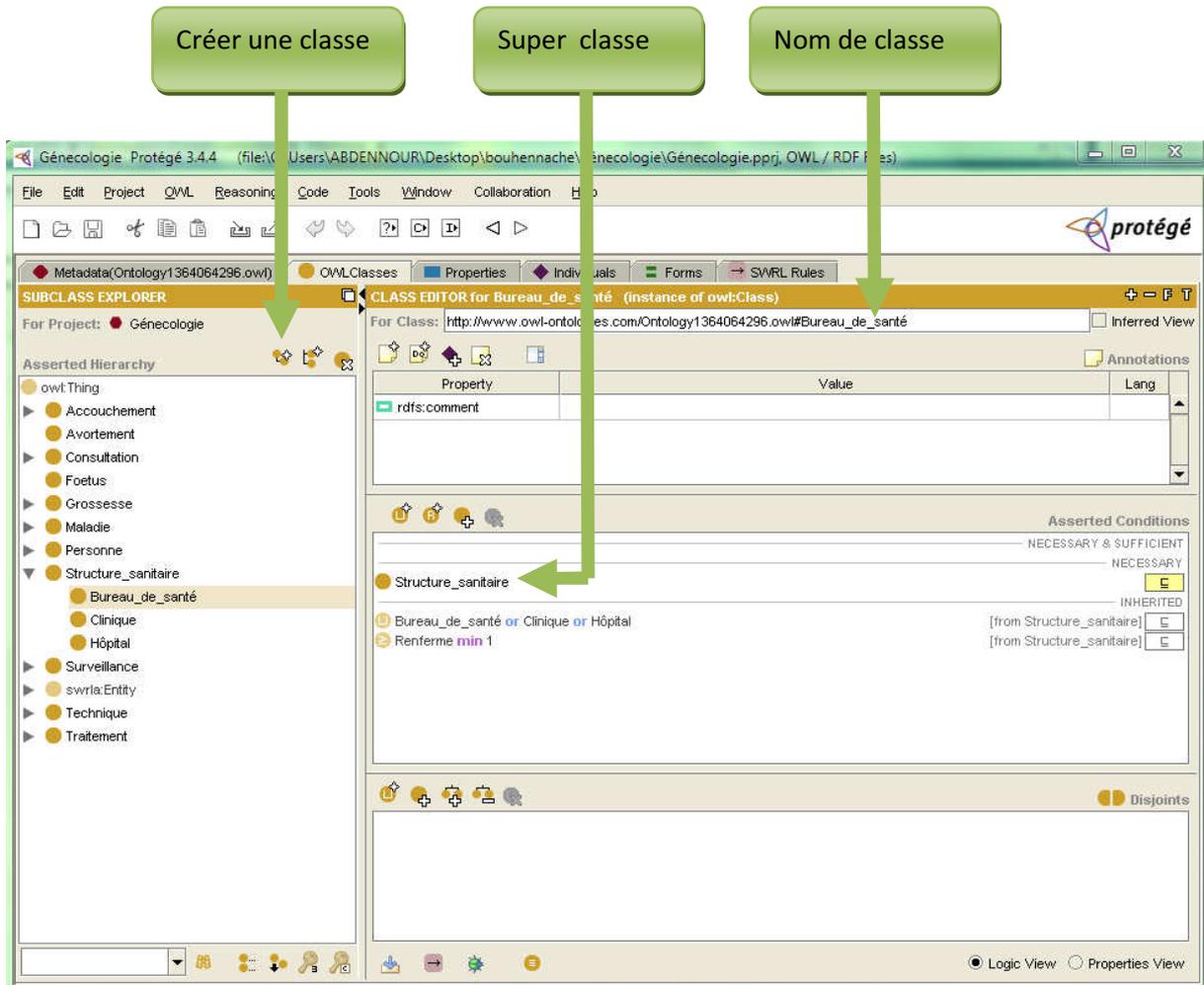


Figure 4.6. Création de classes.

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

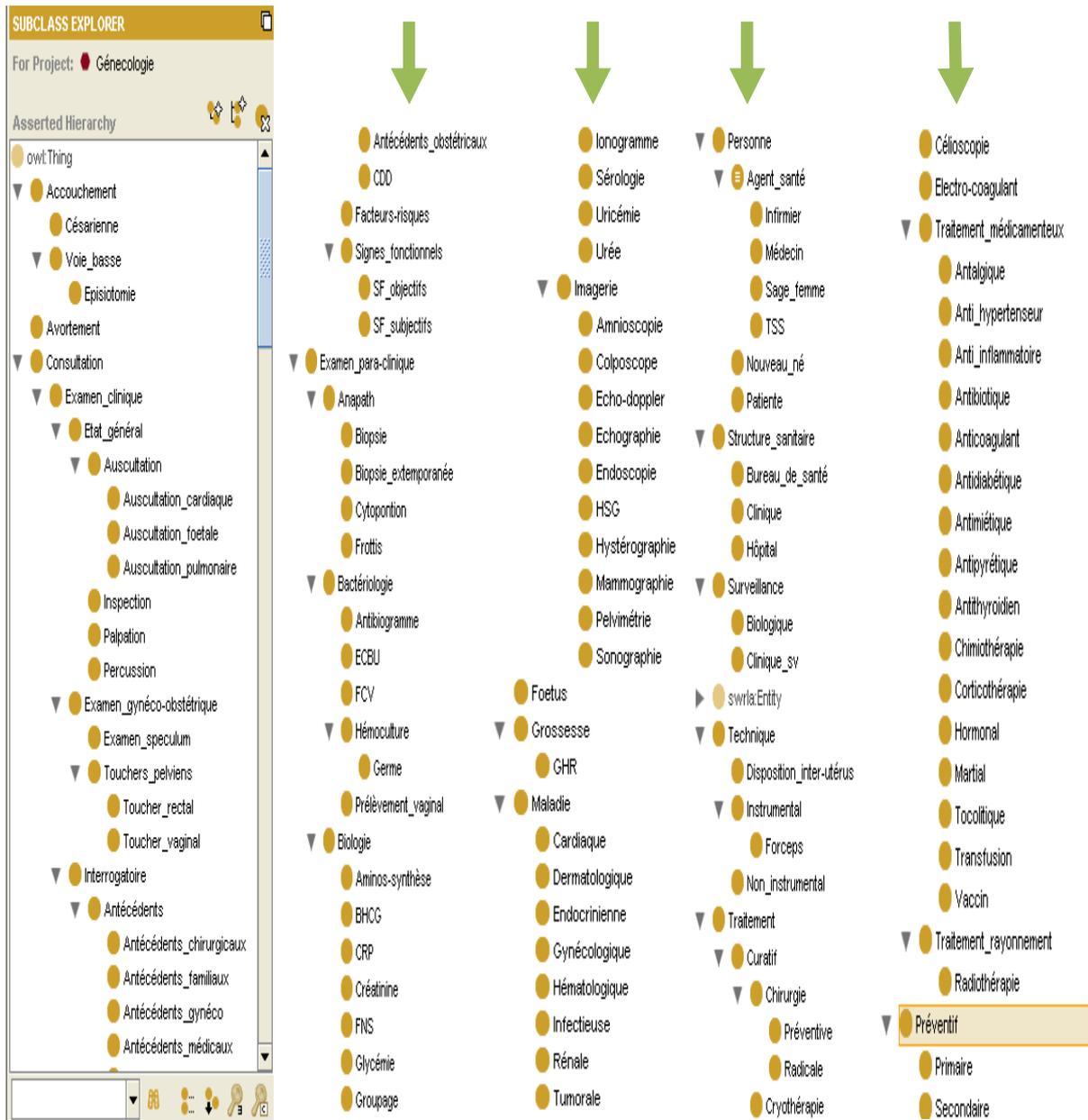


Figure 4.7. Hiérarchies des concepts

3.2. 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 4.8 montre les possibilités de PROTEGE OWL pour la création des propriétés.

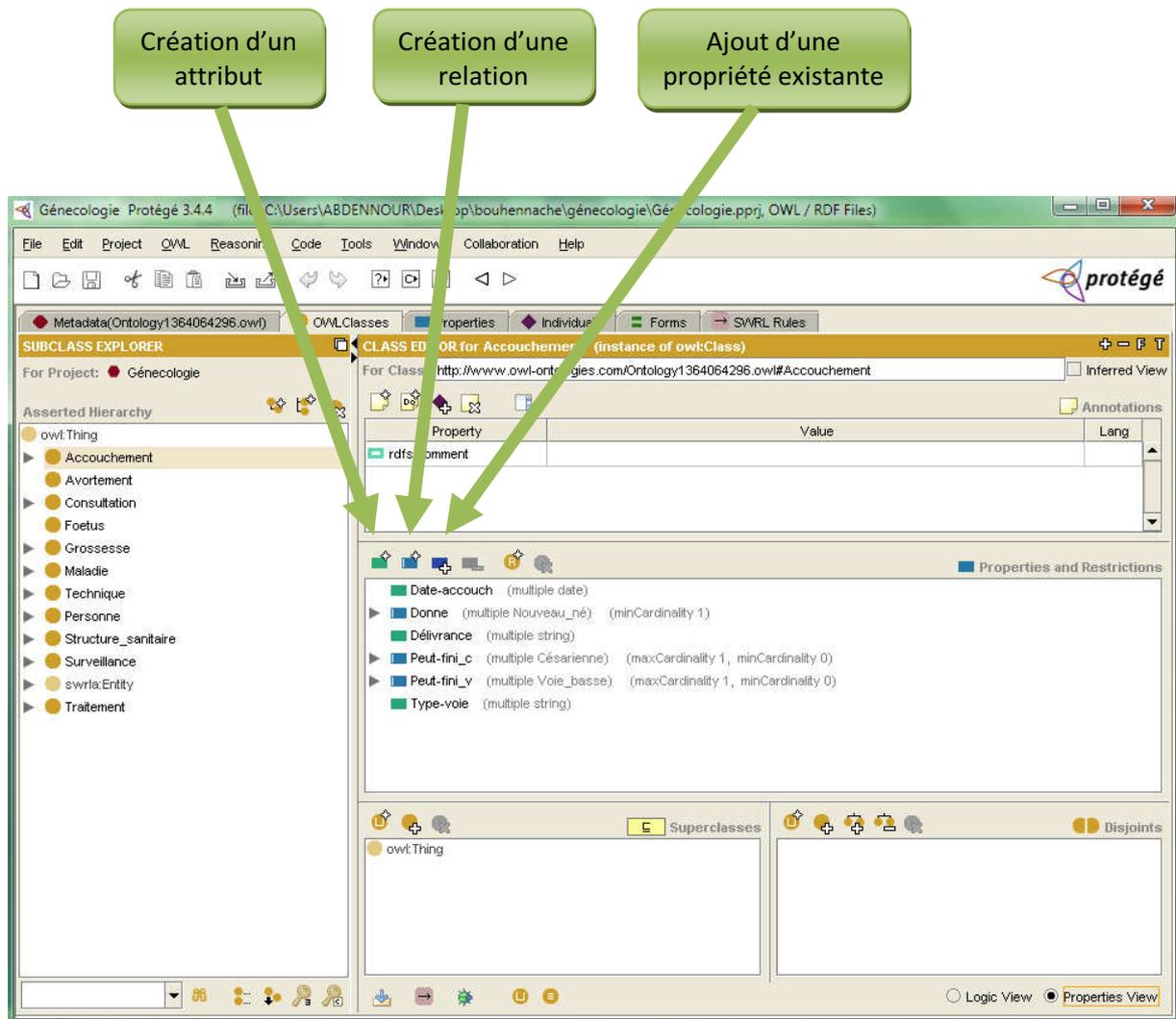


Figure 4.8. Création de propriétés pour une classe.

3.2.1. Création d'attributs (DataTypeProperty)

La création des attributs se fait en cliquant sur le bouton 'dataTypeProperty', ce qui génère la fenêtre présentée par la figure 4.9. cette fenêtre permet de fournir les champs pour remplir le nom, le domaine, et le type de l'attribut.

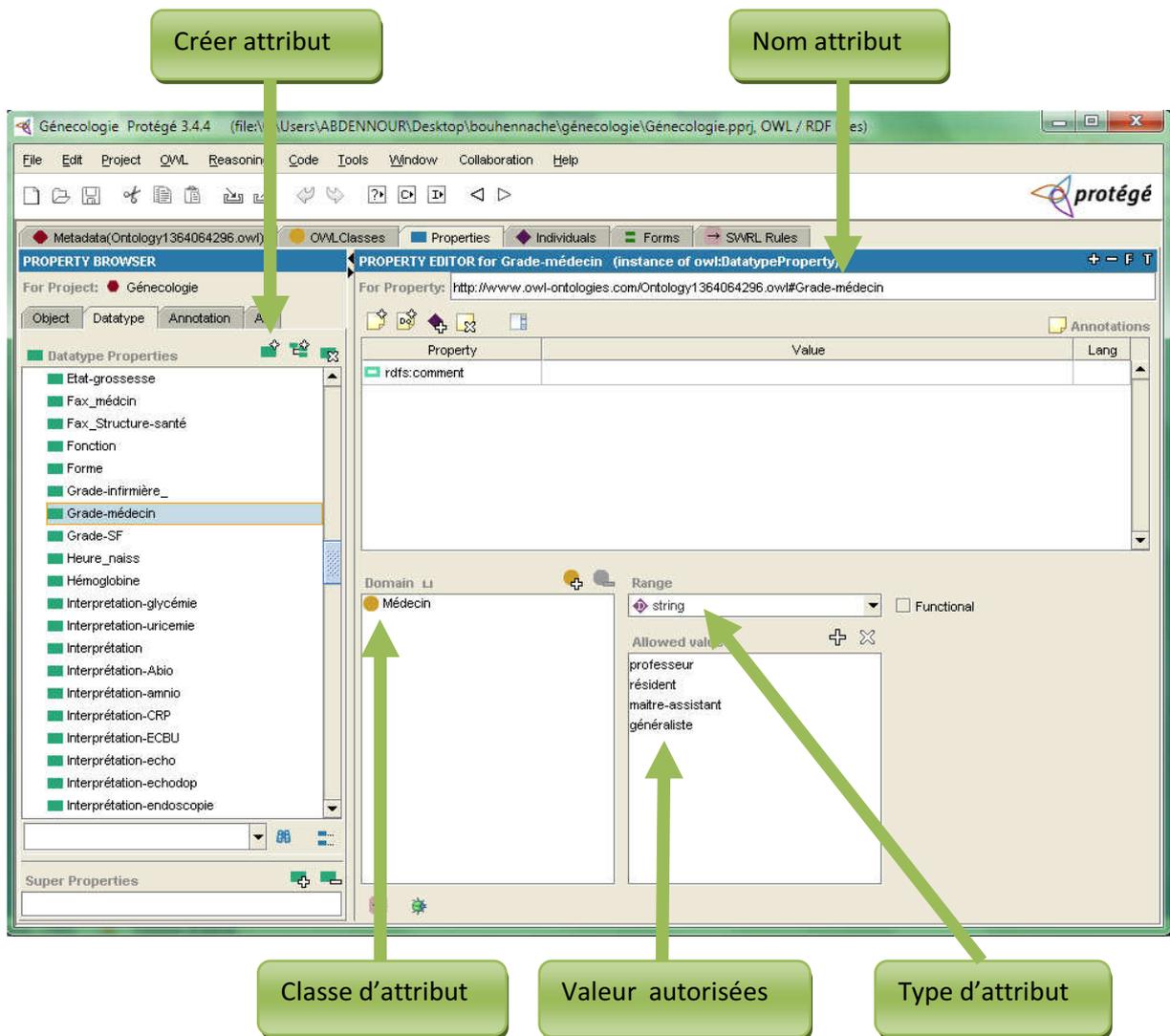


Figure 4.9. Création d'un attribut.

3.2.2. Création des relations (ObjectProperty)

Pour la création de relations, nous devons spécifier le nom de la relation, le domaine, le co-domaine 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 "apaise "

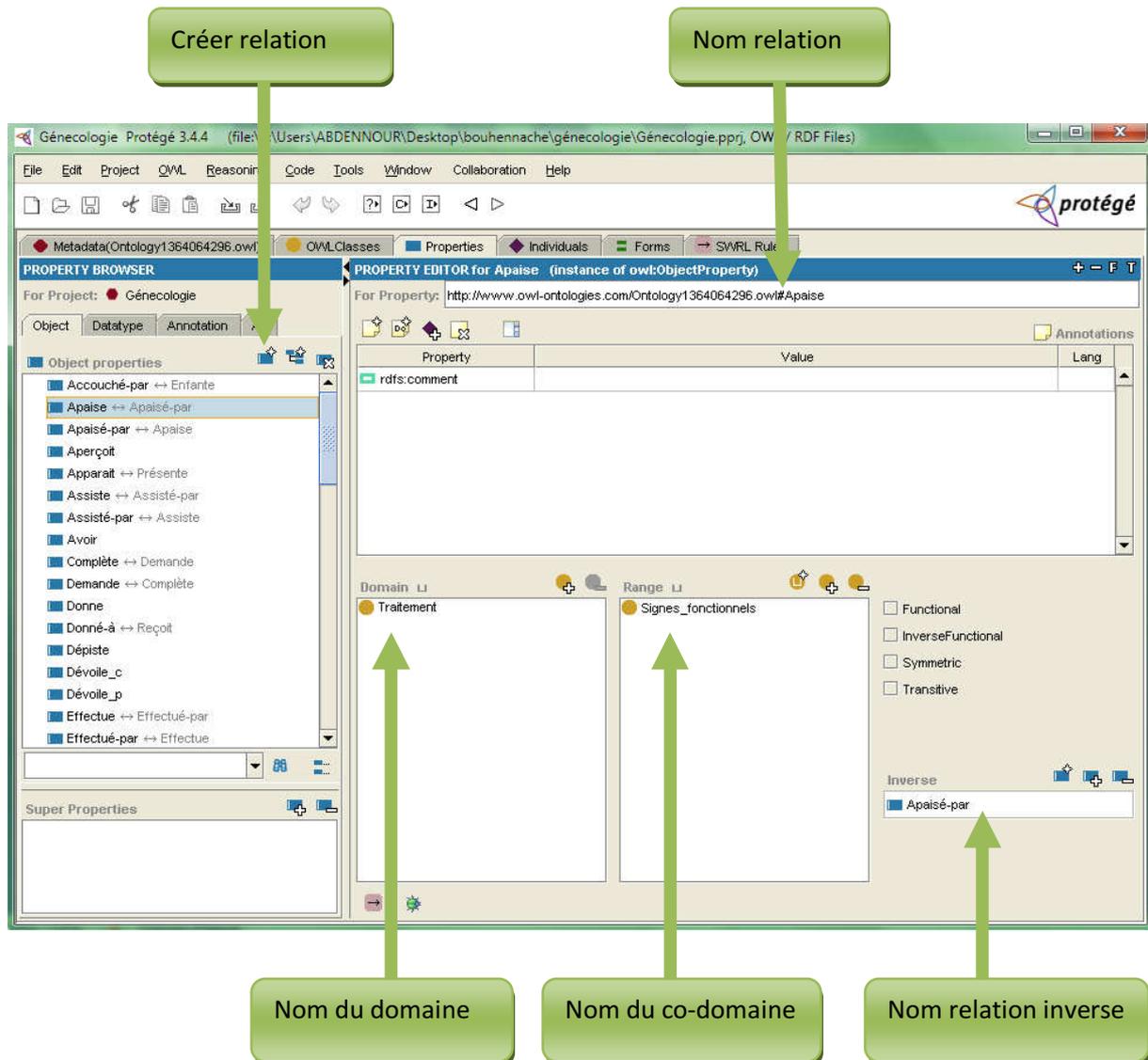


Figure 4.10. Création d'une relation entre deux classes.

3.2.3. 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 (4.11) ci-dessous montre comment créer une restriction.

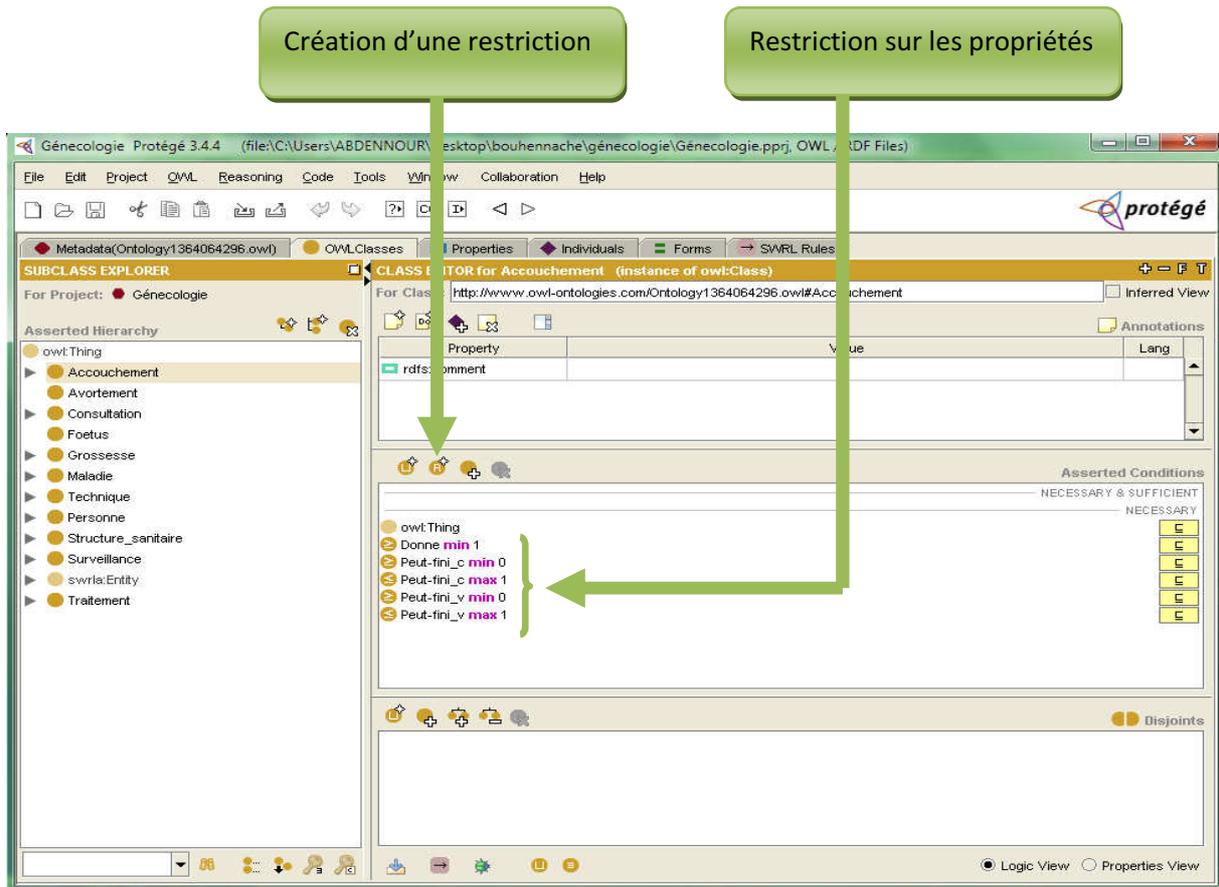


Figure 4.11. Les restrictions créées sur les propriétés d'une classe.

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

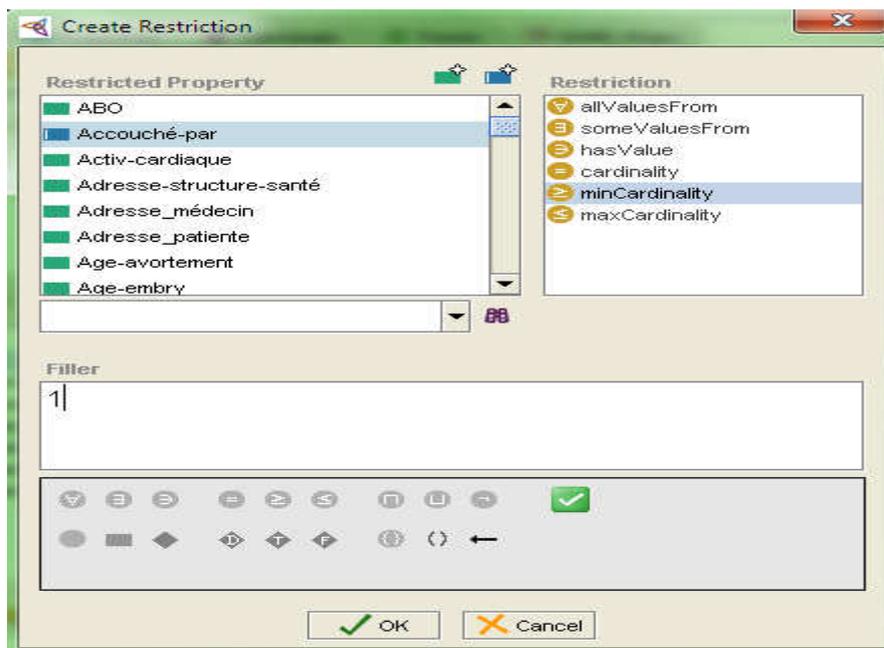


Figure 4.12. Création d'une restriction.

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,...

3.3. 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 4.13 montre la création d'un individu de la classe « Médecin »

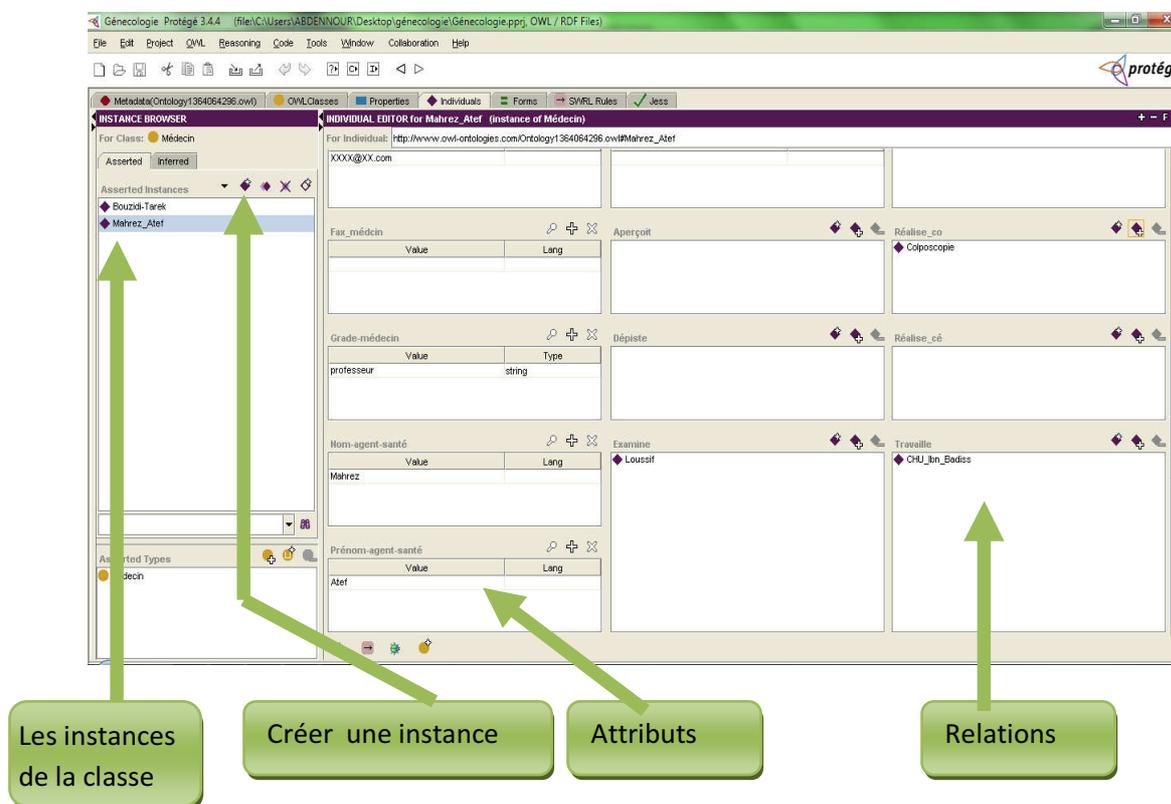


Figure 4.13. Création d'instances.

3.4. 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 4.14 un extrait du code OWL généré par PROTÉGÉ de notre ontologie « GynécOntologie ».

```
...
</owl:Class>
<owl:Classrdf:about="#Curatif">
<rdfs:subClassOfrdf:resource="#Traitement"/>
<rdfs:subClassOf>
<owl:Class>
<owl:unionOfrdf:parseType="Collection">
<owl:Classrdf:ID="Chirurgie"/>
<owl:Classrdf:ID="Cryothérapie"/>
<owl:Classrdf:ID="Célioscopie"/>
<owl:Classrdf:ID="Disposition_inter-utérus"/>
<owl:Classrdf:about="#Electro-coagulant"/>
<owl:Classrdf:about="#Traitement_médicamenteux"/>
<owl:Classrdf:about="#Traitement_rayonnement"/>
</owl:unionOf>
</owl:Class>
</rdfs:subClassOf>
</owl:Class>
<owl:Classrdf:ID="Sage_femme">
.....
```

Figure 4.14. Fragment du code OWL généré par PROTÉGÉ

3.5. 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.

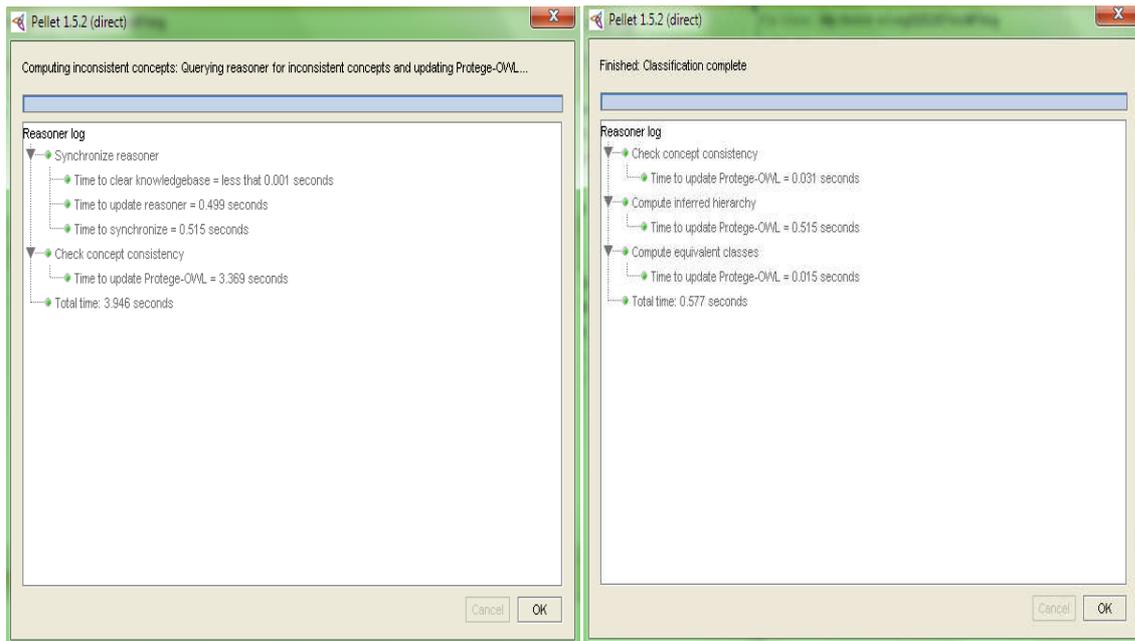


Figure 4.15. Le test de consistance

Figure 4.16. Le test de classification.

4. La représentation sémantique des règles SWRL

4.1. Les règles SWRL

Grace à SWRL Tab, les règles SWRL sont facilement intégrées à l'ontologie OWL. Les règles peuvent être entrées directement du clavier. Cependant, l'utilisateur peut sélectionner les entités OWL directement du panneau des icônes.

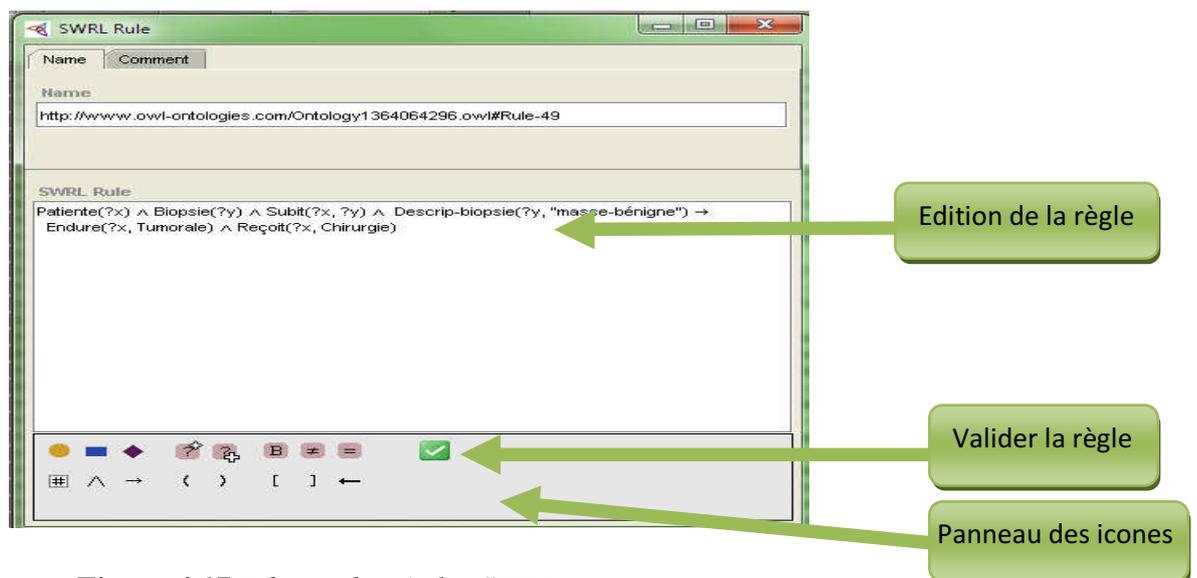


Figure 4.17. Edition des règles SWRL.

Ce qui rend l'édition des règles SWRL interactive, est la présence d'une entité très importante, le panneau des icônes. La figure qui suit donne un aperçu de ce panneau ainsi que ses différents composants.

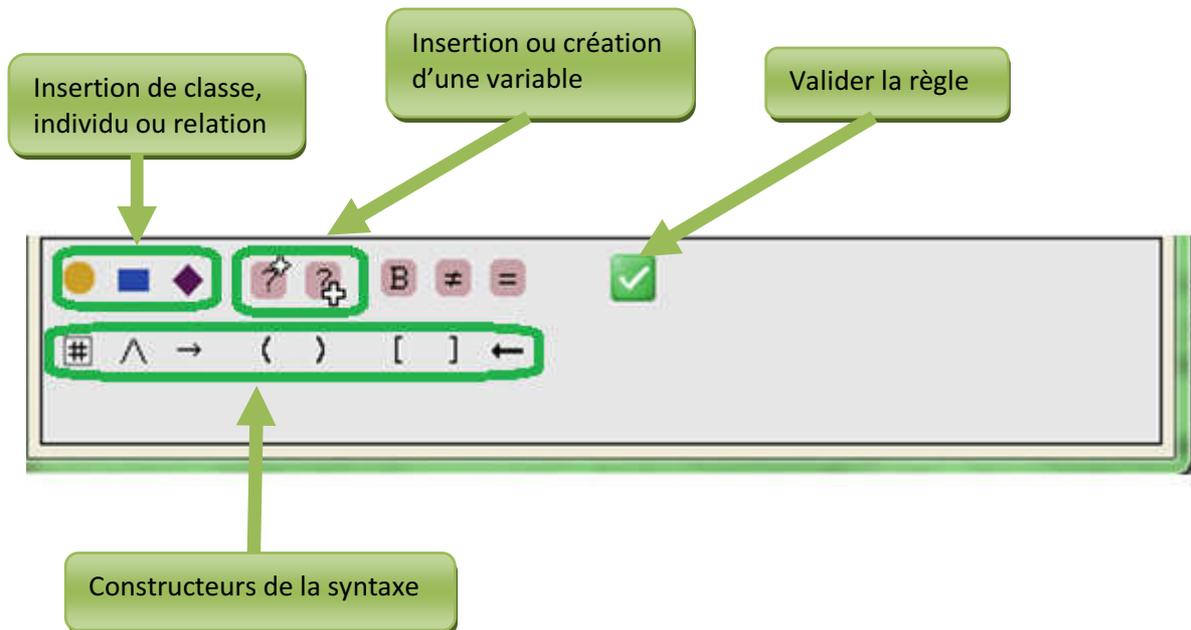


Figure 4.18. Présentation du panneau des icônes.

4.2. Les built-ins

SWRL built-ins sont des prédicats définis par l'utilisateur qui peuvent être utilisés dans les règles SWRL. Les Bibliothèques de SWRL built-ins comprennent des commandes pour construire des règles SWRL comme (swrlb:greaterThan, swrlb:add, ...). Parmi les built-ins utilisés pour l'interrogation de l'ontologie OWL on trouve sqwrl:count, sqwrl:equal, sqwrl:select,

Il existe plusieurs types de built-ins (de comparaison, de mathématiques, ...)

Exemple :

Une règle SWRL utilisant un built-in, qui exprime qu'une patiente qui n'a pas de menstrues à l'âge de 35 ans endure une ménopause précoce s'écrit de la façon suivante :

- $\text{Patiente} (?x) \wedge \text{Régulé} (?x, \text{false}) \wedge \text{swrlb:lessThan}(\text{Age}, 35) \rightarrow$
 $\text{Nom-maladie} (?x, \text{"ménopause précoce"})$

Pour exprimer qu'une patiente âgée de 50 ans et plus, et qu'elle n'a pas eu d'enfants durant sa vie, doit se rendre pour faire une mammographie, est exprimée comme suit :

- $\text{Patiente} (?x) \wedge \text{swrlb:greaterThanOrEqual}(\text{Age}, 50) \wedge \text{swrlb:equal}(\text{Pariété}, 0) \rightarrow$
 $\text{Subit} (?x, \text{Mammographie})$.

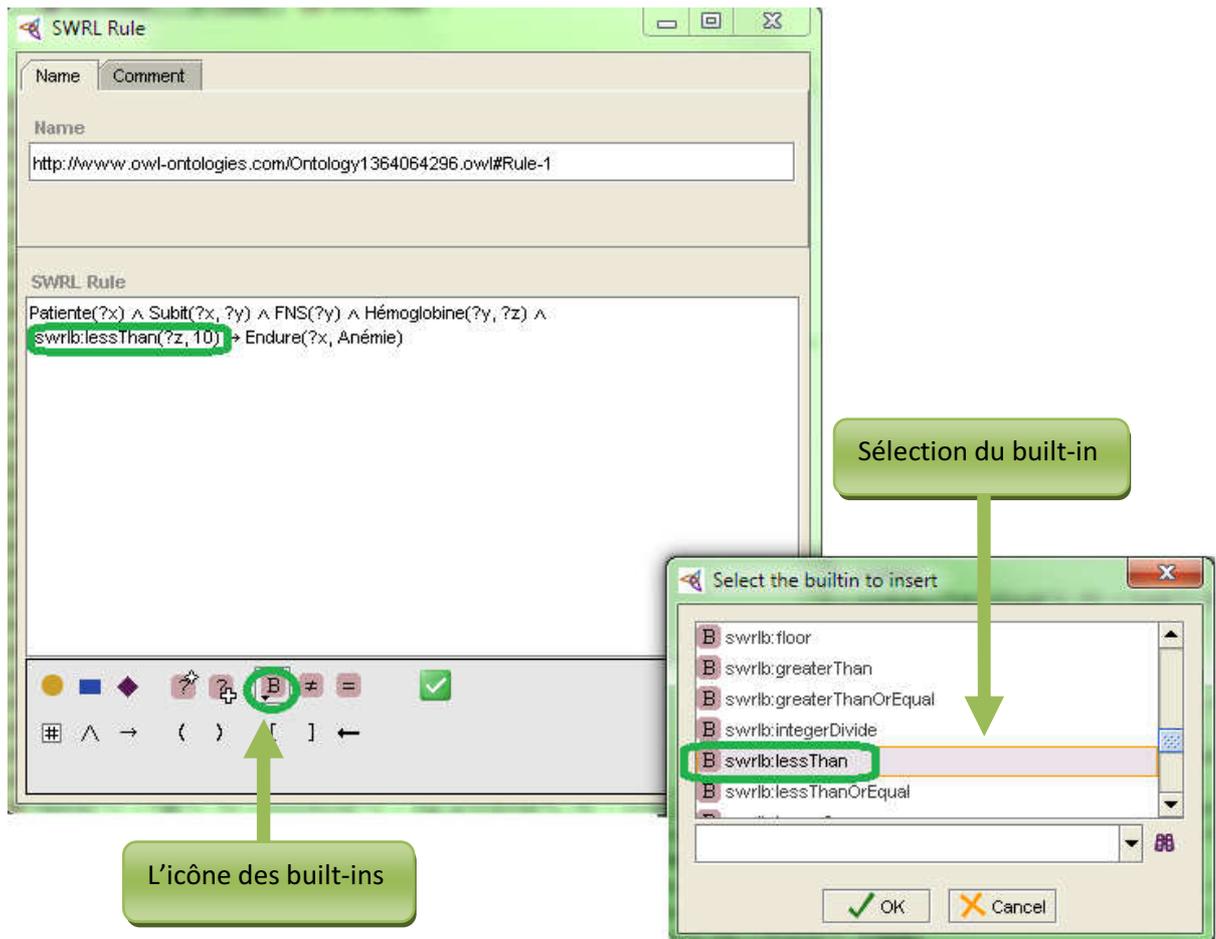


Figure 4.19. Editer une règle SWRL contenant une built-ins.

4.3. Le langage de requête SQWRL

SQWRL (SemanticQueryenhanced Web RuleLanguage) un langage basé sur SWRL pour l'interrogation d'ontologies, fournissant des opérateurs pour la recherche de connaissances sous OWL. Il utilise la bibliothèque de SWRL built-ins comme un point d'extension.

SWRL Tab peut être utilisé pour générer et modifier les requêtes. Pour exécuter la requête, SQWRLQuery Tab doit être ajoutée [50], par ajouter le fichier jar jess 7.1p2 dans le répertoire de plug-ins de protégé. L'opérateur principal est `sqwrl: select`.

Il prend un ou plusieurs arguments qui sont typiquement des variables utilisées dans le corps de la requête.

Par exemple, une requête qui permet de compter le nombre des patientes qui donne naissance à des jumeaux ou plus.

peut-être écrite comme suit:

```
Patiente(?x) ^ Fait(?x, ?y) ^ Grossesse(?y) ^ Nbr-fœtus(?y, ?z)
^ swrlb:greaterThanOrEqual(?z, 2) → sqwrl:count(?x)
```

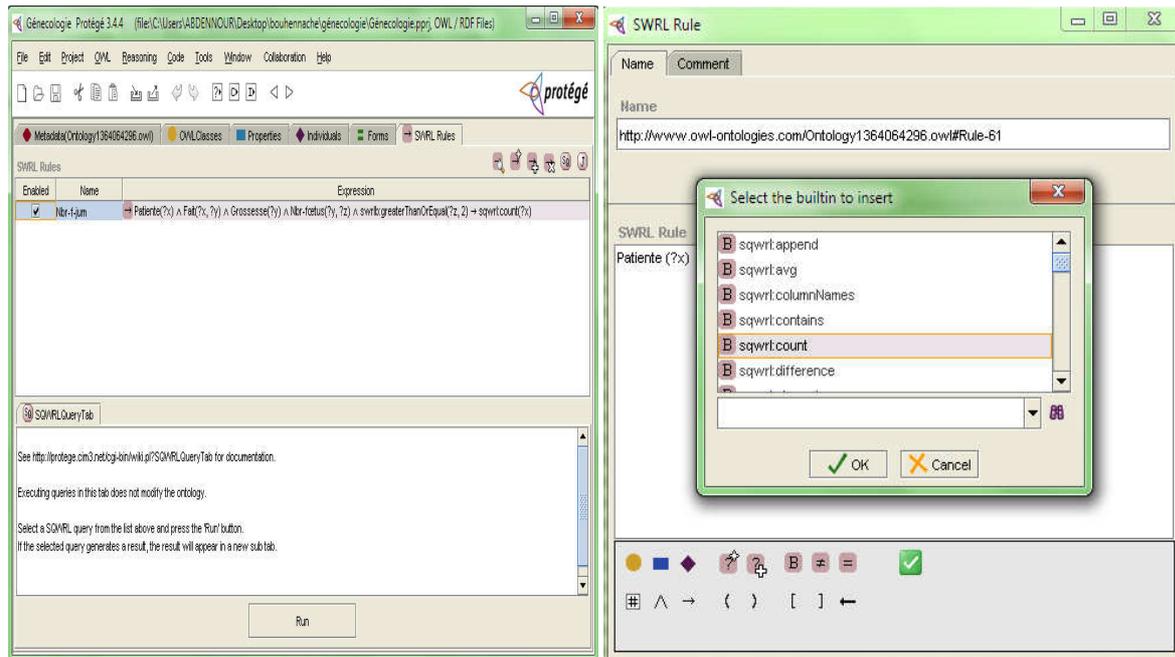


Figure 4.20. SQWRL Query Tab Figure 4.21. Éditer une requête SQWRL

4.4. Exécution des règles

Pour exécuter les règles, il faut sélectionner les règles que nous pouvons exécuter, sachant que nous pouvons exécuter toutes les règles à la fois, comme nous pouvons exécuter juste une partie de ces règles. Après la sélection, nous cliquons sur le bouton « OWL+SWRL → Jess » qui charge l'ontologie OWL et les règles SWRL sélectionnées, après nous lançons l'exécution réelle par le bouton « Run Jess » puis nous constatons l'assertion des nouveaux faits résultant à partir du bouton « Jess → OWL ». La figure suivante représente le déroulement de cette exécution.

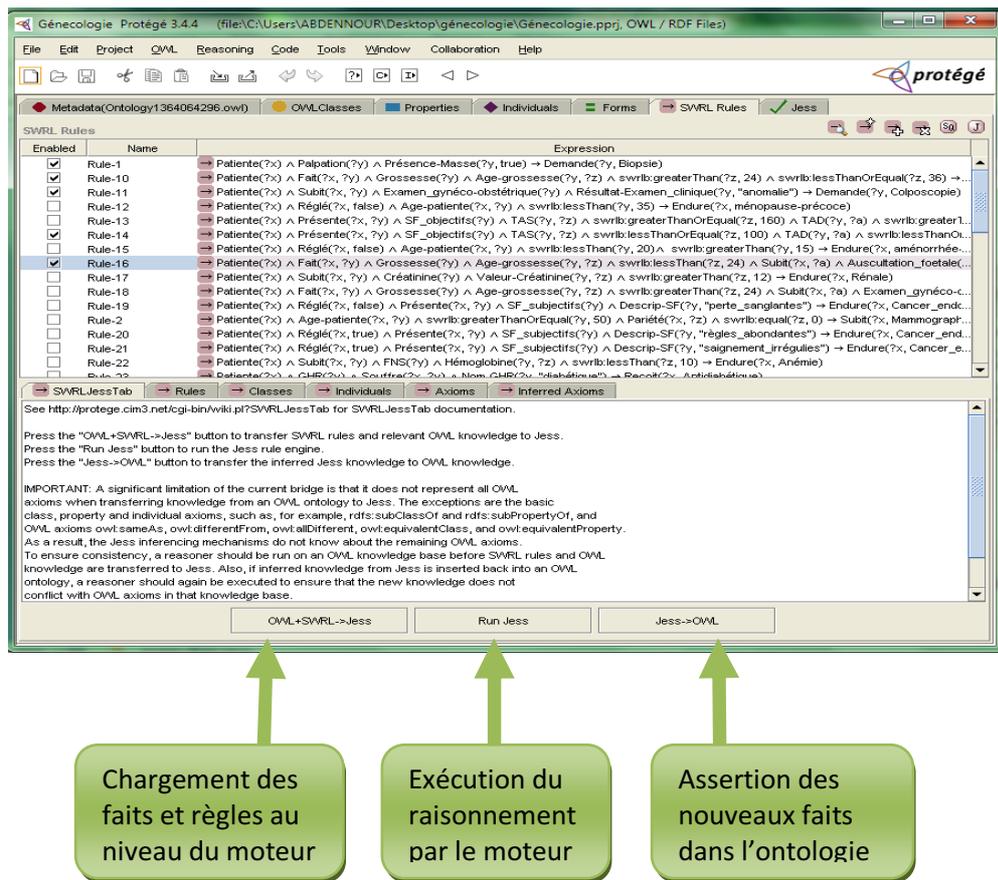


Figure 4.22. Exécution des règles.

4.5. Assertion des nouveaux faits

Pour montrer l'intérêt des règles SWRL, nous prenons l'exemple de cette règle : « Si unepatiente est âgée de moins de 35 ans et souffre d'abondances des règles alors, elle endure uneménopause précoce ».

Nous prenons le cas d'une instance de la classe patiente qui vérifie les conditions de cette règle. Mais OWL seul ne nous permet pas de déduire la maladie dont elle souffre. La figure qui suit donne un aperçu de cette instance :

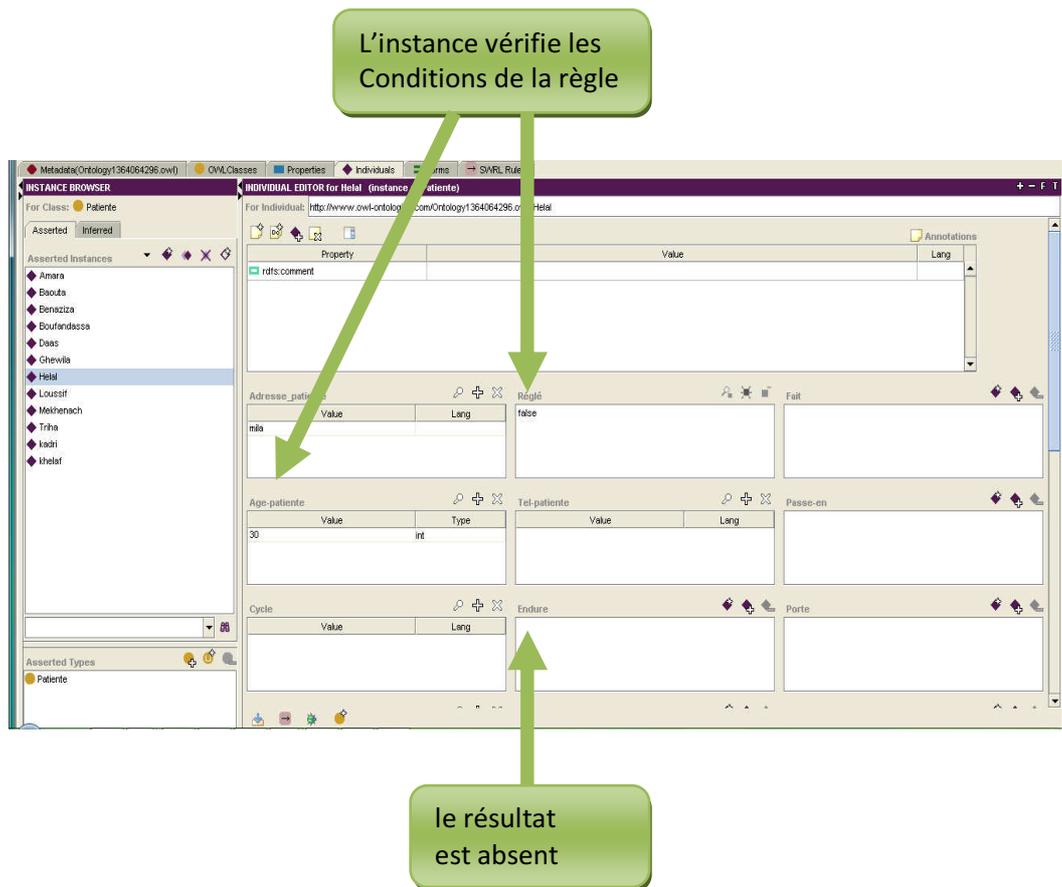


Figure 4.23. Une instance de la classe *Patiente* avant l'exécution de la règle.

En introduisant la règle SWRL, nous ajoutons de l'expressivité à l'ontologie. Parce que cette dernière ajoute une nouvelle assertion qui ne pouvait pas être représentée en OWL. C'est ce que va montrer la figure qui suit, où nous remarquons la présence de la nouvelle assertion dans l'instance.

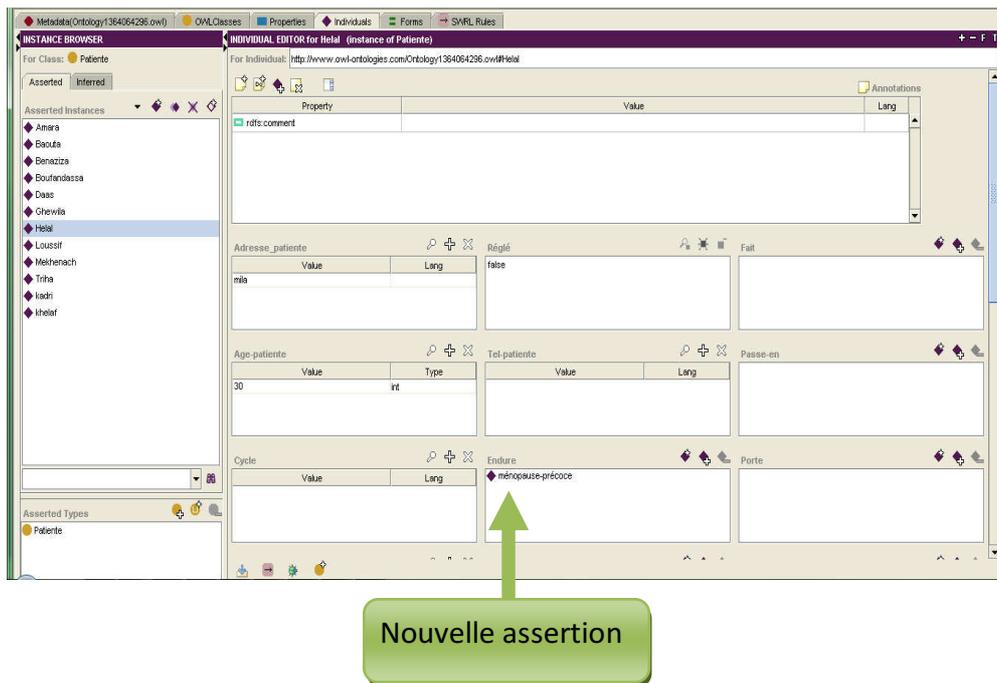


Figure 4.24. Une instance de la classe *Patiente* après l'exécution de la règle.

5. l'interfaceGynomed

5.1. Gynomed

L'application Gynomed est une interface conçue pour faciliter l'accès à l'ontologie "GynécOntologie". Même les noms experts peuvent consulter l'ontologie, et extraire les informations voulues.

Gynomed accepte en entrée les fichiers avec l'extension .OWL et affiche dans le volet Gauche tous les concepts présente dans l'ontologie, avec leurs informations complètes (attributs, instances, relations,...etc.).

Aussi, il permet d'afficher graphiquement les hiérarchies de l'ontologie concept par concept.

5.2. Outils pour construire l'interface « Gynomed »

Autres que les outils d'implémentation et d'édition d'ontologies, nous pouvons trouver aussi les outils permettant de construire des applications basées sur les ontologies, ils fournissent également un environnement de programmation pour RDF, RDFS et OWL. Parmi ces outils nous pouvons citer langage java et plateformeJena.

5.2.1. 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...).

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.

5.2.2. 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.

5.2.3. NetBeans

NetBeans IDE(*integrateddevelopmentenvironment*) est un environnement de développement intégrer (On peut y écrire, compile 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 ajouterde la bibliothèque pour construire des applications, dans notre cas en ajoute la bibliothèque Jena.

La figure suivante montre notre projet avec la bibliothèque Jena.

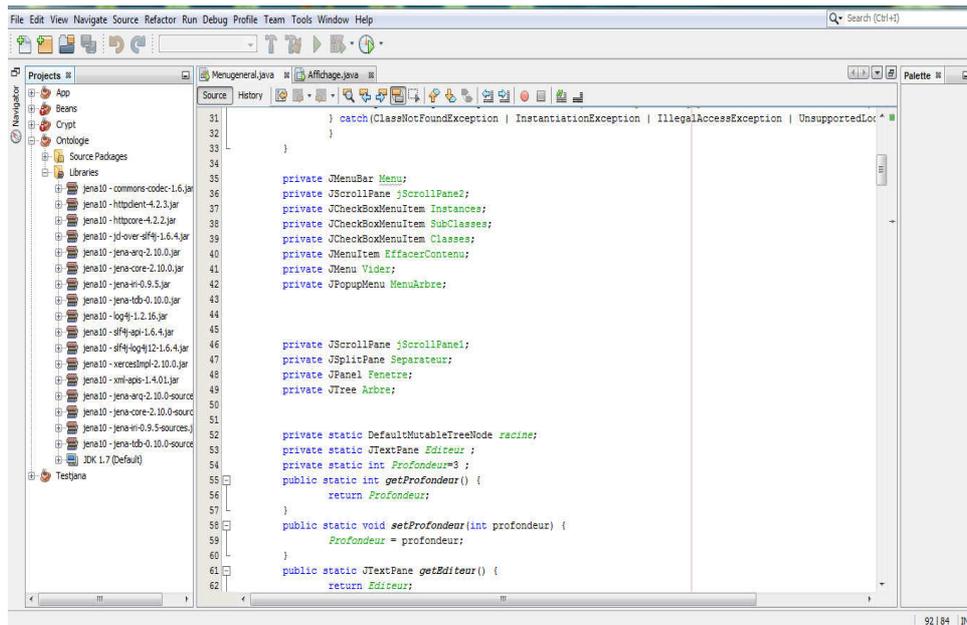


Figure 4.25. Le projet Ontologie avec la bibliothèque Jena.

6. Interrogation de la base de connaissance

L'interface d'authentification permet à l'utilisateur d'accéder à son application par le biais de son login et son mot de passe. Comme montre la figure suivante

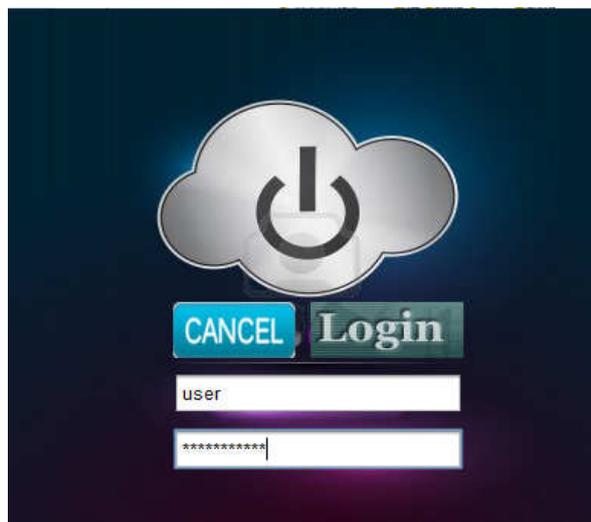


Figure 4.26. Interface authentification.

Une ontologie valide est prête d'être chargée par le système pour être interrogée. Une fois l'application est active, l'interface présentée par la figure 4.21 est affichée.



Figure 4.27. Interface de l'application.

Dans ce qui suit, nous présenterons les étapes pour charger un fichier de notre ontologie puis affiche la hiérarchie de notre ontologie et affiche les informations tel que attributs, relations, ... etc. Concernent le concept choisi.

Pour ouvrir un fichier OWL Il suffit de cliquer sur le menu fichier et de choisir le menu Ouvrir.

La boîte suivante apparaît à l'écran

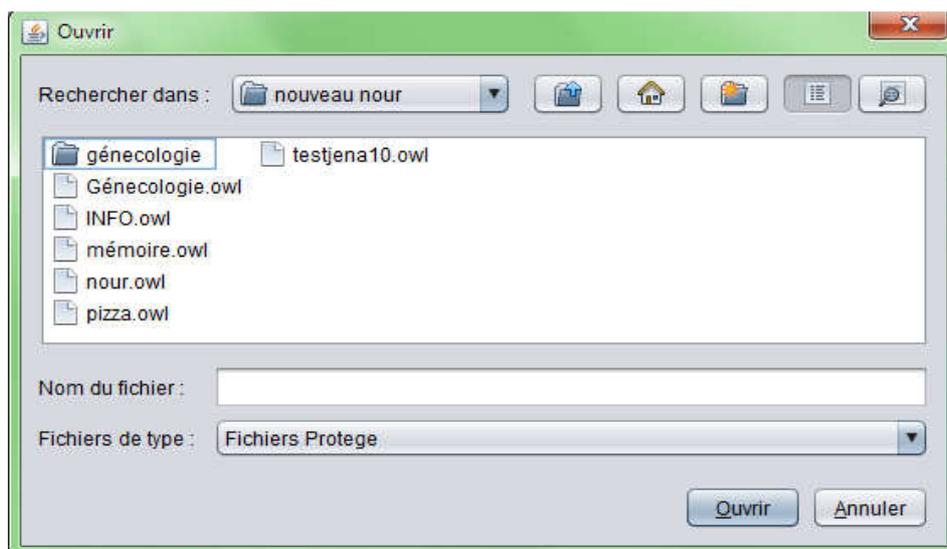


Figure 4.28. Boîte de dialogue pour ouvrir fichier OWL.

L'application fournit un filtre par défaut pour n'afficher que les fichiers avec l'extension OWL. Vous pouvez bien sûr le changer. Dans ce cas le logiciel affichera tous les fichiers.

Maintenant il ne reste plus qu'à choisir le fichier voulu et cliquer sur ouvrir.

Le temps de chargement du fichier est fonction de la dimension de l'ontologie. Cela peut prendre plusieurs dizaines de secondes. La figure suivante illustre l'affiche des informations.

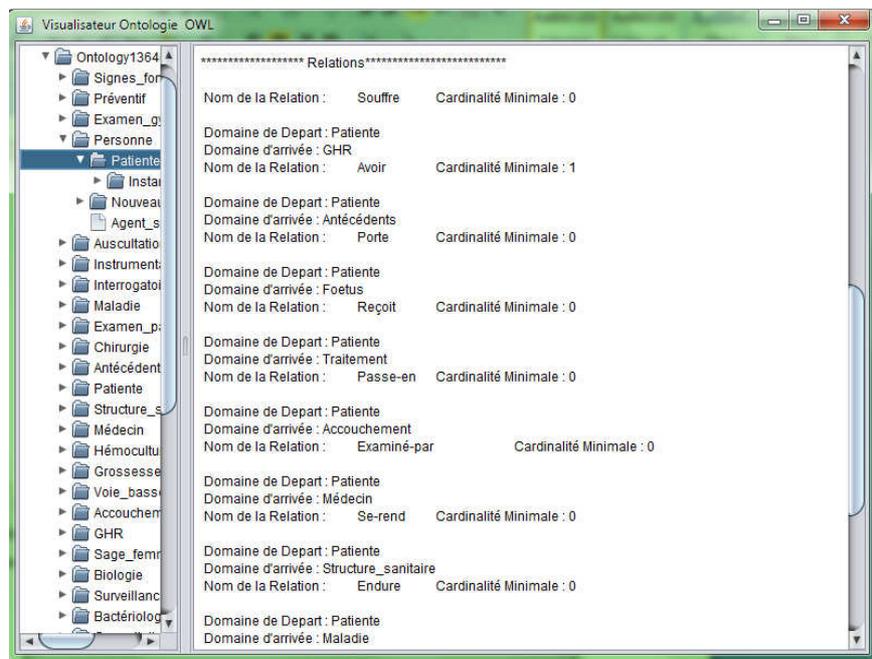


Figure 4.29. Affiche les informations.

Dans le menu affichage nous pouvons afficher la hiérarchie de l'ontologie par concepts avec l'affichage des informations sur ce concept. Comme montre la figure 4.30

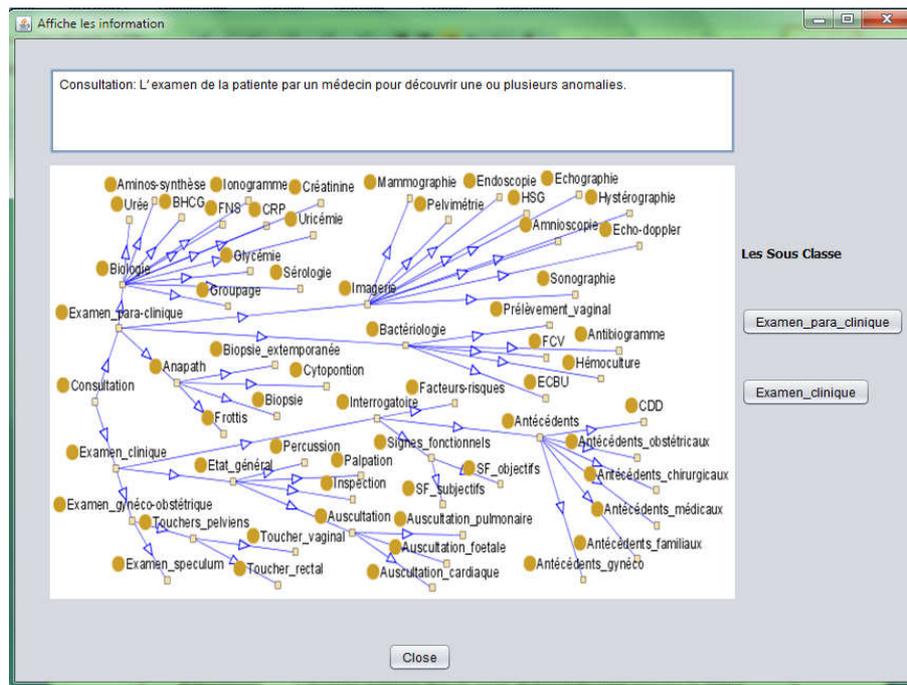


Figure 4.30. Affiche la hiérarchie d'ontologie.

7. Conclusion

Dans ce chapitre, nous avons implémenté l'ontologie médicale «GynécOntologie». Cela consiste à sa codification dans un langage du web sémantique OWL-DL.

Le langage OWL nous a 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.

Pour enrichir notre Ontologie médicale «GynécOntologie»avec des règles SWRL, nous avons intégré les règles SWRL qui ont apportées une hautes expressivité, nous l'avons édité à travers le plug-in SWRL Tab qui fournit une interface très interactive pour l'édition des règles prenant en charge l'ensemble des fonctionnalités du langage SWRL.

Le plug-in SWRL JESS TABprend en entrée l'ontologie OWL et les règles SWRL pour en fin ajouter de nouvelles assertions à la base de faits, en utilisant le moteur *Jess*.

Et finalement, nous avons présenté les outils utilisés dans le développment de notre application « Gynomed », cette dernière qui permet de représenter toutes les informations de l'ontologie « GynécOntologie », et afficher la hiérarchie de chaque concept avec ses informations correspondantes.

Conclusion général et perspectives

1. Conclusion générale

Un langage d'ontologie ayant une sémantique formelle est nécessaire pour aller vers un websémantique. Ce langage doit offrir d'une part un pouvoir d'expression suffisant pour représenter intelligemment de grandes quantités de connaissances, et d'autre part des mécanismes efficaces pour raisonner sur ces ontologies : classification automatique, vérification formelle de cohérence, services de réponse à des requêtes, ...etc. De ce fait, OWL est un bon candidat, mais à condition d'être renforcé par une couche de règles.

SWRL est un langage qui enrichit la sémantique d'une ontologie définie en OWL. Il ajoute une très haute expressivité au langage OWL, et assure l'expression des entités qu'OWL seul ne permet pas d'exprimer.

Dans ce mémoire, nous avons développé une ontologie hybride dans le domaine médical. Nous avons suivi un processus de construction d'une ontologie, inspiré des différentes phases proposées par la méthode METHONTOLOGY et également de l'exploitation d'autres travaux, tels que le processus proposé dans [46], afin d'atteindre un ensemble de représentations intermédiaires qui facilite sa formalisation ultérieure et cela en adoptant l'approche basée sur la logique de descriptions.

L'intérêt de ce formalisme est qu'il est d'une part suffisamment simple pour que des non spécialistes puissent l'utiliser et, d'autre part, qu'il est plus expressif que les graphes conceptuels et les frames. Basé sur cette formalisation, nous avons choisi le langage OWL pour codifier l'ontologie formelle, et utiliser l'éditeur graphique PROTÉGÉ-OWL, afin de guider l'implémentation et de produire un document OWL. Par ailleurs, pour vérifier et filtrer l'ontologie OWL au cours du processus de développement, nous utilisons le système Pellet. Ce dernier, fournit un support de raisonnement en traduisant des expressions OWL à des expressions de la logique de description. Les services d'inférence fournis par pellet incluent le test de satisfiabilité d'un concept et le test de subsumption. Le plugin SWRL Tab était utilisé pour éditer les règles SWRL. SWRL Jess Tab assurant la liaison entre PROTÉGÉ-OWL et le moteur de règles Jess, permet l'exécution des règles, et l'affichage des données inférées. Finalement nous construirons une application pour faciliter l'accès aux les informations de notre ontologie.

2. Perspectives

L'architecture proposée pour le raisonnement sémantique sur une ontologie hybride OWL enrichie par les règles SWRL afin de modéliser le service gynécologie du domaine médicale non efficace. Nous nous sommes limité par l'utilisation de deux raisonneurs différents : un pour raisonner sur l'ontologie OWL (pellet), l'autre sur les règles SWRL (Jess). De ce fait, plusieurs limites et perspectives se dégagent de ce mémoire. Nous citons dans la suite, celles que nous croyons les plus importantes.

- **L'introduction de la logique floue**

Les informations issues du domaine médicale ne sont pas spécifiées exactes. Donc il faut faire appel à la logique floue. De ce fait, un nouveau langage pour les ontologies floues doit voir le jour, et l'intégration de nouveaux constructeurs au langage OWL est nécessaire pour pouvoir modéliser les informations imprécises et floues.

- **L'intégration des ontologies**

Notre Ontologie GynécOntologie ne retrouve pas des informations concernant par exemple l'anatomie du corps humain, or pour diagnostiquer une maladie, nous avons besoin de la localiser en premier lieu. Donc, une intégration de GynécOntologie avec une ontologie d'anatomie est nécessaire pour rendre GynécOntologie opérationnelle.

- **Evolution de l'application**

Pour rendre l'utilisateur capable d'introduire les informations directement à travers l'application sans avoir besoin d'accéder à l'éditeur graphique PROTEGE-OWL, aussi nous envisageons d'intégrer les interrogations de la base de connaissance via l'application Gynomed avec le logiciel "SPARQL".

Bibliographies

- [1] G. Antoniou, C. V. Damasio, B. Grosz, I. Horrocks, M. Kifer, J. Maluszynski and P. F. Patel-Schneider. Combining Rules and Ontologies: A survey. Technical Report IST506779/Linkoping/I3-D3/D/PU/a1, Linkoping University, February 2005. IST-2004 506779 REVERSE Deliverable I3-D3. <http://reverse.net/publications/>.
- [2] A. Napoli. Une introduction aux logiques de description. Technical report, INRIA, 1997.
- [3] B. Motik, I. Horrocks, R. Rosati and U. Sattler. “Can OWL and Logic Programming Live Together Happily Ever After?”. In Proceedings of the International Semantic Web Conference (pp. 501-514). Springer, 2006.
- [4] P. F. Patel-Schneider and I. Horrocks. “Position Paper: A Comparison of Two Modeling Paradigms in the Semantic Web”. In Proceedings of the International Conference on World Wide Web (pp 3-12). ACM Press, 2006.
- [05]: T. Berners-Lee, J. Hendler et O. Lasilla, The Semantic Web. Scientific American, 284 (2001).
- [06] : Hanoï, Ontologies et Web services – Rapport du travail d'intérêt personnel encadré, http://www..com/ tipe-phan_quang_trung_tien, juillet 2005.
- [07]: A. Rector. Thesauri and formal classifications: Terminologies for people and machines. Methods of Information in Medicine, 37(4_5), 501_509, 1998.
- [8] T. Eiter, G. Ianni, A. Polleres, R. Schindlauer and H. Tompits. “Reasoning with rules and ontologies”. In Reasoning Web 2006, volume 4126 of Lecture Notes in Computer Science, pages 93–127. Springer, September 2006.
- [09] N. Aussenac-Gilles et 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.
- [10]: S. Handschuh, S. Staab and A. Maedche. “CREAM - Creating relational metadata with a component-based, ontology-driven annotation framework”. In Proceedings of the Knowledge Capture Conference (KCAP'01), Banff, Canada, pp. 76-83, 2001.
- [11] T. R. Gruber. “A translation approach to portable ontology specifications”. Knowledge Acquisition, 5(2): 199–220, 1993.
- [12] T-R. Gruber. “Towards Principles for the Design of Ontologies Used for Knowledge Sharing”. International Journal Human-Computer Studies, 1995.

Bibliographies

[14] B. Bachimont, A[13] Hernandez, N., Ontologies de domaine pour la modélisation du contexte en recherche d'information. Thèse de doctorat, Université de Toulouse, 2005.

arts 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.

[15] 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.

[16] M.Uschold&M.Grüninger, "ONTOLOGIES: Principles, Methods and Applications". Knowledge Engineering Review. 1996

[17]: Sowa, J.F. "Conceptual structures: information processing in mind and machine", Addison-Wesley, 1984.

[18]: M. Buchheit, f. Donini, and A. Shaerf" Decidable reasoning in terminological knowledge representation systems". Journal of artificial intelligence research,1: 109-138, 1993.

[19] : A. Napoli, " une introduction aux logiques de descriptions" N° 3314, Décembre 1997.

[20]: I. Horrocks "DAML+OIL : A Description Logic for the Semantic Web". IEEE Data Engineering Bulletin Num 1, vol.25, pp 4-9, 2002

[21] O. M. Drews "Raisonnement classificatoire dans une représentation à objets multipoints de vue". Thèse de doctorat, Université Joseph Fourier, Grenoble. pp 30-51. Mars 1993.

[22] I. Horrocks, F. Baader, & U. SattlerL. "Description Logics as Ontology Languages for the Semantic Web". In Festschrift in honor of Jorg Siekmann, LNAI.Springer-Verlag. 2003.

[23] <http://www.w3.org/RDF/>.

[24] G. V. Heijst, A. Th. Schreiber, B.J. Wielinga, Using explicit ontologies in KBS development. Int. J. Human.-Computer Studies. (1997)

[25] R. Mizoguchi, M. Ikeda. "Towards ontology engineering.In The Joint 1997 Pacific Asian Conference on Expert systems" - International Conference on Intelligent Systems, p. 259-266, Singapore, 1995.

[26] N. Guarino. "Formal ontology in information systems". In the 1st International Conference on Formal Ontology in Information Systems (FOIS), p. 3-15, 1998.

[27] F.Frédéric, Travail de diplôme 2007, "Filière informatique de gestion, Web 3.0 : Interrogation intelligente", 2007.

[28] 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 Artificial

Bibliographies

Intelligence, 2001.

- [29] 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.
- [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] E. Sirin, B. Parsia, B.C. Grau, A. Kalyanpur and Y.Katz, Pellet: A Practical OWL-DL Reasoner, University of Maryland, 2005.
- [32] P. F. Patel-Schneider and I. Horrocks (2006). Position Paper: A Comparison of Two Modelling Paradigms in the Semantic Web. In Proceedings of the International Conference on World Wide Web (pp 3-12). ACM Press.
- [33] I. Horrocks, L. Li, D. Turi and S. Bechhofer (2004a). The Instance Store: Description Logic Reasoning with Large Numbers of Individuals. In Proceedings of the Workshop on Description Logics. Vol. 104 (pp. 31-40). CEUR-WS.
- [34] V. Haarslev, R. Moller (1999). An Empirical Evaluation of Optimization Strategies for ABox Reasoning in Expressive Description Logics. In Proceedings of the International Workshop on Description Logics. Vol. 22. CEUR-WS.
- [35] T. Eiter, T. Lukasiewicz, R. Schindlauer, H. Tompits, “Combining answer set programming with description logics for the semantic web”. In Proc. of the International Conference of Knowledge Representation and Reasoning (KR04), 2004.
- [36] M. F. Donini, M. Lenzerini, D. Nardi, A. Schaerf, AL-log: “Integrating Datalog and Description Logics”, Journal of Intelligent Information Systems, (pp. 227-252), 1998.
- [37] A. Levy, M. Rousset, CARIN: “A representation language combining Horn rules and description logics”. Artificial Intelligence,(pp. 165 – 209), 1998.
- [38] I. Horrocks, PF. Patel-Schneider, H. Boley, S. Tabet and B. Grosz SWRL: “A Semantic Web Rule Language Combining OWL and RuleML”. W3C, May 21, 2004.
- [39] I. Horrocks, PF. Patel-Schneider, H. Boley, S. Tabet and B. Grosz SWRL: “A Semantic Web Rule Language Combining OWL and RuleML”. W3C, May 21, 2004.
- [40] V. Papataxiarhis, Technologies Knowledge Representation and Inference: The Case of an Intelligent Assistant Coach football. MageserThses . Department of Informatics and Telecommunications, University of Athens – Greece, (2009)
- [41] J. Mei, E. P. Bontas, “Reasoning Paradigms for SWRL-enabled Ontologies”, Protégé With Rules Workshop, Madrid, 2005.

Bibliographies

- [42] KAON2 – Ontology Management for the Semantic Web, (2008).
<http://kaon2.semanticweb.org>.
- [43] R. Mizoguchi, M. Ikeda (1997). Towards ontology engineering. In The Joint 1997 Pacific Asian Conference on Expert systems - International Conference on Intelligent Systems, p. 259–266, Singapore.
- [44] C. Golbreich, Combining Rule and Ontology Reasoners for the Semantic Web, Invited talk, Rules and Rule Markup Languages for the Semantic Web, Hiroshima, Japan, Grigoris Antoniou, Harold Boley Editors, LNCS 3323, Springer, 2004.
- [45] M. Hemam. Un processus de developpement d'ontologie dans le cadre du web sémantique, Centre Universitaire Larbi Ben M'hidi -Oum El Bouaghi- Institut des sciences exactes (2005).
- [46] F. Afri. ‘‘Raisonnement sur une ontologie pour la recherche d’informations médicales’’. Mémoire de magister, Ecole doctorale réseaux et systèmes distribués, Université Abderrahmane Mira -Bejaia-Faculté des sciences exactes, 2009.
- [47] S. Bouaroudj, ‘‘Raisonnement sur une ontologie enrichie par des règles SWRL pour la recherche sémantique d’images annotées’’. Mémoire de magister, Ecole doctorale en informatique de l’Est Pole ANNABA, 2010.
- [48] F. Natalya, La création de votre première ontologie. Université de Stanford, CA.
- [49] 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
- [50] T. Eiter, G. Ianni, A. Polleres, R. Schindlauer and H. Tompits. ‘‘Reasoning with rules and ontologies’’. In Reasoning Web 2006, volume 4126 of Lecture Notes in Computer Science, pages 93–127. Springer, September 2006.

Glossaire

ABO	Groupe sanguins
Activ-cardi	Activité cardiaque
Age-embry	Age embryonnaire
ATCD	Antécédent
ATCD-chir	Antécédents chirurgicaux
ATCD-med	Antécédents médicaux
ATCD-gyn	Antécédents gynécologiques
Aus	Auscultation
BHCG	Hormone gonadotrophine chorionique humaine
CDD	Circonstances de découverte
CRP	Créactive protéine
Date-accouch	Date d'accouchement
DDR	Date des dernières règles
Dil-col	Dilatation du col de l'utérus
DIU	dispositif intra-utérin
ECBU	Examen cyto bactériologiques des urines
E .Speculum	Examen Speculum
Examen-G/O	Examen gynéco/obstétrical
Fcls	Fonctionnels
FNS	Numération formule sanguine
F.risque	Facteur risque
GHR	Grossesse à haut risque
HCG	Gonadotrophine chorionique humaine
HSG	Hystérosalpingographie
HTA	Hypertension artérielle

Glossaire

Long-col	Longueur du col de l'utérus
MFA	Mouvements foetal actifs
Obst	Obstétrique
Perim-cranien	Périmètre crânien
Signes-fcls	Signes fonctionnels
SFO	Signes fonctionnels objectifs
SFS	Signes fonctionnels subjectifs
TAD	Tension artérielle diastolique
TAS	Tension artérielle systolique
Touché-P	Toucher pelviens
TR	Toucher rectal
TV	Toucher vaginal
TSS	Technicien supérieur de la santé
TV	Toucher vaginal
V. basse	voie basse