

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

**Université Saad Dahlab Blida**

N° D'ordre :.....



Faculté des sciences

**Département d'informatique**

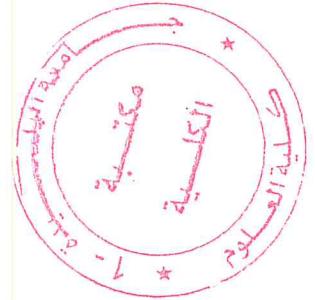
Mémoire Présenté par :

**Belmebarki Mohamed Amine      Boukhari Bilal**

**En vue d'obtenir le diplôme de master**

**Domaine : Mathématique et informatique**

**Filière : Informatique  
Spécialité : Informatique  
Option : Ingénierie de logiciel**



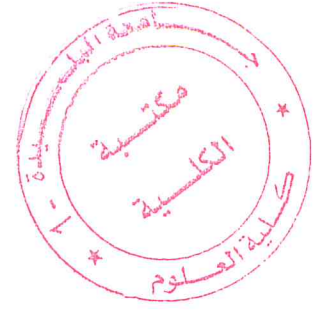
***Thème : Modélisation de l'incertitude du web sémantique, en utilisant les réseaux bayésiens orienté objets.***

**Soutenu le :**

Mme.  
Mme.  
Mme. Fareh  
M. Riali

Président  
Examinateur  
Promotrice  
Encadreur

**Promotion  
2016 / 2017**



## **Remerciement**

*Avant de présenter notre travail, il est de notre devoir d'exprimer nos remerciements, tout d'abord à Allah tout-puissant qui nous a donné la force pour réaliser ce travail et puis nos parents qui nous ont été d'un grand soutien pendant toute la période de nos études et dès le début de ce travail.*

*Nous somme profondément reconnaissant à notre promotrice, Mm Fareh et Mr Riali pour le temps qu'ils ont investi pour suivre et analyser les résultats de ce travail ainsi que pour leurs conseils qu'ils n'ont cessé de nous prodiguer.*

*Un énorme remerciement à nos parents, nos sœurs et frères pour leurs encouragements.*

*Belmebaraki Mohamed Amine*

**BOUKHARI BILAL**

## *Dédicace*

*Avec un énorme plaisir, un cœur ouvert et une immense joie, que je dédie mon travail à mes très chers, respectueux et magnifiques parents qui m'ont soutenu tout au long de ma vie ainsi à mes sœurs, mes frères et à toute ma famille.*

*Et mes amis en particulier à Echiker Amine et Belafia Mouad.*

*A toutes personnes qui m'ont encouragé ou aidé au long de mes études.*

*Belmebarki Mohamed Amine*

*BOUKHARI BILAL*

**Résumé :**

Les ontologies sont au cœur du web sémantique, elles permettent de représenter les connaissances du web, cette représentation qui regroupe un ensemble de concepts et de relations décrivant un domaine particulier. Les ontologies sont vraiment puissantes, mais leurs principaux défauts c'est leur incapacité de représenter et de raisonner sur l'incertitude, d'où la modélisation des connaissances du web sémantique.

Nous proposons dans ce mémoire une nouvelle extension de l'ontologie qui est capable de représenter et raisonner l'incertitude qui existe dans le web sémantique. Et pour cela nous utilisons les réseaux bayésiens orientés objets qui permettent d'exprimer l'incertitude qui manque dans les ontologies. Nous avons aussi détaillé comment construire ce réseau bayésien orienté objet et l'intégrer dans une ontologie classique pour obtenir une ontologie probabiliste, et nous avons effectué l'inférence dans ces réseaux bayésiens orientés objet.

**Mots clé:** OWL, ontologies, réseaux bayésiens, réseau bayésien orienté objet, classes, nœuds.

**Abstract:**

The ontologies are at the heart of the semantic web, they allow to represent the knowledge of the web. An ontology is a representation that groups together a set of concepts and relationships describing a particular domain. They are really powerful, but their main drawback is their inability to represent and reason on uncertainty, hence the modeling of semantic web knowledge.

We propose in this paper an ontology that is able to represent and reason the uncertainty that exists in the semantic web. And for this we use the object-oriented bayesian networks which allow to express the uncertainty that is lacking in the ontologies. Furthermore, we specify how to build this RBOO and integrate it into a classical ontology to obtain a probabilistic ontology. We made the inference in these object oriented Bayesian networks.

**Keywords:** OWL, ontologies, Bayesian networks, object oriented bayesian network, classes, nodes.

## ملخص:

الانطولوجيا هي قلب الويب الدلالي وأنها يمكن أن تمثل المعرفة على شبكة الإنترنت. والانطولوجيا هو تمثيل يتضمن مجموعة من المفاهيم والعلاقات واصفا منطقة معينة. فهي قوية حقا ولكن عيوب الرئيسي هو عدم قدرتهم على تمثيل الغموض و عدم التأكد و منه عدم قدرتهم على نمذجة المعرفة على شبكة الإنترنت الدلالي. ونقترح في هذه الورقة الانطولوجيا التي هي قادرة على تمثيل عدم اليقين موجود في الويب الدلالي. ولهذا نستخدم شبكات النظرية الافتراضية وجوه المنحى التي تسمح التعبير عن عدم اليقين المفقودة في الانطولوجيا. و أيضا دمجها في الانطولوجيا الكلاسيكية لعلم الوجود الاحتمالي. أجرينا الاستدلال في شبكات النظرية الافتراضية وجوه المنحى بالتفصيل كيفية بناء هذا التمثيل من هذا القبيل.

الكلمات الرئيسية : الانطولوجيا , شبكات النظرية الافتراضية , شبكات النظرية الافتراضية وجوه المنحى , عدم اليقين , الاستدلال.

## Sommaire

1. Introduction générale .....	1
1.1. Contexte du travail .....	1
1.2. Problématique.....	1
1.3. Objectif.....	2

### Chapitre1 : Le web sémantique et ses technologies

1. Introduction .....	5
2. Définition .....	5
2.1. La couche URI et Unicode .....	6
2.2. La couche XML, XML Schéma .....	6
2.3. La couche RDF & RDFS .....	7
2.4. La couche ontologie.....	7
2.5. La couche de règles .....	7
2.6. La couche Logique .....	8
2.7. La couche confiance et preuve .....	8
3. Les langages du Web sémantique .....	8
4. Les principales composantes du web sémantique .....	10
4.1. Les ontologies.....	10
4.1.1. Les ontologies historiquement .....	10
4.1.2. Les composants d'une ontologie.....	11
4.1.3. Utilisation des ontologies.....	11
4.1.4. Classification des ontologies.....	12
4.2. Annotation sémantique .....	13
5. Gestion de l'incertitude .....	13
5.1. Définition de l'incertitude .....	14
5.2. Types d'incertitude .....	14
5.3. Modèles d'incertitude .....	14
6. Objectives du web sémantique .....	14
7. Conclusion.....	15

### Chapitre2: Réseaux bayésiens et réseaux bayésiens orientés objets

1. Introduction .....	17
2. Réseaux bayésiens Standard.....	17
2.1. Définition.....	17

2.2. L'apprentissage.....	18
2.2.1 Apprentissage de la structure .....	19
2.2.2. Apprentissage des paramètres .....	19
2.3. L'inférence dans les réseaux bayésiens .....	19
2.3.1. Inférence exacte .....	20
2.3.2. Inférence approchée .....	20
3. Réseaux bayésien orientée objet .....	20
3.1. Définition.....	21
3.2. Le cadre orienté objet .....	23
3.2.1. Classe dans RBOO.....	23
3.2.2. Hiérarchie de classes.....	23
3.2.3. Héritage.....	23
3.2.4. La liaison.....	24
3.3. Inférence en RBOO .....	24
3.3.1. Généré Réseau Bayésien Standard.....	24
3.3.2. Généré Réseau Bayésien à Sections Multiples .....	25
4. Les ontologies probabilistes .....	27
4.1. Les travaux précédents .....	27
5. Conclusion.....	28

### Chapitre3: Conception du système

1. Introduction .....	30
2. Architecture globale de notre système.....	30
3. La création de l'ontologie probabiliste.....	31
3.1. La création du réseau bayésien oriente objet à partir d'ontologie déterministe .....	31
3.1.1. L'extraction des classes et des propriétés de l'ontologie.....	31
3.1.2. La construction de réseaux bayésien oriente objet.....	33
3.2. L'intégration de RBOO dans l'ontologie pour qu'il soit formaliser .....	39
3.2.1. Méta-ontologie de RBOO .....	39
3.3. Inférence .....	41
4. Conclusion.....	43

### Chapitre4: Implémentation du système

1. Introduction .....	45
2. Outils et langages utilisés.....	45

2.1. OWL API.....	45
2.2. Unbbayes .....	45
2.3. Éclipse .....	46
2.4. JAVA.....	46
3. Les Interfaces .....	46
3.3. L'interface d'entrée.....	47
3.2. Gestion de l'ontologie probabiliste.....	49
3.2.1. Créer RBOO.....	49
3.2.2. Création Classes.....	49
3.2.3. Apprentissage des paramètres.....	51
3.2.4. Instanciation .....	51
3.2.5. Ontologie probabiliste.....	52
3.2.6. Inférence.....	53
4. Conclusion.....	53

#### Chapitre5 : Test du système

1. Introduction .....	55
2. Motivation étude de cas.....	55
3. Cancer du foie .....	55
3.1 Définition .....	55
3.2. La propagation du cancer du foie .....	56
3.3. Facteurs de risque .....	56
3.4. Symptômes possibles d'un cancer du foie.....	56
3.5. Le diagnostic.....	57
4. Ontologie du cancer du foie .....	57
5. La création de l'ontologie probabiliste.....	59
5.1. Chargement de l'ontologie Liver Cancer .....	59
5.2. Extraction des TBox et ABox.....	60
5.3. Créer le réseau bayésien orienté objet .....	61
5.4. Création des sous structures (les classes) .....	61
5.5. Apprentissage des paramètres .....	62
5.6. Création de la structure globale (instanciation).....	63
5.7. L'intégration .....	63
5.8. L'inférence.....	63



6. Résultat de test .....	64
6.1. La hiérarchie de l'ontologie probabiliste obtenue .....	65
6.2. Les syntaxes de l'ontologie probabiliste .....	66
7. Conclusion.....	69
Conclusion générale : .....	70
Bibliographies: .....	76

### **Liste des figures :**

Figure 1 : pyramide du web sémantique selon Sir Tim Berners Lee [2]. .....	6
Figure 2 : Exemple du réseau bayésien [18]. .....	18
Figure 3 : (a) La classe Genericcow. (b) La classe expert Milk cow. (c) La classe Meatcow [25]. .....	22
Figure 4 : la classe situation Stock avec deux instanciations de la classe Milk cow et deux instanciations de la classe Meatcow [25]. .....	22
Figure 5 : Le RB du RBOO de stock [25]. .....	25
Figure 6 : Le RBSM du RBOO de stock, (a) section de cow3.MilkCow, (b) section de cow4.MilkCow, (c) section de cow1.MeatCow, (d) section de cow2.MeatCow.....	26
Figure 7 : Système global.....	31
Figure 8 : Exemple d'une hiérarchie des classes d'une ontologie .....	32
Figure 9 : Exemple des propriétés d'une ontologie (datatypeproperty et objectproperty).....	32
Figure 10 : Exemple des individus d'une ontologie.....	33
Figure 11 : étapes de la construction d'un RBOO. ....	33
Figure 12 : étapes de la création d'un RBOO et ces classes. ....	34
Figure 13 : étapes de la création des nœuds.....	34
Figure 14 : étapes de la création de la liaison entre les nœuds.....	35
Figure 15 : étapes de l'apprentissage des paramètres .....	36
Figure 16: étapes de Construction la structure globale .....	37
Figure 17: étapes d'Ajouter la liaison entre les instances. ....	38
Figure 18 : vue globale de méta ontologie de RBOO. ....	39
Figure 19 : visualisé en OWL de la méta-ontologie d'un RBOO.....	41
Figure 20 : les classes de RBOO : (a) classe employé, (b) classe recrutement .....	41
Figure 21: la classe situation (décision) de RBOO. ....	42
Figure 22: les sections de RBSM : (a) section employé, (b) section recrutement .....	42
Figure 23 : interface d'entrée.....	47
Figure 24: interface choisir le fichier OWL de l'ontologie.....	48
Figure 25 : interface de la création de RBOO. ....	49
Figure 26 : interface de création et gérer des classes. ....	50
Figure 27 : Interface de l'Apprentissages des paramètres. ....	51
Figure 28 : Interface pour Définir Les Instanciation.....	52
Figure 29 : Interface pour gérer l'inférence. ....	53
Figure 30 : la taxonomie de l'ontologie Liver Cancer [33]. .....	58

Figure 31 : vision générale sur l'ontologie Liver Cancer. ....	59
Figure 32 : chargement de l'ontologie liver cancer .....	60
Figure 33 : Extraction des propriétés des données pour la classe Patient. ....	60
Figure 34: Donner un nom pour le réseau bayésien orienté objet.....	61
Figure 35 : création la classe RiskFactors (1), ces nœuds (2) et la liaison entre les nœuds(3).62	
Figure 36 : Table de probabilité du nœud HepatiteC de la classe RiskFactors.....	62
Figure 37 : création de la structure globale de notre RBOO.....	63
Figure 38 : l'inférence dans notre RBOO. ....	64
Figure 39 : Hiérarchie des classes .....	65
Figure 40 : La hiérarchie des propriétés d'objet.....	65
Figure 41 : La hiérarchie des propriétés des données .....	66
Figure 42 : syntaxe du RBOO. ....	66
Figure 43 : Syntaxe de la classe. ....	67
Figure 44 : syntaxe du Nœud. ....	67
Figure 45 : syntaxe du State. ....	68
Figure 46 : Syntaxe de Situation. ....	68
Figure 47 : syntaxe de l'instance.....	69

***LISTE DES ABREVIATIONS:***

OWL: Ontology Web Language.

W3C: World WideWeb Consortium.

URI: Uniform Ressource Identifier.

XML: eXtensible Markup Language.

RDF: Resource Description Framework.

RDFS: Resource Description Framework Schema.

SBC: Systèmes à Base de Connaissances.

RB: Réseau Bayésien.

RBOO : Réseau Bayésien Oriente Objet.

RBSM : Réseau Bayésien à Sections Multiples.

### 1. Introduction générale

#### 1.1. Contexte du travail

Au cours des dernières années le domaine de la sémantique a été extrêmement actif. Le web sémantique est une vision du web reposant sur le partage structuré et intelligent des données, en généralisant un système de métadonnées, parmi les technologies du Web sémantique : les ontologies.

Les ontologies sont la technologie dorsale pour le web sémantique et plus généralement pour le management des connaissances formalisées décrivant les ressources du Web. Elles fournissent la sémantique exploitable par machine des données et des sources d'informations pouvant être communiquées entre différents agents (logiciel et humaines), tandis que les annotations sémantiques décrivent les ressources en utilisant la "sémantique" définie dans l'ontologie. Mais leur base de la logique classique ne fournit aucun support intégré pour l'incertitude.

L'incertitude est une caractéristique intrinsèque de la plupart des tâches requises, et une réalisation complète du World Wide Web comme une source de données traitables qui exigent des formalismes capables de représenter et de raisonner sous incertitude.

Les formalismes sémantiquement riches et informatiquement efficaces ont émergé pour représenter et raisonner avec la connaissance incertaine. Une ontologie probabiliste est capable de modéliser plusieurs types d'incertitudes, telle que peut être utilisée par la suite dans des applications d'aide à la décision qui doivent être capable de raisonner avec des connaissances incertaines.

#### 1.2. Problématique

Les ontologies simples (classique) sont vraiment puissantes, mais leurs base de la logique classique ne fournit aucun support intégré pour l'incertitude, c'est pour ça qu'il nous a fallu se tourner vers d'autres ontologies qui permettent de supporter l'incertitude que nous utilisons dans le web sémantique, à l'instar des ontologies probabilistes qui sont basés principalement sur les ontologies simples et les modèles graphiques probabilistes.

Les modèles graphiques probabilistes font partie de plusieurs modèles qui représentent les connaissances et le raisonnement. Et parmi ces modèles graphiques y'a les réseaux bayésiens standards. Ces derniers parmi les modèles d'analyse probabiliste. Ils proposent un formalisme mathématique et des bases théoriques solides pour la modélisation des systèmes complexes.

Mais parmi ces inconvénients ne sont pas capable de présentés les relations hiérarchique, et c'est pour cela qu'on a opté pour les réseaux bayésiens orientés objet qui sont non formels mais ils permettent d'exprimer l'incertitude qui manque dans les ontologies.

Pour créer une ontologie probabiliste qui supporte l'incertitude, on doit construire la structure de cette ontologie probabiliste ainsi que ses tables de probabilité.

Donc :

- Comment construire cette ontologie probabiliste à partir d'une ontologie simple en utilisant les réseaux bayésiens orientés objet ?
- Comment représenter le réseau bayésien résultat d'une manière formelle?
- Comment créer une homogénéité entre RBOO et l'ontologie?

### 1.3. Objectif

Les réseaux bayésiens orientés objet (RBOO) sont une représentation pratique des connaissances contenant des structures répétitives, sont un outil souple et performant de gestion de l'incertitude il s'agit d'un outil approprié pour représenter des réseaux bayésiens dynamiques ainsi que certaines relations spéciales qui ne sont pas évidentes pour représenter en utilisant des RB standard. Un des principaux avantages de RBOO est leur capacité de gérer l'incertitude: cela permet de représenter les informations incertaines avec une manière très lisible et clair.

L'objectif de ce projet est de formaliser la connaissance incertaine et de l'intégrer dans l'ontologie, à fin de la rendre compréhensible par les utilisateurs, en utilisant les réseaux bayésiens orientés objet, pour permettre de raisonner sur les connaissances incertaines et exprimer l'incertitude qui manque dans les ontologies , et de créer un outil qui génère une ontologie probabiliste d'un domaine précis .

Afin d'atteindre le but de notre travail, le mémoire sera présenter en deux parties :

**La première partie :** Contient deux chapitres, elle présente le contexte du travail, elle a pour but de présenter le web sémantique et les ontologies.

**Chapitre 1-Le web sémantique et ses technologies:** Ce chapitre propose une présentation du web sémantique et une généralité sur les ontologies. Nous présentons le web sémantique et l'incertain dans le web sémantique, les ontologies avec leurs caractéristiques, leurs constructions, ainsi que les domaines d'application de ces ontologies. Nous allons décrire quelques langages de représentation des ontologies utilisées dans le Web sémantique.

**Chapitre 2- réseaux bayésiens et réseaux bayésiens orientés objets:** Nous éclaircissons la notion des réseaux bayésiens standard et leurs différentes méthodes d'apprentissage et d'inférence. En plus nous donnons une vision sur le réseau bayésien orienté objets qui fait l'objet de notre application, Ainsi on présente leurs différentes méthodes d'inférence.

**La deuxième partie :** Après les généralités présentées dans la première partie. Cette partie met l'accent sur la conception de notre système de la modélisation de l'incertitude dans une ontologie OWL, et aussi on a présenté l'implémentation de notre système.

**Chapitre 3- La conception du système :** Dans ce chapitre, Nous avons défini et expliqué les différentes étapes du processus de la construction d'une ontologie probabiliste d'un domaine donné, on a schématisé les diagrammes nécessaires pour expliquer étape par étape le fonctionnement de notre application. Ce chapitre constitue un point de départ à l'implémentation.

**Chapitre 4- Implémentation et test du système:** Ce chapitre est le résultat de la conception pour réaliser le système de la modélisation de l'ontologie probabiliste. A la fin de ce chapitre on va effectuer un test sur notre application.

**Chapitre 5- Ontologie de test:** Dans ce chapitre, nous avons parlé et défini l'ontologie utilisée pour tester notre application.

La conclusion de ce mémoire synthétise les principales contributions de notre travail.

**Chapitre 1:**  
**Le web sémantique et ses  
technologies**

### 1. Introduction

Le web sémantique est devenu une évolution dans le monde du web, ce dernier a attiré depuis 1999 l'attention de nombreux chercheurs. 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'un certain point de vue, le web sémantique est une évolution pour les systèmes de recherche d'information.

Le Web sémantique 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.

### 2. Définition

La notion de web sémantique fait référence à la vision du web de demain dans lequel les utilisateurs devraient être déchargés d'une bonne partie de leurs tâches de recherche et d'exploitation des résultats, grâce aux capacités accrues des machines à accéder aux contenus des ressources et à effectuer des raisonnements sur ceux-ci.

Concrètement, le web sémantique est une infrastructure qui permet l'utilisation des connaissances formalisées, en plus du contenu informel que l'on peut trouver dans le web actuel. Cette infrastructure s'appuie sur un certain niveau de consensus portant, par exemple, sur les langages de représentation ou sur les ontologies utilisées. Ainsi, elle permet, le plus automatiquement possible, l'interopérabilité et les transformations entre les différents formalismes et les différentes ontologies, qui correspondent à la vision du Tim Berners-Lee qui a défini le Web sémantique en [1] comme suit : « *le web sémantique est une extension du web actuel, dans laquelle on donne à une information un sens bien défini pour permettre aux ordinateurs et aux personnes de travailler en coopération* ».

L'architecture du web sémantique s'appuie sur une pyramide de langages, spécifications et normes pour représenter des connaissances sur le web en satisfaisant les critères de standardisation, d'interopérabilité et de flexibilité.

- Il s'agit de la vision schématique du web sémantique proposée par son inventeur Tim Berners-Lee en 2001.

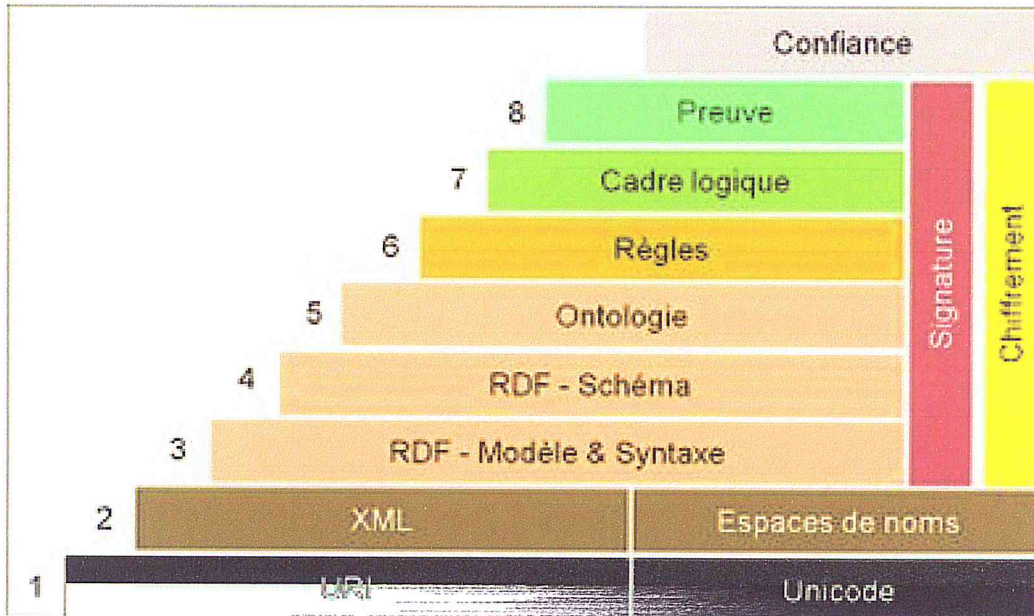


Figure 1 : pyramide du web sémantique selon Sir Tim Berners Lee [2].

Chaque boîte correspond à une technologie spécifique (Unicode, XML, espaces de noms, etc.) ou à une méthode générale (règles, logique, preuve).

Dans ce qui suit, nous présenterons plus en détail les différents niveaux de cette architecture.

### 2.1. La couche URI et Unicode

- **L'URI (Uniform Resource Identifier)** : 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 proposées.

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 URL (Uniform Resource Locator), au nommage URN (Uniform Resource Name) ou au deux.

- **Unicode** : se donne comme but de coder le texte, dans toutes les écritures et avec tous les symboles nécessaires à tout type de texte. Elle constitue une réponse unique aux problèmes résultant de la multiplicité des codages.

### 2.2. La couche XML, XML Schéma

- **XML (eXtensible Markup Language)** : est un langage qui 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.

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].

### **2.3. La couche RDF & RDFS**

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 (Resource Description Framework)** : fournit un moyen d'insérer de la sémantique dans un document, où l'information est conservée principalement sous forme de déclarations RDF.
- **Le schéma RDF** : décrit les hiérarchies des concepts et des relations entre les concepts, les propriétés et les restrictions domaine pour les propriétés.

### **2.4. La couche ontologie**

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 priorisant en utilisant l'échange de sémantique plutôt que de syntaxe seulement.

### **2.5. 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 de déduire toutes les inférences possibles.

### 2.6. La couche Logique

Se trouve au-dessus de la couche ontologie. Certains considèrent ces deux couches comme étant au même niveau, c-à-d 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 représentée.

### 2.7. La couche confiance et preuve

Les premières couches présentées précédemment ont déjà été standardisées et recommandées par le W3C. Dans le cadre de l'architecture du web sémantique, il est tout différent pour les couches de haut niveau.

- **La couche preuve** : a pour but de prouver la pertinence de l'information retournée par les couches de plus bas niveau afin de pouvoir lui accorder un niveau de confiance.
- **La couche confiance** : dans l'architecture proposée par Tim Bernes-Lee, a pour objectif d'évaluer la fiabilité de l'information et des raisonnements.

## 3. Les langages du Web sémantique

Dans le contexte du Web Sémantique, plusieurs langages ont été développés. La plupart de ces langages reposent sur XML ou utilisent XML comme syntaxe. Nous allons présenter brièvement certains langages principaux XML, XML Schéma, RDF(S) et OWL.

- **Le langage XML** : est un langage de description et d'échange de documents structurés permettant de représenter un document texte de manière arborescente en utilisant un système de balisage qui marque les éléments qui composent la structure et les relations entre ces éléments.

Ce langage a été élaboré pour faciliter l'échange, le partage et la publication des données à travers le web. Ainsi, la majorité des langages/modèles proposés pour le web sémantique sont exprimés en XML.

La connaissance de la structure d'un document XML permet notamment de vérifier la validité de ce document. Un fichier de description de structure (XML Schéma Description, ou fichier XSD) est donc lui-même un document XML.

**Les espaces de nommage**: XML offrent une méthode simple pour qualifier les noms des éléments et les attributs utilisés dans des documents XML, en associant ceux-ci avec des espaces de nommage désignés par des références d'URI.

- **RDF (Resource Description Framework)** [4]: est un modèle de données pour les objets (ressources) et les relations entre eux fournissant des sémantiques simples pour ce modèle de données qui peuvent être représentés en XML. RDF permet de représenter des métadonnées à propos des ressources (identifiées par des URI) du web. La construction de base en RDF consiste en un triplet d'éléments (Ressource, Propriété, Valeur), qu'on appelle déclaration RDF. Par analogie, un triplet RDF est similaire à la déclaration :
  - Ressource (Sujet) : 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.).
  - Propriété (prédicat) : Critère, caractéristique, attribut ou relation qui peut décrire la ressource (titre, couleur, taille, auteur, etc.).
  - Valeur (objet) : C'est la valeur qui sera affectée à la propriété de la ressource. Cette affectation peut être soumise à certaines.
- **RDF-S (RDF Schéma)** : est un vocabulaire de base pour décrire les déclarations RDF, au même titre que le XML-S pour le langage XML. Il ajoute à RDF la possibilité de définir des hiérarchies de classes et de définir les genres et les propriétés des ressources, d'assigner des contraintes spécifiques sur la nature des documents et de fournir des informations sur l'interprétation des déclarations RDF. Les schémas RDF permettent donc de garantir qu'un document RDF est sémantiquement consistant. Dans un schéma, de nouvelles ressources peuvent être définies comme des spécialisations d'autres ressources. Les schémas contraignent aussi le contexte d'utilisation des ressources.
- **Le langage OWL** : est un composant de l'activité Web Sémantique qui vise à rendre les ressources Web plus accessibles aux processus automatisés en ajoutant des informations qui décrivent le contenu Web. Le langage d'ontologie Web OWL définit des ontologies Web. Une ontologie OWL peut contenir des descriptions de classes, de propriétés et de leurs instances. Pour une telle ontologie donnée, la sémantique formelle OWL indique comment déduire ses conséquences logiques, c'est-à-dire les faits qui ne sont pas littéralement présents dans l'ontologie mais déduits par la sémantique.

L'OWL peut être défini en trois sous langages complémentaires proposant une

expressivité croissante [5], chacun conçu pour des communautés de développeurs et des utilisateurs spécifiques : *OWL Lite*, *OWL DL*, *OWL Full*.

### 4. Les principales composantes du web sémantique

Nous examinons dans cette section les deux composants essentiels et importants du Web sémantique: l'ontologie et l'annotation sémantique.

#### 4.1. Les ontologies

Sont la technologie dorsale pour le Web sémantique et plus généralement pour le management des connaissances formalisées décrivant les ressources du Web.

Elles fournissent la "sémantique" exploitable par machine des données et des sources d'informations pouvant être communiquées entre différents agents (logiciel et humaines), tandis que les annotations sémantiques décrivent les ressources en utilisant la "sémantique" définie dans l'ontologie.

Les ressources annotées par les méta-données faciliteront la recherche, l'extraction, l'interprétation et le traitement de l'information d'une manière plus efficace.

##### 4.1.1. Les ontologies historiquement

Le terme Ontologie a été défini en philosophie comme une branche de la métaphysique qui s'applique à l'être en tant qu'être, indépendamment de ses déterminations particulières. En effet, ce terme grec est composé des mots « ontos » et « logos » qui veulent dire respectivement l'essence de l'être.

Au début des années 90, des chercheurs en intelligence artificielle se sont intéressés à cette notion pour la formalisation des connaissances. 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 celle formulée par GRUBER « Une ontologie est une spécification formelle et explicite d'une conceptualisation partagée » [6] et [7].

Dans cette définition :

- la conceptualisation : signifie un modèle abstrait d'un certain aspect du monde, prenant la forme d'un algorithme qui détermine des relations.
- Le terme explicite : signifie que les concepts utilisés ainsi que les contraintes sur leur utilisation sont définis d'une manière claire et précise.
- L'adjectif formelle : se réfère au fait que l'ontologie doit être lisible et compréhensible par les machines.

- **Le terme partagée** : renvoie à l'idée qu'une ontologie n'est pas l'objet d'un individu mais elle est reconnue par un groupe.

### 4.1.2. Les composants d'une ontologie

Une ontologie peut être vue comme un ensemble structuré de concepts et de relations entre ces concepts destinés à représenter les objets du monde sous une forme compréhensible aussi bien par les hommes que par les machines. Les composants d'une ontologie sont :

- **Les Concepts:** Ou classes, définissant un ensemble d'objet, abstrait ou concret, 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 [8]. Les concepts dans l'ontologie sont habituellement organisés dans des taxonomies.
- **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.
- **Les axiomes** : Une ontologie est en outre composée d'axiomes qui forment des contraintes sémantiques pour le raisonnement et donnent un acompte d'une conceptualisation, ils prennent la forme d'une théorie logique.

### 4.1.3. Utilisation des ontologies

Même si le besoin de développer une ontologie est très varié et dépend du domaine d'application, nous pouvons facilement énumérer un certain nombre d'utilités, notamment:

- **La connaissance du domaine** : Les ontologies permettent la modélisation des connaissances dans un domaine particulier, dans lequel opère le système à développer.
- **La communication:** les ontologies assurent une communication fiable et hétérogène entre personnes et machines (agents logiciels ou organisations) du fait qu'elle permet de mettre en place un langage ou un vocabulaire conceptuel commun.
- **L'interopérabilité:** La représentation explicite des connaissances dans un domaine donné sous forme d'une ontologie, permet à son tour une plus grande réutilisation, un partage plus large et une interopérabilité plus étendue.

- **L'aide à la spécification des systèmes:** La représentation conceptuelle des éléments du domaine, permet aux systèmes de réaliser des raisonnements logiques qu'on appelle inférences, et de sortir avec des conclusions capables d'aider l'utilisateur ou le gestionnaire dans ses décisions.
- **L'indexation et la recherche d'information:** Dans le web sémantique, 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.

#### 4.1.4. Classification des ontologies

La classification de van Heijst [9] repose sur deux critères :

- la structure de la conceptualisation.
- le sujet de la conceptualisation.

Pour le premier critère, van Heijst et ses collègues distinguent trois catégories, à savoir :

- Les ontologies terminologiques telles que les lexiques, précisent les termes qui sont utilisés pour représenter la connaissance dans le domaine du discours. Un exemple d'une telle ontologie dans le domaine médical est le réseau sémantique dans UMLS.
- Les ontologies d'information qui spécifient la structure d'enregistrement des bases de données.
- Les ontologies de modélisation de la connaissance précisent les conceptualisations de la connaissance. Par rapport aux ontologies d'information, les ontologies de modélisation de la connaissance ont habituellement une structure interne plus riche. De plus, ces ontologies sont souvent accordées à une utilisation particulière des connaissances qu'ils décrivent.

En ce qui concerne le sujet de la conceptualisation, ils distinguent quatre catégories :

- **Les ontologies de représentation** [9] N'appartiennent à aucun domaine, mais définissent et organisent les primitives de la théorie logique pour permettre la représentation des ontologies. Elles sont généralement utilisées pour écrire les ontologies de domaine et les ontologies de haut niveau.
- **Les ontologies génériques** [10] [11] [12] Sont aussi appelée ontologie de haut niveau ou ontologie top, elles décrivent des concepts généraux, indépendants d'un domaine ou d'un problème particulier. Elles permettent par exemple de formaliser les aspects temporels ou spatiaux des objets du monde réel.

- **Les ontologies de domaine** [9] 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.
- **Ontologies d'application** [9] 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.

### 4.2. Annotation sémantique

Pour le Web Sémantique, l'un des aspects les plus importants est de pouvoir manipuler des annotations sémantiques de documents Web, puisque le Web Sémantique permettra aux machines de comprendre la sémantique des documents et des données. Les annotations sémantiques décrivent le contenu des documents, en associant une sémantique à ces descriptions. Donc ils peuvent être considérés comme des méta-données de documents, ressources du Web.

Clairement, la sémantique de l'annotation est fondée sur des vocabulaires dans les ontologies qui sont spécifiées explicitement dans un langage de représentation.

### 5. Gestion de l'incertitude

Avec l'émergence du Web, l'accès à l'information est de moins en moins restreint. Cependant, la fiabilité de ces informations n'est pas toujours assurée. La problématique du traitement automatique de l'information est en pleine évolution. Mais cette automatisation est confrontée à de nombreux problèmes, en particulier, les imperfections liées à l'information véhiculée par les données textuelles. Dans [13] les auteurs distinguent 8 types d'imperfections

- **ambigüe** : si elle se rapporte à deux éléments distincts et pour lesquels une distinction est difficile;
- **bruitée** : si elle contient des informations extérieures non pertinentes et qui faussent sa bonne interprétation ;
- **biaisée ou non objective**: si son interprétation peut être influencée ;
- **incomplète** : si une partie des informations est absente ;
- **imprécise** : si elle contient du flou ;

- **incertaine** : si on ne peut s'assurer de la véracité de l'information, elle est sujette à un doute ;
- **incohérente** : si elle entre en contradiction avec d'autres informations ;
- **redondante** : si elle est répétée sous plusieurs formes, pouvant par la suite entraîner gestion de l'incertitude une ambiguïté ou une incohérence.

### 5.1. Définition de l'incertitude

Le terme "incertitude" désigne une information qui n'est pas établie avec certitude, qui peut ou non se produire et qui peut être de nature vague. Le but sera alors d'attribuer une valeur de probabilité ou une possibilité à une proposition pour que cette dernière soit vraie ou fausse.

Plusieurs travaux ont été consacrés à l'incertitude, [14], [15] considèrent plusieurs représentations formelles se basant sur des logiques probabilistes, possibilistes et flous.

### 5.2. Types d'incertitude

Les auteurs de [16] ont considéré l'aspect caractériel de l'information, ils identifient deux types d'incertitude :

- **Incertaine stochastique** : caractérise l'aspect aléatoire de l'information, ainsi que sa variabilité.
- **Incertaine épistémique** : liée à l'incomplétude de l'information et au manque de connaissance de l'auteur.

### 5.3. Modèles d'incertitude

Cette classe contient des informations sur les théories mathématiques pour les types d'incertitude. Les types spécifiques de théories comprennent, mais sans s'y limiter, ce qui suit:

- Probabilité.
- Ensembles flous.
- Fonctions de croyance.
- Combinaison de plusieurs modèles (hybrides) [17].

## 6. Objectifs du web sémantique

Un des principaux objectifs du Web sémantique est de permettre aux utilisateurs d'utiliser la totalité du potentiel du Web, ainsi, ils pourront trouver, partager et combiner des informations plus facilement, tel que utiliser des réseaux sociaux, de chatter, de faire des



recherches ou même d'acheter différents produits. Cependant les machines ne peuvent pas accomplir toutes ces tâches sans direction humaine, parce que les pages web sont conçues pour être lues uniquement par des personnes et non par des machines. Le Web sémantique a donc comme principal objectif que ces mêmes machines puissent réaliser seules toutes les tâches fastidieuses comme la recherche ou l'association d'informations et d'agir sur le Web lui-même.

### **7. Conclusion**

À travers ce premier chapitre, nous avons commencé par introduire les notions de base de notre travail à savoir les technologies du Web sémantique. En plus, des ontologies qui permettent de décrire une terminologie pour la base de connaissances, avec leurs caractéristiques, leurs classification, leurs utilisation, ainsi que les domaines d'application de ces ontologies.

Nous avons parlé aussi de l'incertitude dans le web sémantique, leur nature, les types et leurs modèles, et dans notre projet on s'intéresse à l'incertitude probabiliste. Ainsi, on constate que les ontologies probabilistes jouent un grand rôle pour représenter l'incertitude dans le web sémantique.

Dans notre travail on aura besoin des réseaux bayésiens qu'on va les traiter dans le chapitre suivant.

**Chapitre 2 :**

**Réseaux bayésiens**

**et**

**Réseaux bayésiens orientés**

**objets**

### 1. Introduction

Les modèles graphiques probabilistes (MGPs) sont des outils puissants de représentation et de raisonnement dans l'incertain. Les réseaux bayésiens [18] sont des modèles graphiques probabilistes dirigés acycliques conçus par Judea Pearl dans les années 1980, permettant de coder et de manipuler des distributions de probabilité sur des espaces à grande échelle.

L'intérêt particulier des réseaux bayésiens est de tenir compte simultanément de connaissances a priori d'experts (dans le graphe) et de l'expérience contenue dans les données. Domaines d'utilisation principaux : diagnostic (médical et industriel), analyse de risques, détection de spams, datamining, détection de fraudes, exploitation du retour d'expérience, modélisation et simulation de système des complexes, détection d'intrusions, TextMining, analyse de BioPuces, analyse de trajectoires de santé.

### 2. Réseaux bayésiens Standard

Un réseau bayésien (RB) est un système représentant la connaissance et permettant de calculer des probabilités conditionnelles apportant des solutions à différentes sortes de problématiques.

Les réseaux bayésiens figurent parmi les modèles d'analyse probabiliste. Ils proposent un formalisme mathématique et des bases théoriques solides pour la modélisation des systèmes complexes. Un réseau bayésien est constitué de deux composantes :

- Une composante graphique qui consiste en un graphe orienté sans circuit. Les nœuds représentent les variables pertinentes du domaine et les arcs représentent les relations de dépendance entre les variables.
- Une composante numérique qui consiste en un ensemble de distributions de probabilités conditionnelles de chaque nœud dans le contexte de ses parents.

#### 2.1. Définition

Les RBs sont des modèles qui permettent de représenter des situations de raisonnement probabiliste basé sur le théorème de Bayes exprimé par la formule suivante, et ce à partir de connaissances incertaines :

$$P\left(\frac{a}{b}\right) = p(b) \cdot p(b/a) / p(a) [18].$$

Un réseau bayésien  $B = (G, P)$  est défini par :

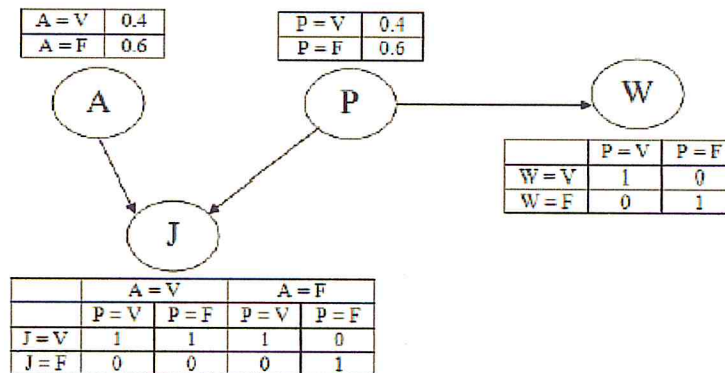
- un graphe dirigé sans circuit  $G = (X, E)$  où  $X$  est l'ensemble des nœuds et  $E$  est l'ensemble des arcs,
- un espace probabilisé  $(\Omega, P)$ ,

- un ensemble de variables aléatoires  $X = \{X_1, \dots, X_n\}$  associées aux nœuds du graphe et définies sur  $(\Omega, P)$ , telles que

$$P(X_1, \dots, X_n) = \prod_{i=1}^n P(X_i / Pa(X_i)) \quad [18].$$

Où  $Pa(X_i)$  est l'ensemble des parents du nœud  $X_i$  dans  $G$ .

**Tables de probabilités :** Les tables de probabilités sont définies par des statistiques relatives au problème à résoudre (peuvent aussi être déterminées par des experts). Chacune des variables dispose d'une table de probabilités conditionnelles relatives aux variables causales dont elle dépend.



**Figure 2 : Exemple du réseau bayésien [18].**

## 2.2. L'apprentissage

La construction d'un réseau bayésien consiste à trouver une structure ou un graphe et estimer les paramètres (probabilités conditionnelles). Cependant devant une très grande base de données, personne ne peut extraire seule la structure adaptée à une telle quantité de données. C'est ici qu'intervient l'apprentissage artificiel.

On distingue deux types d'apprentissage :

- l'un permet d'obtenir la structure du graphe associé au réseau bayésien.
- l'autre permet d'obtenir la distribution de probabilité.

### 2.2.1 Apprentissage de la structure

L'objectif de l'apprentissage de la structure est de trouver une structure du graphe à partir des données disponibles et qui représente le mieux un problème. Une solution naïve pour trouver la meilleure structure d'un réseau bayésien, est de parcourir tous les graphes possibles, de leur associer un score, puis de choisir le graphe qui a le score le plus élevé. Cependant, le nombre de structures différentes pour un réseau bayésien de  $n$  nœuds est super exponentiel. Il est donc impossible d'effectuer un parcours exhaustif en un temps raisonnable. Pour cette raison, la plupart des méthodes d'apprentissage de structure utilisent une heuristique de recherche dans l'espace des graphes acycliques dirigés.

Les méthodes d'apprentissage de structure qui se basent sur un calcul de score maximisent le score de la structure  $G$  qui décrit le mieux les données  $D$ .

### 2.2.2. Apprentissage des paramètres

L'apprentissage des paramètres pour une structure particulière consiste à estimer les distributions de probabilités a priori ou les paramètres des lois de probabilités à partir des données disponibles.

- Dans le cas où l'on dispose de données complètes, on peut utiliser le maximum de vraisemblance qui utilise la fréquence d'apparition d'un événement dans les données [19][20].
- Au contraire, si les données sont incomplètes, l'algorithme Expectation-Maximisation peut être utilisé [21]. Cet algorithme itératif part du modèle, infère le modèle pour calculer la distribution de probabilité. Puis, sur la base de cette distribution, l'algorithme construit un modèle meilleur. Le processus se répète jusqu'à obtenir le modèle le plus vraisemblable.

### 2.3. L'inférence dans les réseaux bayésiens

L'inférence dans un réseau bayésien concerne le calcul de la probabilité de n'importe quelle variable ou sous ensemble de variables à partir des autres variables observées. Il s'agit donc de déterminer les probabilités conditionnelles d'événements reliés par des relations d'influences.

Les algorithmes d'inférence dans les réseaux bayésiens se répartissent en deux groupes : algorithmes *d'inférence exacte* et algorithmes *d'inférence approchée*.

### 2.3.1. Inférence exacte

La tâche de base de tout système d'inférence probabiliste consiste à calculer la distribution de probabilités a posteriori d'un ensemble de variables de requête, étant donné un événement observé \_ autrement dite, une affectation de valeurs à un ensemble de variables d'observation.

$X$  représente la variable de requête,  $E$  représente l'ensemble de variable d'observation  $E_1, \dots, E_m$ , et  $e$  est un événement observé particulier,  $Y$  les variables non observées  $Y_1, \dots, Y_l$  (parfois nommées variables cachées). D'où l'ensemble complet de variables  $X = \{X\} \cup E \cup Y$ .

Une requête de type demande la distribution de probabilité à posteriori :  $P(X|e)$  [22].

### 2.3.2. Inférence approchée

Etant donné que l'inférence exacte est impraticable dans de grands réseaux multiplement connectés, il est essentiel d'envisager des méthodes d'inférence approchée. Il existe un algorithme d'échantillonnage aléatoire, également nommé algorithme de Monte Carlo, qui fournissent des réponses approchées dont l'exactitude dépend du nombre d'échantillons générés. Ces dernières années, les algorithmes de Monte-Carlo ont été largement utilisés en informatique pour estimer des quantités difficiles à calculer exactement.

## 3. Réseaux bayésien orientée objet

Les paradigmes orientés objets ont été longtemps appliqué dans le cadre des langages de programmation pour rendre le code plus logique et pour faciliter sa réutilisation. Les langages de programmation orientés objets [23] fournissent une structure modulaire qui augmente la robustesse et la flexibilité des programmes. Ainsi, ils sont particulièrement bien adaptés à la conception des programmes à grande échelle. Changements faits d'une propagation de classe naturellement à tous les objets de cette dernière. De même, cette approche peut être appliquée aux réseaux bayésiens pour les rendre plus efficaces et flexibles, permettant leur utilisation au-dessus des domaines plus complexes. Les réseaux orientés objets se sont également avérés particulièrement bien adaptés aux domaines fortement dynamiques.

### 3.1. Définition

Les réseaux bayésiens orientés objet (RBOO) sont une représentation pratique des connaissances contenant des structures répétitives. Il s'agit donc d'un outil approprié pour représenter les réseaux bayésiens actifs ainsi que certaines relations spéciales qui ne sont pas évidentes pour représenter en utilisant des RBs standards (par exemple, examiner un caractère héréditaire d'une personne donnée à celles de ses parents).

La représentation orientée objet utilise les notions suivantes :

- **Classe:** est un fragment du RB. utilisée pour modéliser le domaine, sa spécification inclus des “nœuds”.
- **Nœud:** représente la spécification d'une classe.
- **RB sous-jacent:** d'une classe est le RB qui correspond à la classe [24].

Formellement, une classe T se compose de trois ensembles disjoints de nœuds (I, P, O), où :

- **Input (I) :** est l'ensemble des nœuds d'entrée. Tous les nœuds d'entrée sont des références à des nœuds définis dans d'autres classes (appelés nœuds référencés). Chaque nœud d'entrée a au plus un nœud référencé, il n'a pas de parents en sa classe et aucun enfant en dehors de la classe.
- **Private (P) :** est l'ensemble des nœuds internes, Ils sont des nœuds protégés qui ne peuvent pas avoir des parents ou des enfants en dehors T.
- **Output (O):** est l'ensemble des nœuds de sortie. Ce sont les seuls nœuds de la classe visibles de l'extérieur (utilisables en dehors des instanciations de la classe) et ils ne peuvent pas avoir des parents en dehors de T.

#### **Remarque :**

- Tout nœud N inclus dans la spécification d'une classe T sera “instancié” dans tout instance t de classe T. t:N est le nœud de t représentant N du classe T.
- Le nœud référence : est le moyen de rendre visible des nœuds non accessibles autrement (nœud interne).

Les figures suivantes montrent une modélisation de stock de ferme avec 2 vaches de lait et 2 vaches de viande: en utilisant des classes RBOO :

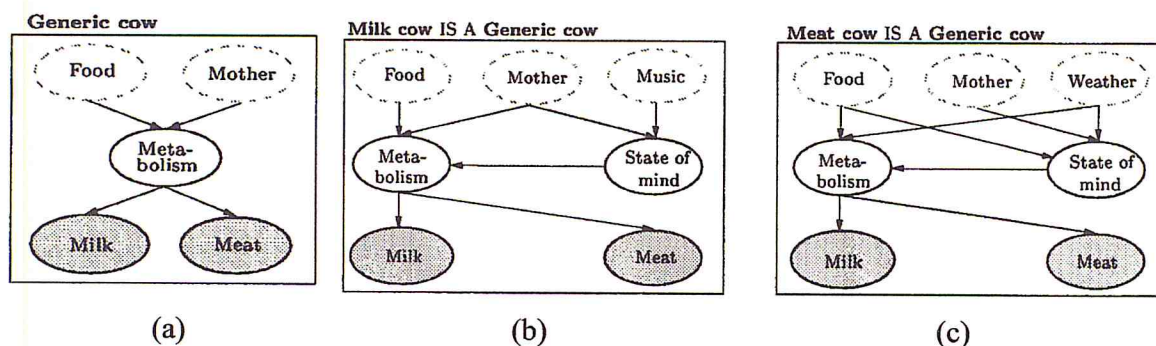


Figure 3 : (a) La classe Genericcow. (b) La classe expert Milk cow. (c) La classe Meatcow [25].

La figure 3 montre une hiérarchie des classes pour notre exemple. (a) classe de vache générique et (b) (c) sont des sous-classes (héritant) de la classe de vache générique. Ceci indique que la classe fille hérite de l'ensemble des nœuds définis par la classe mère. Mother et Food sont des nœuds d'entrée; Un nœud d'entrée est une référence à un nœud en dehors de la classe. Milk et Meat sont des nœuds de sortie utilisables en dehors des instanciations. Les ellipses simples sont des nœuds internes, les ellipses à points représentent les nœuds d'entrée et les ellipses ombragées représentent les nœuds de sortie.

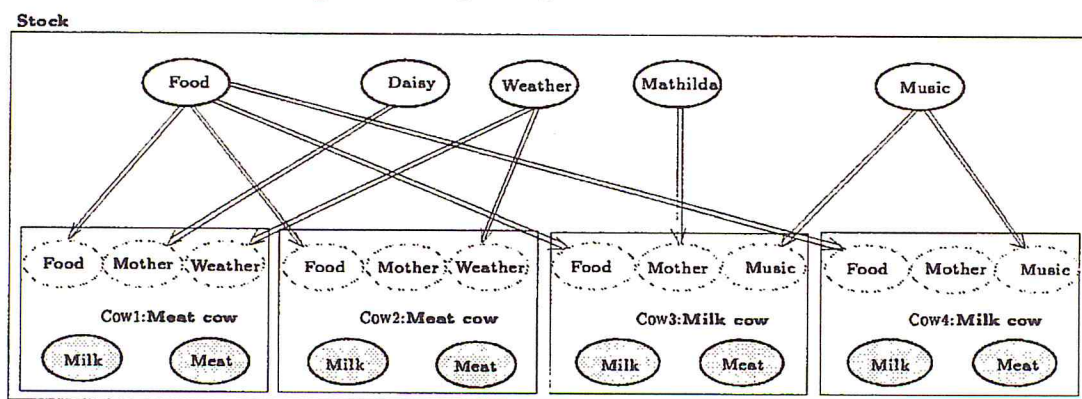


Figure 4 : la classe situation Stock avec deux instanciations de la classe Milk cow et deux instanciations de la classe Meatcow [25].

La figure 4 montre la classe stock qui représente la structure globale du RBOO de l'exemple (figure1). Cow2 et Cow1 sont des instanciations de la classe Meat cow. Cow3 et Cow4 sont des instanciations de la classe Milk cow. Food, Daisy, Weather, Mathilda et Music sont des nœuds de sortie.

Notez que seuls les nœuds d'entrée et les nœuds de sortie sont visibles, car ils sont la seule partie de l'instanciation disponible pour la classe situation (Stock). Les doubles flèches sont des liens de référence, où la feuille d'un lien est une référence à sa racine.



Par exemple, le nœud d'entrée Mother de Cow1 est une référence au nœud Daisy. Cela signifie que chaque fois que le nœud Mother doit être utilisé à l'intérieur de l'instanciation Cow1, le nœud Daisy sera effectivement utilisé. Tous les nœuds sont toujours associés à un CPT, les nœuds qui ne font pas référence à un nœud, mais en raison du potentiel par défaut, un potentiel par défaut est une distribution de probabilité sur les états d'un nœud d'entrée, qui est utilisé lorsque le nœud d'entrée ne fait référence à aucun nœud.

### **3.2. Le cadre orienté objet**

#### **3.2.1. Classe dans RBOO**

Une « classe » est un modèle de données définissant la structure commune à tous les objets qui seront créés à partir de celle-ci (classe). Plus concrètement, nous pouvons percevoir une classe comme un moule grâce auquel nous allons créer autant d'objets de même type et de même structure.

Un réseau bayésien orienté objet se compose d'un ensemble de définitions de classe  $C_1 \dots C_n$  et un RB sous-jacent associé. Où une classe  $C_i$  est un RB sous-jacent sur un ensemble d'entrées et des réelles.

#### **3.2.2. Hiérarchie de classes**

Une hiérarchie de classes répond aux mêmes besoins qu'une hiérarchie de classes en programmation objet : simplifier la spécification des classes similaires, permettre une mise à jour automatique des classes partageant une partie de leurs structures, organiser la connaissance hiérarchiquement.

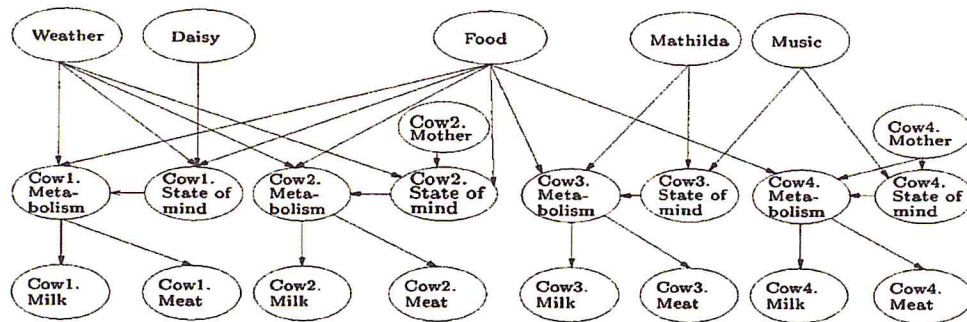
Spécialiser une classe dans le paradigme objet consiste à prendre une classe (attributs et méthodes) et à ajouter ou changer certains de ces attributs ou méthodes. D'une manière analogue, spécialiser une classe dans le réseau bayésien orienté objet consiste à prendre une classe (nœuds et tables de probabilités) et à changer certains d'entre eux ou à ajouter de nouveaux nœuds.

#### **3.2.3. Héritage**

L'une des caractéristiques les plus connues et les plus utiles de la modélisation de l'objet orienté est la capacité de créer des sous-classes qui héritent des propriétés des classes existantes. Les classes sont organisées en une hiérarchie. Une instance d'une sous-classe est une instance de sa classe mère. Cette propriété d'inclusion qui implique une instance d'une sous-classe doit fournir toutes les sorties de la classe parent et ne peut pas exiger toute entrée qui n'a pas été fournie à la classe parente.

Après l'exécution de toutes les tâches nous obtenons un réseau bayésien standard RB(B) bien défini et bien structuré, qui porte la même information que la distribution définie par B. Ainsi, nous pouvons utiliser n'importe quel algorithme d'inférence RB pour répondre à des requêtes : algorithmes d'inférence exacte et algorithmes d'inférence approchée.

La figure 5 décrit le RB du RBOO de stock qui est représenté dans la figure 2 telle que trouvée par l'algorithme ci-dessus.



**Figure 5 : Le RB du RBOO de stock [25].**

### 3.3.2. Généré Réseau Bayésien à Sections Multiples

En supposant RBSM (B) réseau bayésien à sections multiples de B: Leur construction suit les mêmes lignes générales que la simple décrite ci-dessus. En supposant RBSM (B) qui contient un sous-réseau pour chaque instance x de la classe X dans B :

1. Soit RBSM(B) le graphe vide.
2. Ajouter une section (Fragment) dans RBSM(B) pour chaque instantiation de classe dans B, et fixer le nom de section par le nom instantiation.
3. Ajoutez un nœud normal à la section du RBSM(B) pour chaque nœud de sortie et nœud interne dans des instantiations des classes contenues de B correspondant à cette section, et préfixe le nom de l'instanciation au nom du nœud (**Nom d'instanciation. Nom de nœud**).

Notez que chaque nœud d'entrée qui fait référence à un autre nœud deviendra un nœud normal qui représente sa référence, sinon il deviendra un nœud normal doté du potentiel par défaut

4. Ajoutez un lien pour chaque liens dirigés dans chaque instance de B, et répétez ceci pour toutes les instantiations.

Nous obtenons un réseau bayésien à sections multiples  $\_RBSM(B)$  qui porte la même information que la distribution définie par B.

Puisque le RBSM(B) est un réseau bayésien à sections multiples bien défini, nous pouvons utiliser n'importe quel algorithme d'inférence efficace qui [26] s'applique directement aux RBSM pour répondre à des requêtes : un algorithme forêt / arbre de jonction [27].

La figure 6 décrit le RBSM du RBOO de stock qui est représenté dans la figure 2 telle que trouvée par l'algorithme ci-dessus.

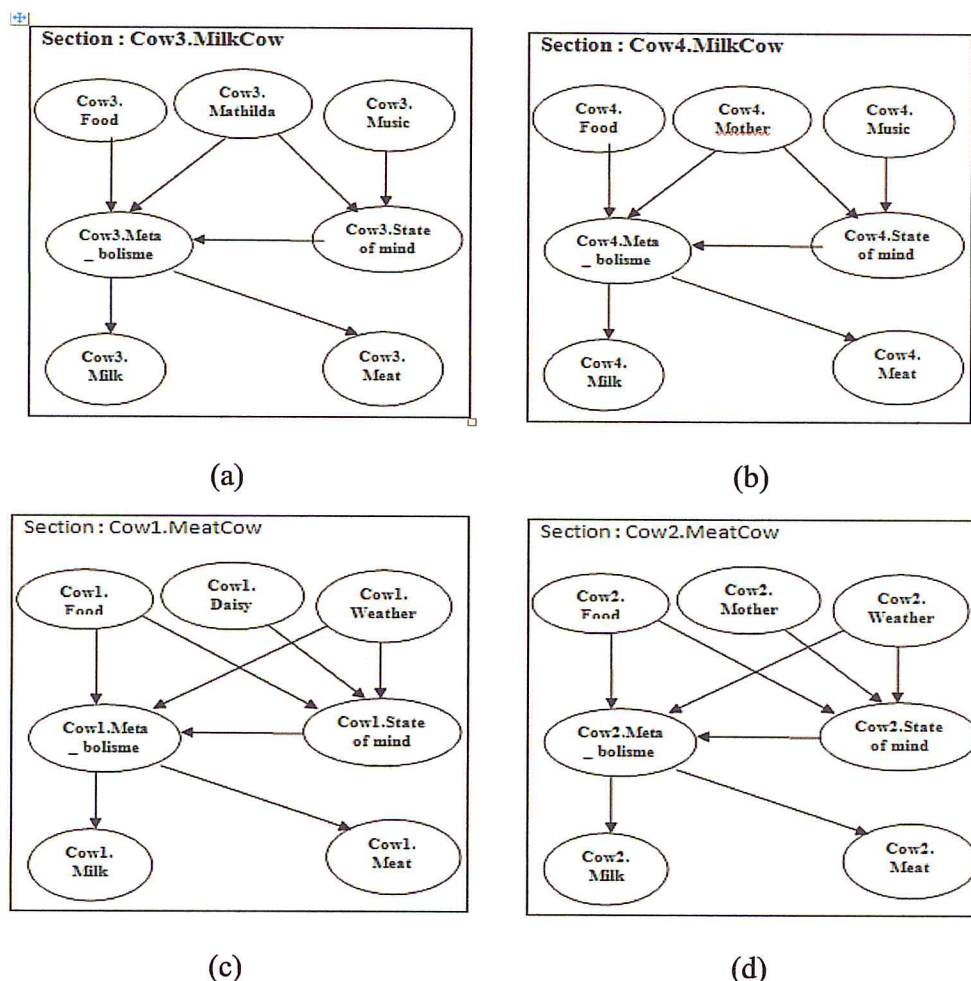


Figure 6 : Le RBSM du RBOO de stock, (a) section de cow3.MilkCow, (b) section de cow4.MilkCow, (c) section de cow1.MeatCow, (d) section de cow2.MeatCow.

La méthode la plus efficace pour effectuer l'inférence dans les RBOO : est la construction du réseau bayésien à sections multiples (RBSM) à partir du RBOO. Ce choix est basé sur plusieurs raisons, y compris:

- permettant de traiter plus rapidement les problèmes de propagations dans des structures réparties.
- il permet une construction simple et facile d'un RBSM à partir du RBOO.

- la décomposition fournie par un RBSM peut conduire à une structure de données qui est plus efficace que celle qui est fournie par un seul arbre de jonction.
- résulte de l'interaction répétée avec un réseau : si un utilisateur demande des requêtes répétées ou ajoute progressivement des éléments de preuve sur un sous-réseau particulier d'un RBSM.

### **4. Les ontologies probabilistes**

Ontologie probabiliste est une extension de l'ontologie classique, elle a la capacité de représenter et raisonner l'incertitude. Divers chercheurs ont proposé des méthodes permettant de représenter formellement un RB sous la forme d'une ontologie probabiliste à l'aide d'un standard de représentation d'ontologie OWL. En outre, les ontologies probabilistes basent généralement sur l'intégration des RBs pour représenter les informations incertaines sous la forme d'un modèle probabiliste.

La représentation formelle obtenue peut être utilisée par la suite comme un support à des opérations de raisonnement dans des contextes différents.

#### **4.1. Les travaux précédents**

Dans les années récentes, plusieurs extensions ont été proposées dans la littérature afin de représenter les ontologies probabilistes, la majorité de ces extensions basée généralement sur la combinaison des RBs et ses extensions avec les ontologies afin de représenter et raisonner avec l'incertitude, parmi ces travaux nous pouvons citer :

(Yi Yang et Jacques Calmet), (Zhongli Ding et Yun Peng) et (Paulo Cesar G. da Costa, Kathryn B. Laskey) ont proposé des extensions du formalisme standard OWL qui sont OntoBayes [29], BayesOWL [30] et PR-OWL [31].

D'un point de vue général, les RBs souffrent ici d'une grande faiblesse. La structure d'un RB est un graphe, qui n'est pas "hiérarchisable" dans le sens où il n'y a pas de moyen direct d'enrichir la représentation, ce qui en fait la représentation de structure et la maintenance d'un tel RB est très difficile. Ainsi, la phase de modélisation dans les RB peut être très complexe et doit pouvoir prendre en compte la modularité du système étudié.

Pour éviter les inconvénients des RBs, nous avons choisi de travailler avec les réseaux bayésiens orientés objets (RBOO) pour notre travail (les réseaux bayésiens orientés objets est une extension de réseaux bayésiens).

Permet les avantages de RBOO :

- Partagent plusieurs similitudes avec les ontologies et conviennent pour représenter les systèmes hiérarchiques puisqu'ils introduisent plusieurs aspects de la modélisation orientée objet, comme les héritages.
- Représenter un outil simple de modélisation rapide.
- Permet une représentation plus compacte des connaissances (en cas particulier des structures répétitives).
- Permet aussi à concepteur le choix entre visualiser globalement son système ou entrer dans les détails de la modélisation d'un des sous-systèmes.

Il est un autre système basé sur les réseaux bayésien orienté objet, intéressé par génération automatique des réseaux bayésiens orienté objet à partir des ontologies [32]. Pour constituer une solution potentielle pour aborder le défi de l'apprentissage structurel des RBOO, tout en minimisant l'implication des experts qui n'est pas toujours évidente à obtenir. Mais ce système il n'est pas conçu pour créer une ontologie probabiliste.

### **5. Conclusion**

Dans ce chapitre nous avons présenté les réseaux bayésiens et nous avons passé en revue quelques algorithmes d'apprentissage ainsi que les méthodes d'inférence les plus connues. Par la suite nous avons présenté les réseaux bayésiens orientés objet avec leurs caractéristiques, comme un outil de représentation graphique des connaissances très efficace et facile à manipuler, par rapport aux autres modèles graphiques.

À la fin nous avons introduit la notion des ontologies probabilistes, ainsi que les travaux les plus connues dans ce contexte.

# **Chapitre3 :**

## **Conception du système**

### 1. Introduction

La gestion de la connaissance incertaine dans le web sémantique est devenue primordiale ces dernières années dans différentes recherches, pour trouver une solution à l'incapacité de représenter et raisonner sur l'incertitude dans les ontologies classique. Dans ce chapitre nous vous proposons une nouvelle solution pour la modélisation et le raisonnement avec la connaissance incertaine dans le web sémantique qui consiste à transformer une ontologie classique formalisée en OWL en une ontologie probabiliste en utilisant les réseaux baysiens orientés objet, le tout en gardant la sémantique de la première ontologie.

### 2. Architecture globale de notre système

L'objectif de notre travail est la construction d'une ontologie probabiliste à partir de la transformation d'une ontologie OWL bien formalisée.

Le diagramme de la figure suivante donne une vision globale de notre travail et les différentes étapes du processus de transformation sont montrées dans le schéma suivant:

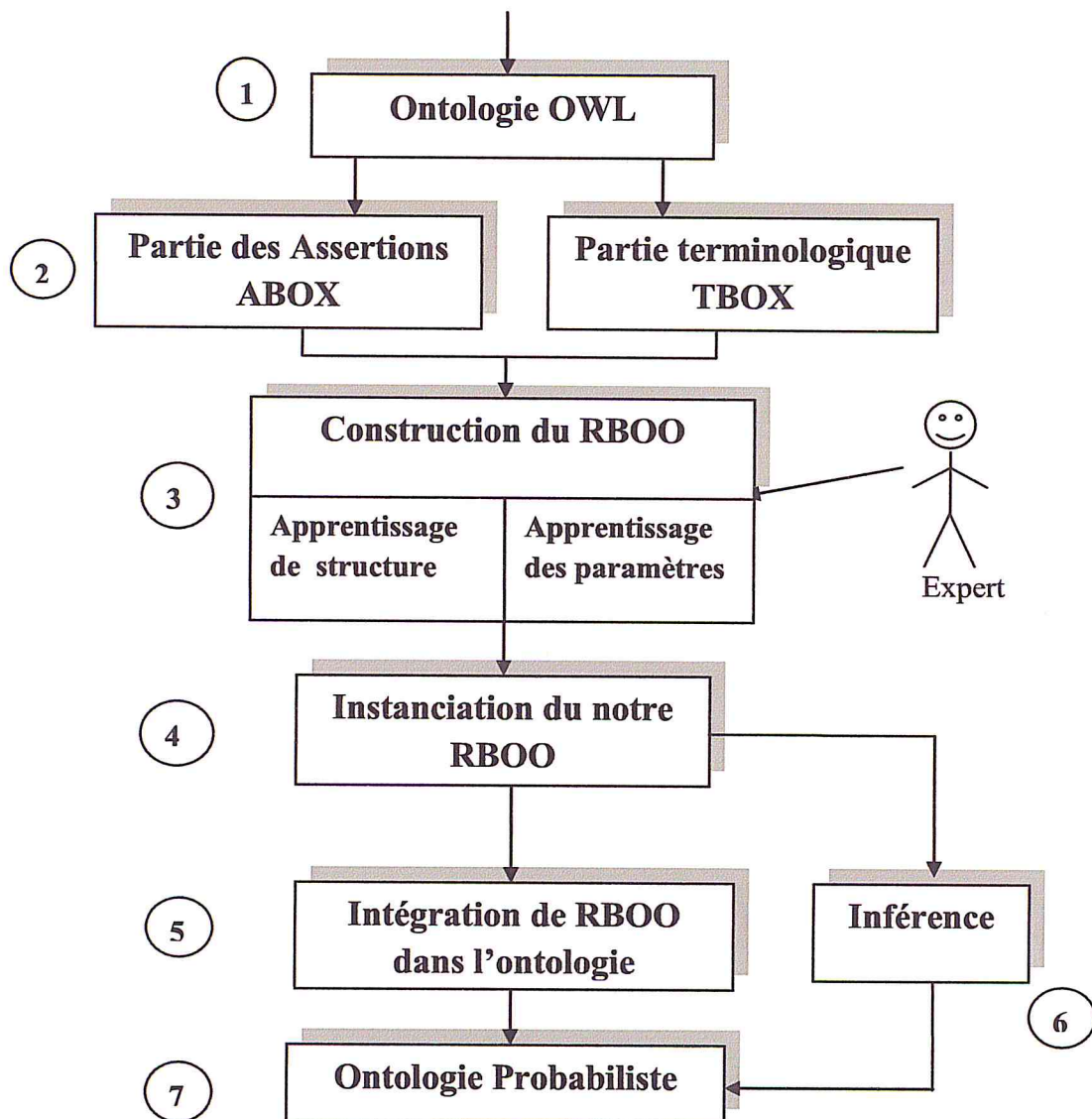


Figure 7 : Système global.

### 3. La création de l'ontologie probabiliste

La création de l'ontologie probabiliste se passe par deux étapes principales :

- La création du réseau bayésien orienté objet à partir de l'ontologie déterministe.
- L'intégration de RBOO dans l'ontologie pour qu'il soit formaliser.

#### 3.1. La création du réseau bayésien orienté objet à partir d'ontologie déterministe

La création du RBOO passe par deux principales étapes, l'extraction des classes et des propriétés de l'ontologie et la création de la structure du réseau bayésien orienté objet :

##### 3.1.1. L'extraction des classes et des propriétés de l'ontologie

Cette partie de l'extraction se fait par l'extraction du schéma de l'ontologie et par l'extraction de la partie assertion de l'ontologie.



➤ L'extraction du schéma de l'ontologie Tbox :

Une ontologie OWL est caractérisée par la partie schéma de l'ontologie qui est plus connue sous le nom de TBOX qui contient :

- Les concepts.
- Les propriétés (ObjectProperty et DataProperty).
- Le domaine et le range de chaque propriété.



Figure 8 : Exemple d'une hiérarchie des classes d'une ontologie

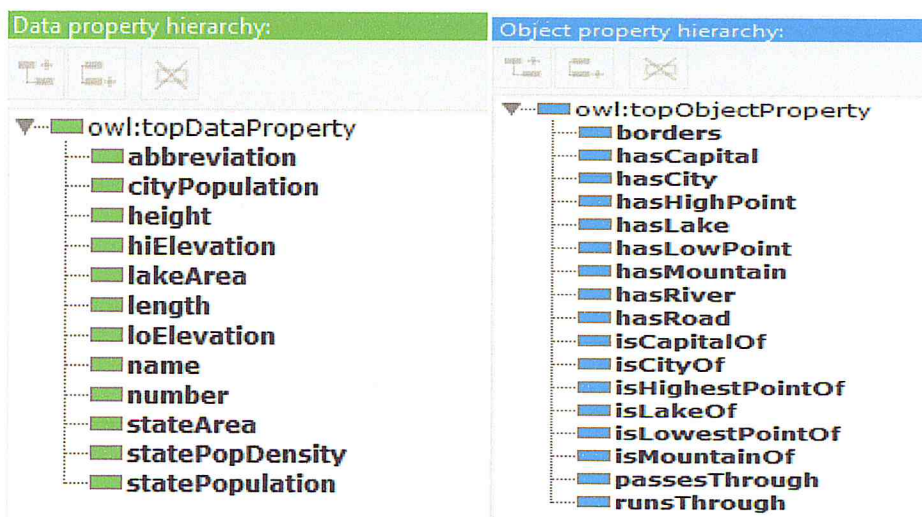


Figure 9 : Exemple des propriétés d'une ontologie (datatypeproperty et objectproperty).

➤ L'extraction de la partie assertion de l'ontologie Abox :

Une ontologie OWL est aussi caractérisée par sa partie ABOX ou ce qu'on appelle la partie des assertions de l'ontologie, elle contient les assertions des attributs.



Figure 10 : Exemple des individus d'une ontologie.

L'extraction de ces deux parties engendra la création du RBOO tout en important les classes, les nœuds les relations entre les nœuds et les relations entre les classes ainsi que les assertions qu'on aura besoin pour la construction de notre ontologie probabiliste.

### 3.1.2. La construction de réseaux bayésien oriente objet

Après l'extraction des TBoxs et ABoxs, La création réseaux bayésien oriente objet (RBOO) se passe par les étapes suivant :

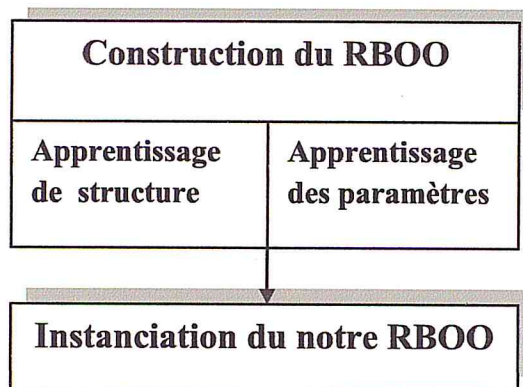


Figure 11 : étapes de la construction d'un RBOO.

#### a. Construction du RBOO

La construction du réseau bayésien orientée objet ce fait par deux étapes :

##### ❖ Apprentissage de structure :

Cette étape passe par le remplissage d'un réseau bayésien par des classes qui contiennent des nœuds précédemment créés.

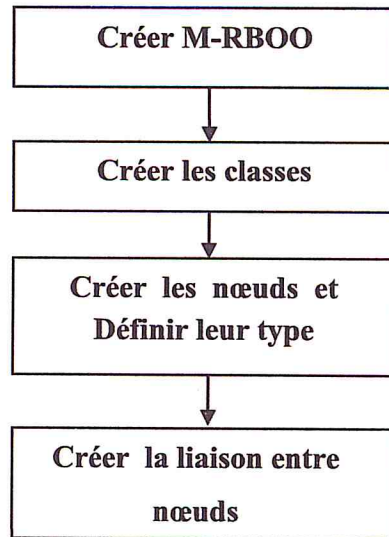


Figure 12 : étapes de la création d'un RBOO et ces classes.

- **Créer M-RBOO :**

La première étape la plus importante est création le monde du réseau bayésien orienté objet, et cette étape nécessite l'importation d'une ontologie qu'on veut modéliser pour notre application.

- **Créer les classes :**

Après la création de RBOO nous commençons par la création des classes de notre réseau qui contient des nœuds (output, private, input), à partir des classes OWL (concepts) que nous avons déjà extraits, et pour cela nous devons la choisir (classe OWL) et l'ajouter dans notre réseau.

- **Créer les nœuds:**

Après la création d'une classe, on doit ajouter les nœuds dans chaque classe, la création d'un nœud passe par les étapes suivantes:

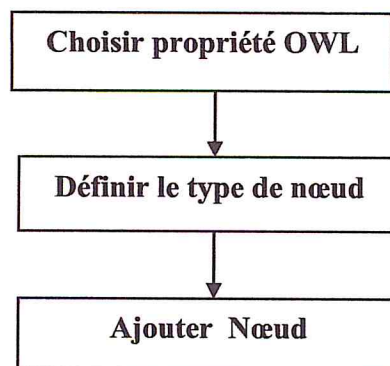


Figure 13 : étapes de la création des nœuds

➤ **Choisir propriété OWL:**

Pour définir un nœud, on doit choisir une propriété OWL que nous devons utiliser parmi les propriétés extraites associées à la classe OWL qui correspond à la classe du nœud.

➤ **Définir nœud et leur type:**

Après le choix, on doit définir le type de nœud, attribuer un seul type pour chaque nœud. Il y a trois types:

- Nœud d'entrée (Input Nœud).
- Nœud de sortie (Output Nœud)
- Nœud interne (Private Nœud).

➤ **Ajout du nœud :**

Après l'exécution de toutes les tâches précédentes on pourra ajouter le nœud à sa classe.

• **Créer la liaison entre nœuds :**

Dans cette étape on va faire la liaison entre les nœuds de la même classe. L'ajout d'un lien se passe par les étapes suivantes :

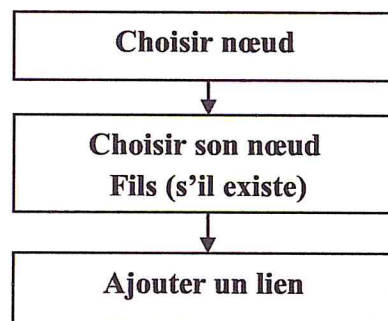


Figure 14 : étapes de la création de la liaison entre les nœuds

➤ **Choisir des nœuds :**

On va choisir un nœud parmi les nœuds déjà définis.

➤ **Choisir son nœud Fils (s'il existe) :**

Après le choix du nœud, On choisit le fils de ce nœud. Où :

- Si le nœud de type input alors son fils est un nœud de type Output ou bien Private.
- Si le nœud de type Private alors son fils est un nœud de type Private ou bien Output.
- Si le nœud Output alors son fils est un nœud de type Output.

➤ **Ajouter un lien :**

Après le choix des nœuds, on va ajouter la liaison entre les nœuds.

❖ **Apprentissage des paramètres :**

Dans notre travail l'apprentissage des paramètres se fait d'une manière semi automatique par l'expert à partir du remplissage de la table probabiliste de chaque nœud déjà défini. Où nous considérons les paramètres sont complètes.

On passe par les étapes suivantes qui montent cette partie par le diagramme suivant:

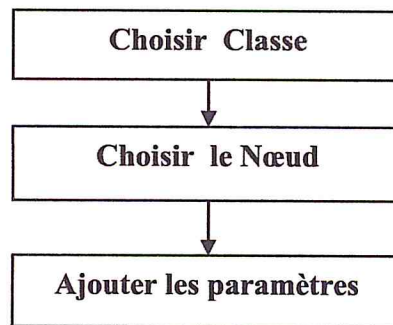


Figure 15 : étapes de l'apprentissage des paramètres

• **Choisir Classe :**

Cette étape consiste à choisir une classe parmi l'ensemble des classes définies déjà.

• **Choisir le Nœud:**

On va choisir les nœuds un par un qu'on veut leur donner la distribution probabiliste de cette classe.

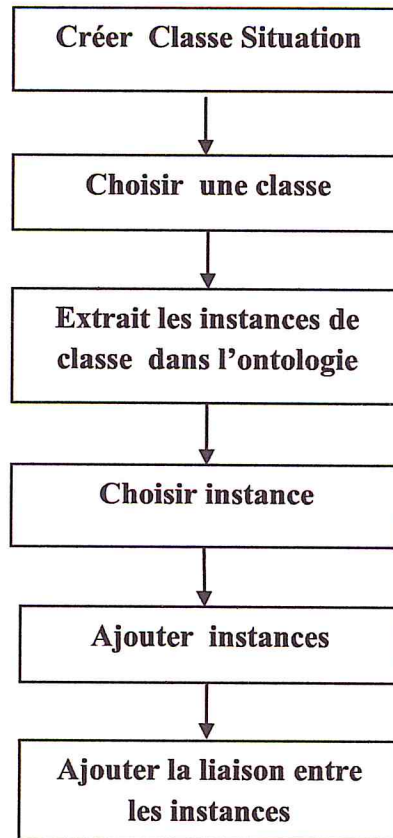
• **Ajouter les paramètres:**

Après l'extraction de la table probabiliste, l'expert va modifier les valeurs initiales de la table selon le besoin.

**b. Instanciation du notre RBOO**

Dés que les étapes précédentes sont achevées, nous pourrons procéder à l'étape finale qui est la construction de la structure globale du réseau bayésien orienté objet. Et pour cela nous devons spécifier et réunir les classes et leur sous structures en utilisant les nœuds d'entrées (Input nœuds) et des nœuds de sortie (Output nœuds) que nous avons précédemment définis pour faire le lien entre les sous- structures.

La construction passe par les étapes suivantes:



**Figure 16: étapes de Construction la structure globale**

- **créer classe situation :**

Class Situation: est une classe qui contient la structure globale de RBOO (les instances des classes et la liaison entre eux), elle représente une situation du monde réel. Pour créer cette classe on va définir le nom de cette classe et l'ajouter dans la liste des classes de réseau RBOO.

- **choisir une classe :**

Nous devons d'abord choisir une classe parmi l'ensemble des classes définies précédemment, et elle doit être différente de la classe situation.

- **Extraire les instances de classe dans l'ontologie :**

Nous extrairions les instances de classe OWL qui correspond à la classe choisir.

- **choisir instance :**

Après l'extraction des instances, on va choisir une instance à partir de l'ontologie.

- **Ajouter instances :**

Ajouter l'instance choisir à la liste des instances dans la classe situation. On peut avoir plusieurs instances pour une classe.

- **Ajouter la liaison entre les instances :**

Cette dernière tâche consiste à faire la liaison entre les instances (*liaison référence*), et cela en reliant deux nœud de deux types différents (Output / Input) de deux instances différentes.

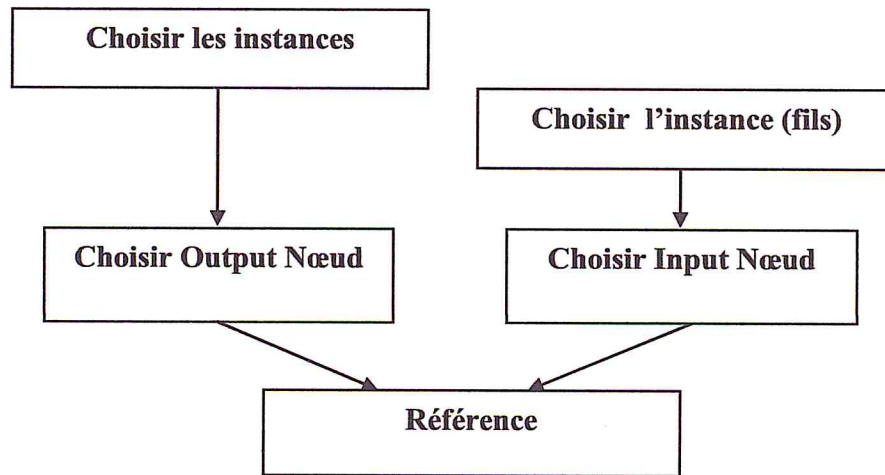


Figure 17: étapes d’Ajouter la liaison entre les instances.

- **choisir les instances :**

Nous choisissons une instance parmi les instances définies dans une classe situation.

- **choisir l’instance (fils) :**

Nous choisissons une autre instance (fils) (s’il existe), selon la structure de notre ontologie. C.à.d. il existe au moins une relation (Object propriétés) qui relie les individus OWL qui correspondent aux deux instances choisies.

- **choisir Output Nœud :**

Après le choix des deux instances, on va choisir l’un des nœuds Output qui sont définis dans cette instance.

- **choisir Input Nœud :**

L’extraction des nœuds input de l’instance fils dépend du nœud output choisi. C.à.d. le nœud input doit avoir les mêmes états que le nœud output. Et à partir de cette règle on va choisir un nœud input parmi les nœuds extraits.

- **Ajouter la liaison :**

Après le choix des deux nœuds (Input Nœud et Output Nœud), on pourra ajouter la liaison entre ces deux nœuds dans la classe situation.

### 3.2. L'intégration de RBOO dans l'ontologie pour qu'il soit formaliser

Notre but est la construction d'une ontologie probabiliste qui donne la main à la distribution de probabilité de chaque concept, et cette dernière (ontologie probabiliste) est la combinaison entre une ontologie simple est un méta model.

#### 3.2.1. Méta-ontologie de RBOO

La méta-ontologie représente la structure globale (les éléments principaux) d'un réseau bayésien orienté objet,

Nous avons présenté notre méta-ontologie dans un diagramme de classe. Comme suite :

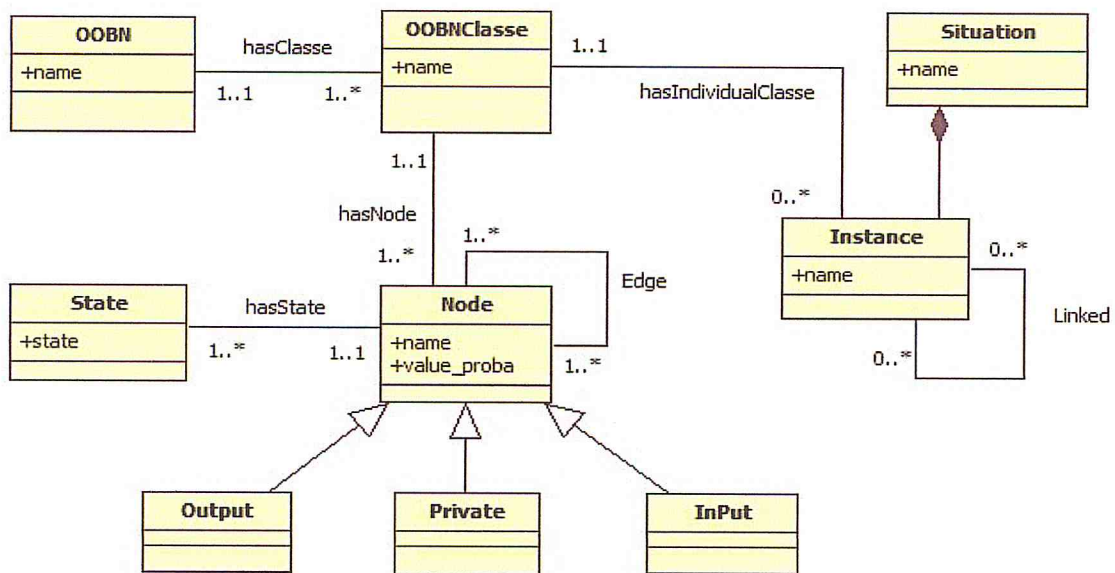


Figure 18 : vue globale de méta ontologie de RBOO.

Description du diagramme de classe :

Classe	Attributs et types	Description des classes
OOBN	NameO: String	Elle représente un concept du monde de réseau bayésien orient objet, et un ensemble des OOBNClasse.
OOBNClasse	NameC : String	Elle représente l'unité de la base de réseau bayésien orient objet, et constitué un ensemble des variables et d'un DAG sur ses variables.



<b>Node</b>	NameN: String Value_Proba : String	Elle représente l'unité élémentaire de réseau bayésien orient objet (les variables). Ainsi que chaque variable contient un type (Input, Output ou Private), des états et des valeurs probabilistes qui correspondent à ces états.
<b>Output</b>		Elle représente les variables de sortie.
<b>Input</b>		Elle représente les variables d'entre.
<b>Private</b>		Elle représente les variables encapsule.
<b>State</b>	State: String	Elle représente les états de chaque variable ( <b>Node</b> ).
<b>Situation</b>	NameS: String	C'est une classe particulière constituée d'un ensemble des <b>instances</b> , et la liaison entre eux.
<b>Instance</b>	NameIC : String	Elle représente les instanciations des classes ( <b>OBNClasse</b> ).

A partir de ce diagramme de classe nous allons créer une ontologie qui représente notre méta-ontologie. Où :

- chaque classe dans notre diagramme de classe devient un concept dans l'ontologie.
- chaque attribut dans une classe devient un Data Properties dans l'ontologie pour le concept correspondant.
- chaque liaison entre les classes devient un Object Properties dans l'ontologie.

Cette ontologie permet à une ontologie OWL de représenter des modèles probabilistes bayésiens complexes d'une manière suffisamment flexible pour être utilisé par divers outils probabilistes bayésiens (exemple : Unbbayes, Netica, Hugin, etc.).

La figure ci-dessous représente les composants de méta-ontologie de RBOO.

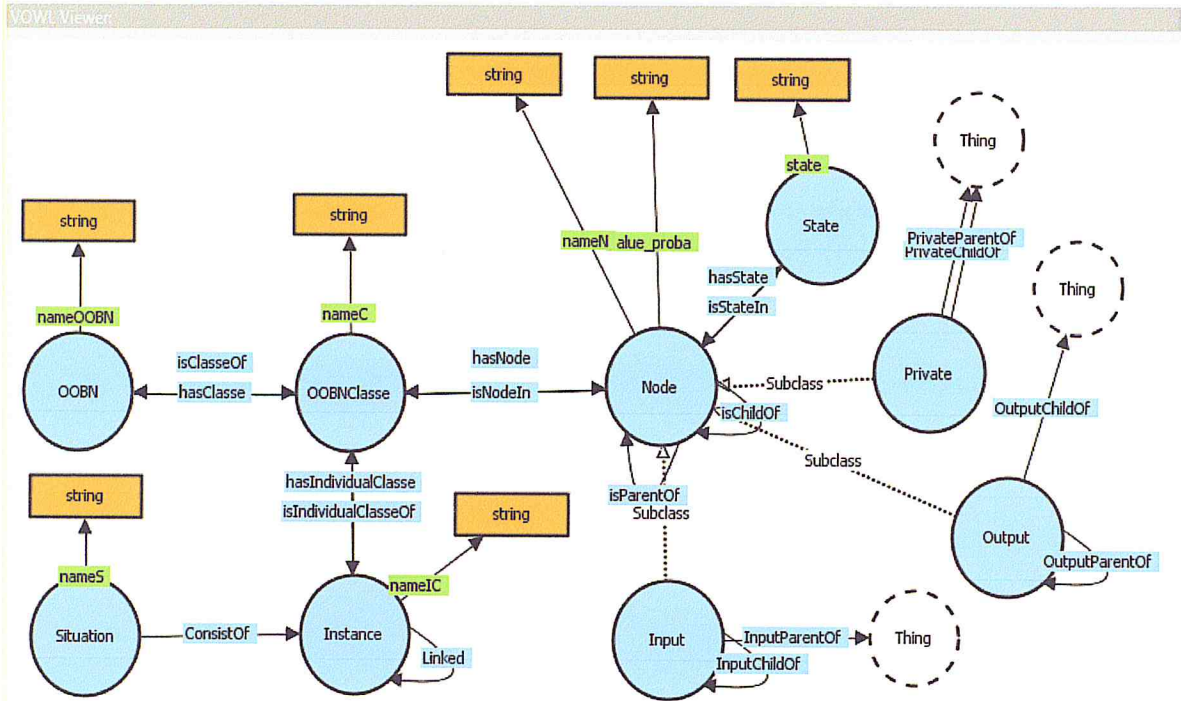


Figure 19 : visualisé en OWL de la méta-ontologie d'un RBOO.

L'objectif de notre méta-ontologie est de nous permettre d'avoir une structure globale de notre domaine étudié et ainsi avoir à notre portée toutes les classes, les nœuds, les relations entre eux pour les utilisés dans l'ontologie probabiliste pour modéliser l'incertain, contrairement aux ontologies traditionnelles qui ne fournissent pas un soutien adéquat à l'incertitude.

### 3.3. Inférence

À la fin, nous pouvons effectuer l'inférence dans notre système et cela en générant un RBSM à partir du RBOO créé, ensuite exécuter l'algorithme : forêt /arbre de jonction [27]. Les figures suivantes montre un exemple d'un RBOO de gestion personnel de recrutement.

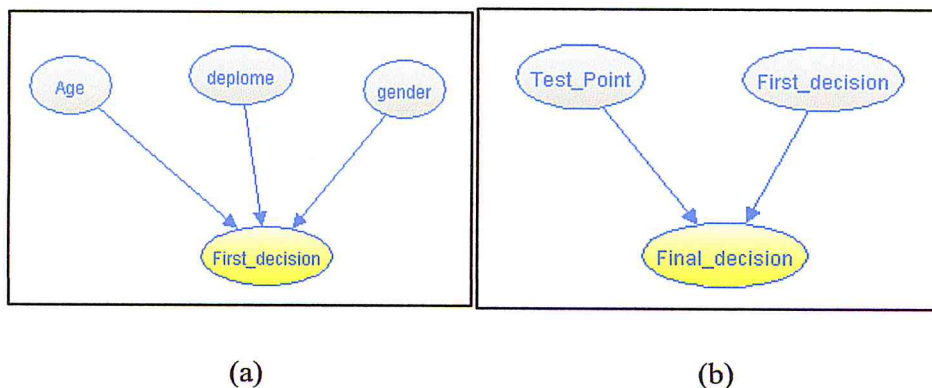


Figure 20 : les classes de RBOO : (a) classe employé, (b) classe recrutement.

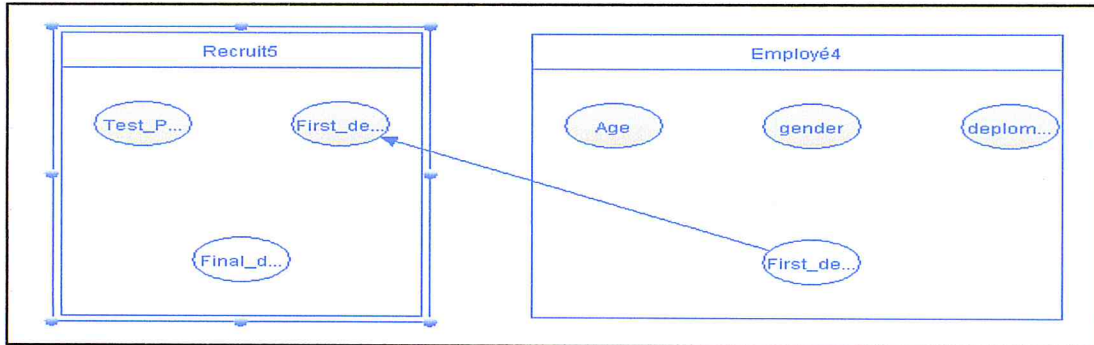
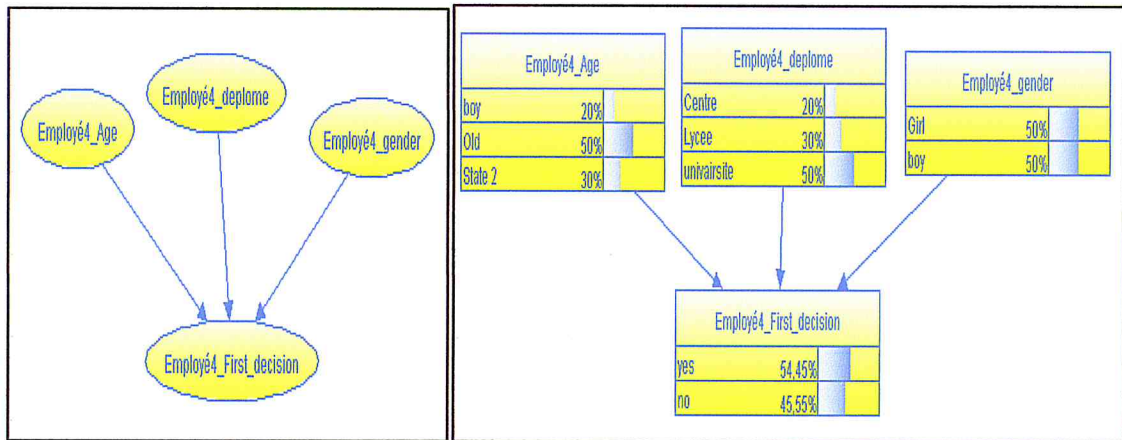


Figure 21: la classe situation (décision) de RBOO.

La figure suivante montrer un RBSM générer à partir le RBOO de gestion personnel de recrutement.



(a)



(b)

Figure 22: les sections de RBSM : (a) section employé, (b) section recrutement.

### **4. Conclusion**

Dans ce chapitre, Nous avons défini et expliqué les différentes étapes du processus de la construction d'une ontologie probabiliste d'un domaine donné, nous avons schématisé les diagrammes nécessaires pour expliquer étape par étape le raisonnement de notre idée et comment elle fonctionne, comme nous avons vu dans ce chapitre les différentes étapes pour l'apprentissage de structure et des paramètres. Après avoir achevé toutes ces étapes on pourra obtenir notre structure globale de notre RBOO et ainsi définir les instanciations des classes et l'ajout des liaisons entre ces dernières. À la fin nous plaçons les valeurs obtenus dans leurs places dans l'ontologie probabiliste pour servir aux inférences ultérieurement, et nous pouvons effectuer l'inférence dans notre RBOO.

**Chapitre4 :**  
**Implémentation**  
**du système**

### 1. Introduction

Après avoir établi une étude complète sur les ontologies et les réseaux bayésien oriente objet, ce chapitre est consacré à l'implémentation de l'application de ce dernier tout en présentant les langages et les outils utilisés ainsi que les copies d'écran de chaque interface.

### 2. Outils et langages utilisés

Il existe plusieurs outils et langages pour la gestion des ontologies, Parmi ces outils nous avons utilisé OWL API, unbbayes, Eclipse, et parmi les langages nous avons utilisé JAVA.

#### 2.1. OWL API

La bibliothèque de code Java open-source pour Ontology Language Web (OWL) est une interface de programmation d'application (API) utilisée pour créer des classes et des méthodes qui chargent spécifiquement et enregistrer des fichiers OWL. L'API peut également être utilisée pour manipuler des données OWL et effectuer une analyse fondée sur la logique. Il est principalement utilisé pour développer des composants et des applications autonomes.

L'API OWL comprend les composants suivants:

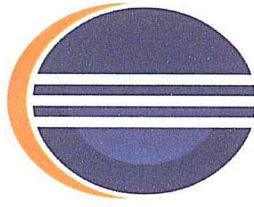
- Une API pour OWL 2 et une implémentation de référence en mémoire efficace
- Analyseur RDF / XML et écrivain
- OWL / XML parser and writer
- OWL Functional Syntax parser and writer
- Analyseur de tortue et écrivain
- Analyseur KRSS
- OBO Flat file format parser

#### 2.2. Unbbayes



UnBBayes est un Framework de réseau probabilisé écrit en Java. Il a une interface graphique et une API avec inférence, échantillonnage, apprentissage et évaluation. Il prend en charge les réseaux bayésiens, les diagrammes d'influence, MSBN, OOBN, HBN, MEBN / PR-OWL, PRM, structure, paramètre et apprentissage progressif.

### 2.3. Eclipse



Eclipse est une plate-forme open source basée sur Java qui permet à un développeur de logiciels de créer un environnement de développement personnalisé (IDE) à partir de composants de plug-in construits par les membres d'Eclipse. Eclipse est géré et dirigé par le Consortium Eclipse.org.

Eclipse a débuté en 2001 lorsque IBM a fait don de trois millions de lignes de code à partir de ses outils Java. L'objectif initial d'Eclipse était de créer et de favoriser une communauté IDE open source qui compléterait la communauté qui entoure Apache.

Bien que la plate-forme Eclipse soit écrite en Java, elle prend en charge les plug-ins qui permettent aux développeurs de développer et de tester un code écrit dans d'autres langues.

### 2.4. JAVA

Java est un langage de programmation informatique orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au SunWorld. Il permet de créer des logiciels compatibles avec de Linux, Macintosh, Solaris). Java donne aussi la possibilité de développer des programmes pour téléphones portables et assistants personnels. Enfin, ce langage peut-être utilisé sur internet pour des petites applications intégrées à la page web (applet) ou encore comme langage serveur (jsp).

**Caractéristiques du Java :** Parmi les caractéristiques les plus importantes de Java  
Orienté Objet : La conception orientée objet est une technique de programmation qui se concentre sur les données (les objets) et sur les interfaces avec ces objets.

### 3. Les Interfaces

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

### 3.3. L'interface d'entrée

La première interface à afficher est la suivante

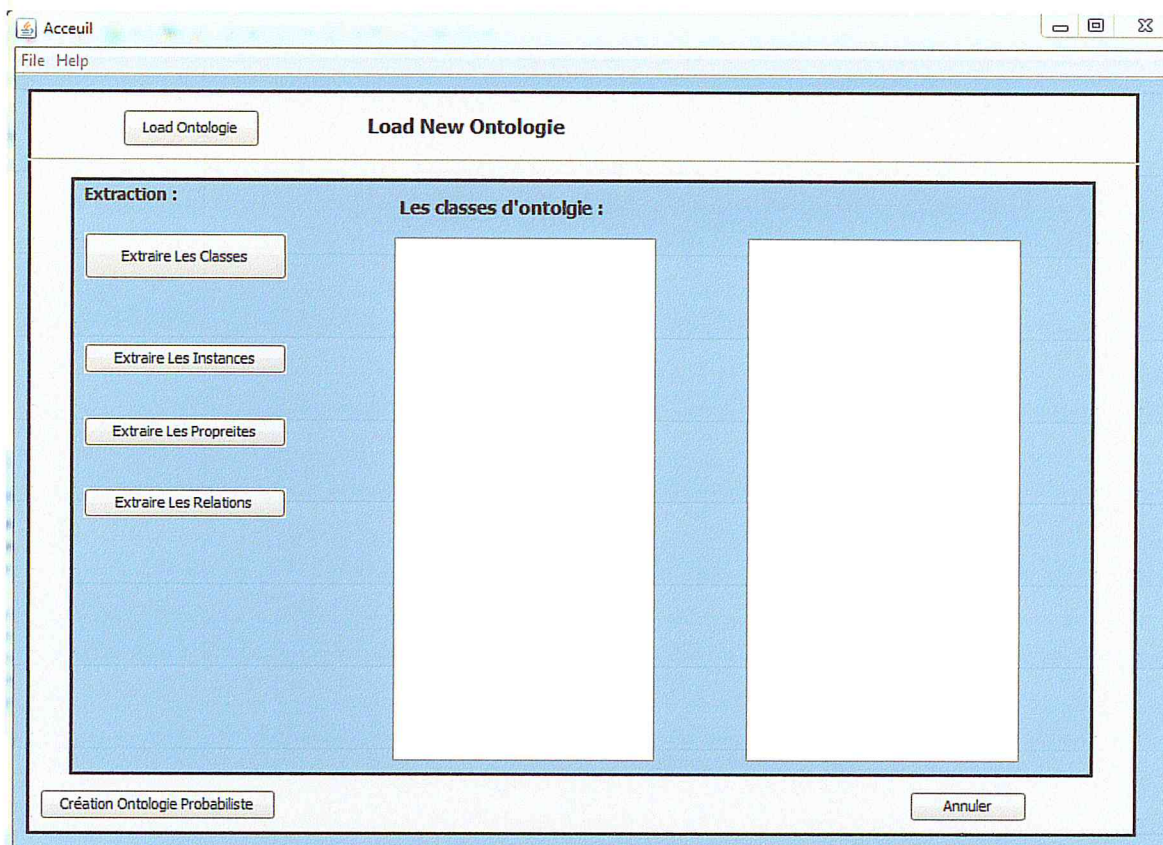


Figure 23 : interface d'entrée

C'est l'interface qui permet l'accès à la modélisation de l'ontologie. On doit d'abord importer l'ontologie OWL depuis un emplacement quelconque, et puis, il faut extraire les parties TBOX et ABOX de l'ontologie pour pouvoir accéder à l'interface de la création de l'ontologie probabiliste.s

- **Load Ontologie** : Elle importe le fichier de l'ontologie OWL en parcourant son chemin :



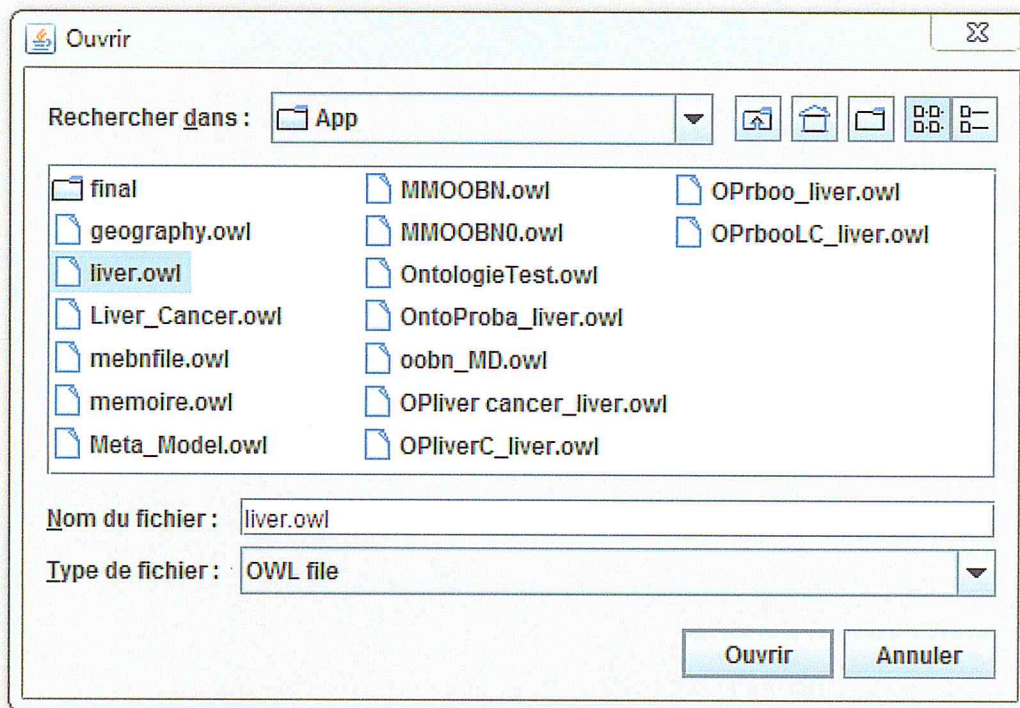


Figure 24: interface choisir le fichier OWL de l'ontologie.

- **Extraire Les Classes** : Sa fonction est l'extraction des classes de l'ontologie OWL, pour les utiliser dans l'ontologie probabiliste.
- **Extraire Les Instances** : Sa fonction est l'extraction des instances de l'ontologie OWL, pour les utiliser comme des instances dans l'ontologie probabiliste.
- **Extraire Les Propriétés** : Sa fonction est l'extraction des propriétés de données de l'ontologie (DataTypeProperty) pour la création des nœuds.
- **Extraire Les Relations** : Sa fonction est l'extraction des objets propriétés de l'ontologie (ObjectProperty) pour ajouter la liaison entre les nœuds et les classes.
- **Création Ontologie Probabiliste** : Après l'extraction des TBOX et ABOX, ça nous donne l'accès à l'interface suivante :

### 3.2. Gestion de l'ontologie probabiliste

Cette interface permet l'effectuation des différentes tâches de notre système :

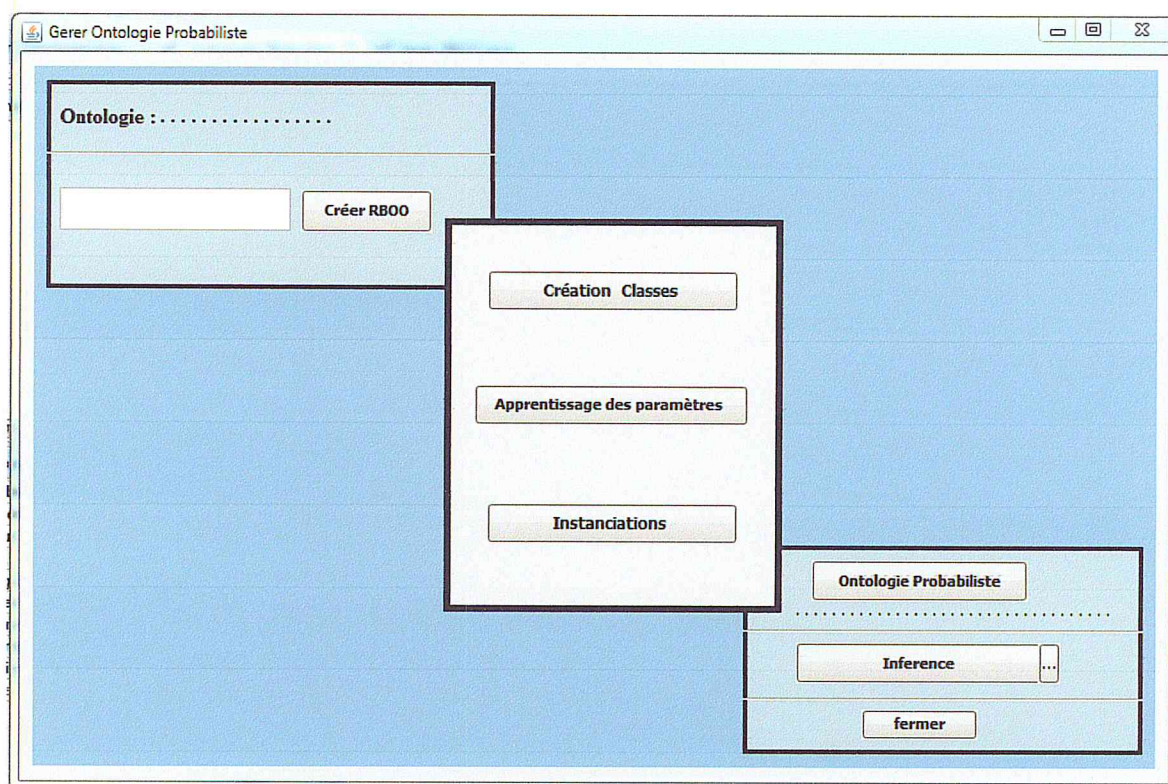


Figure 25 : interface de la création de RBOO.

Elle permet de créer le réseau bayésien orienté objet (RBOO), une fois qu'on saisi le nom de notre RBOO, nous aurons la main pour créer ses classes, et aussi nous allons remplir les paramètres de chaque classe, et à la fin nous allons définir les instanciations.

#### 3.2.1. Créer RBOO

Elle permet de donner un nom et créer notre réseau bayésien orientée objet.

#### 3.2.2. Création Classes

Si on clique sur ce bouton on aura l'interface suivante :

**Gérer les classes :** Cette interface consiste à créer les classes que nous pouvons ajouter à notre RBOO, et pour chaque classe ajoutée nous créons les nœuds associés et définir leur type, et à la fin on va concevoir la liaison entre les nœuds.

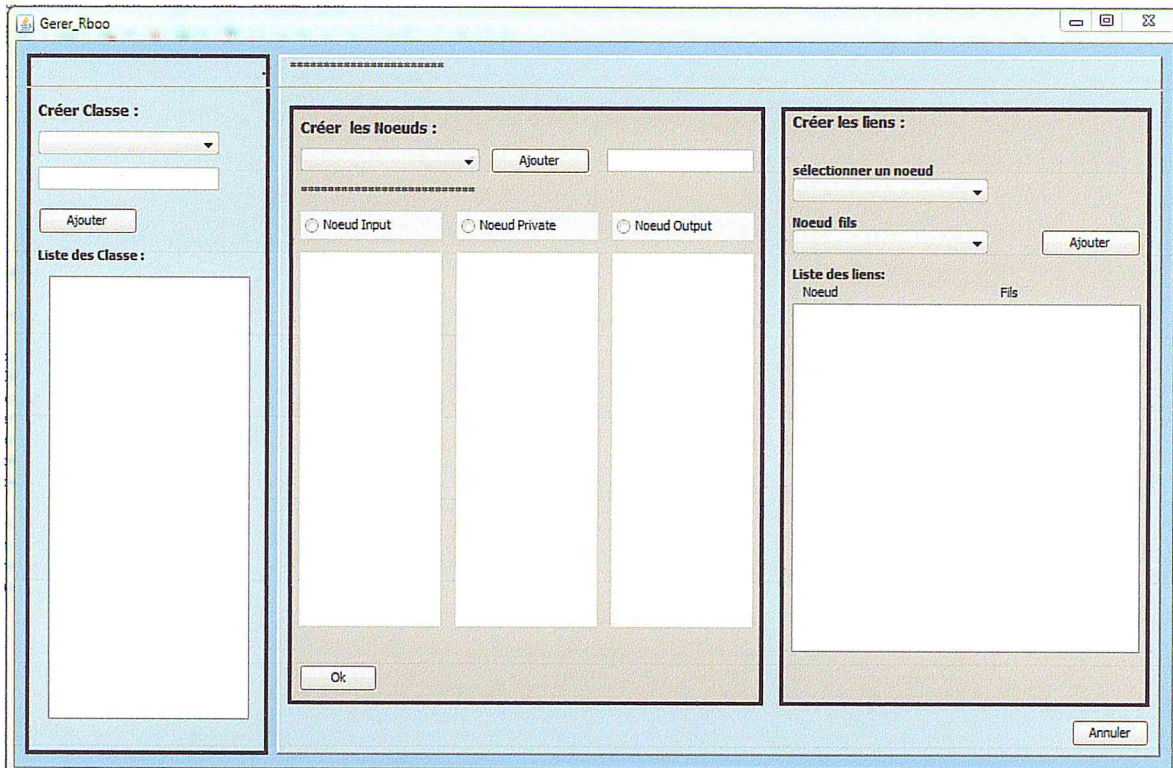
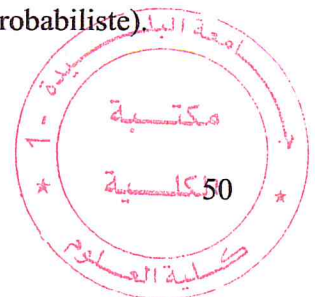


Figure 26 : interface de création et gérer des classes.

- Pour ajouter une classe à notre RBOO on doit saisir le nom de la classe ou la sélectionner à partir des concepts de l'ontologie formelle, après on clique sur le bouton ajouter , pour que notre classe s'ajoute à l'ensemble des classes et elle apparaîtra dans la liste des classes.
- Après l'ajout de la classe on va définir les nœuds de celle ci, et on saisi le nom du Nœud ou on choisi un nœud parmi l'ensemble des propriétés de l'ontologie formelle qui est liée à la classe déjà ajoutée. Ainsi on définit leur types (input, private, output) ensuite on clique sur ajouter pour qu'il s'ajoute à l'ensemble des nœuds de sa classe et le mettre dans l'ensemble de son type.
- Après l'ajout des nœuds, on va créer la liaison eux, avant de cliquer sur le bouton ajouter on sélectionne deux nœuds. Où:
  - Si le type de nœud est input son fils est nœud output ou private.
  - Si le type de nœud est private son fils est nœud output ou private.
  - Si le type de nœud est output son fils nœud output.

Une fois qu'on termine la création d'une classe et ses nœuds, on aura la main pour ajouter une autre classe ou pour retourner à l'interface précédente (gère ontologie probabiliste).



### 3.2.3. Apprentissage des paramètres

Une fois qu'on a terminé la création des classes et ses nœuds on aura la main pour passer à cette étape qui consiste à donner une valeur probabiliste à chaque nœud.

Si on clique sur ce bouton (Apprentissage Des Paramètres) on aura l'interface suivante :

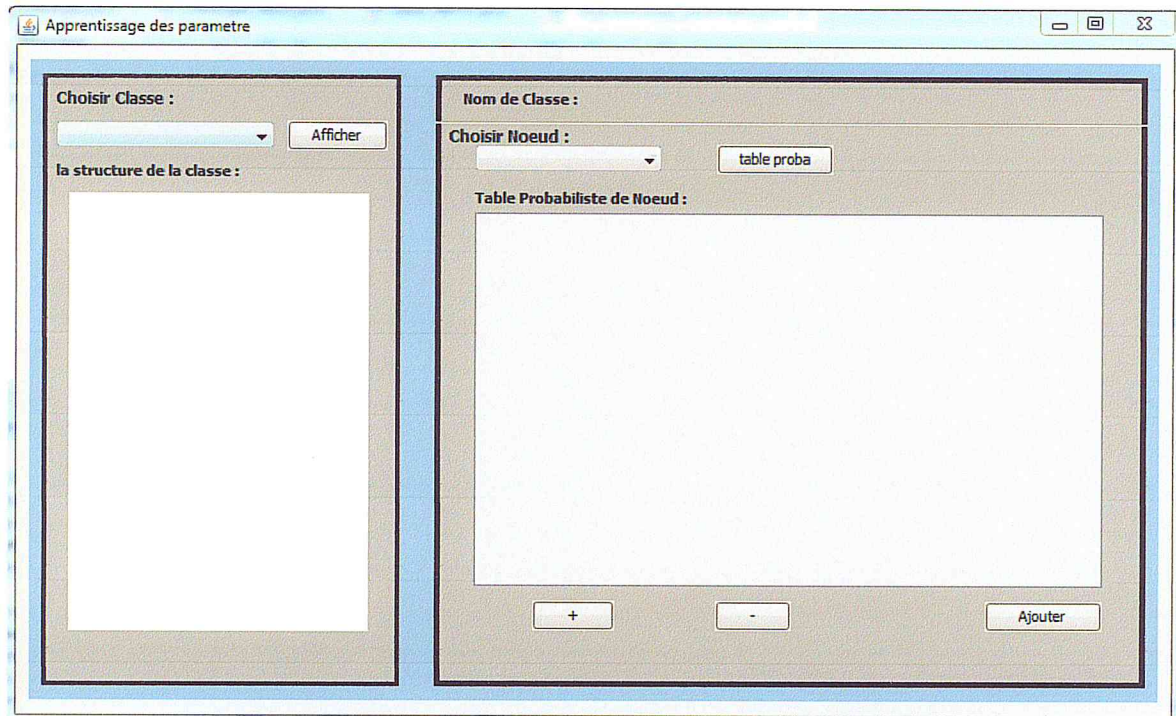


Figure 27 : Interface de l'Apprentissages des paramètres.

- **Afficher** : On va d'abord choisir une classe parmi l'ensemble des classes de notre RBOO. Ce qui vous permet d'extraire tous les nœuds de cette classe et les liaisons entre eux.
- **Table proba** : on sélectionne un nœud parmi l'ensemble des nœuds et on extrait la table probabiliste associée, et modifier les valeurs probabilistes.
- **Le bouton +** : pour ajouter une state (état).
- **Le bouton -** : pour supprimer une state (état).
- **Ajouter** : A la fin on clique sur le bouton ajouter pour sauvegarder ses valeurs.
- 

### 3.2.4. Instanciation

Cette interface nous permet de créer la structure globale de notre RBOO (les instanciations des classes et les liaisons entre eux).

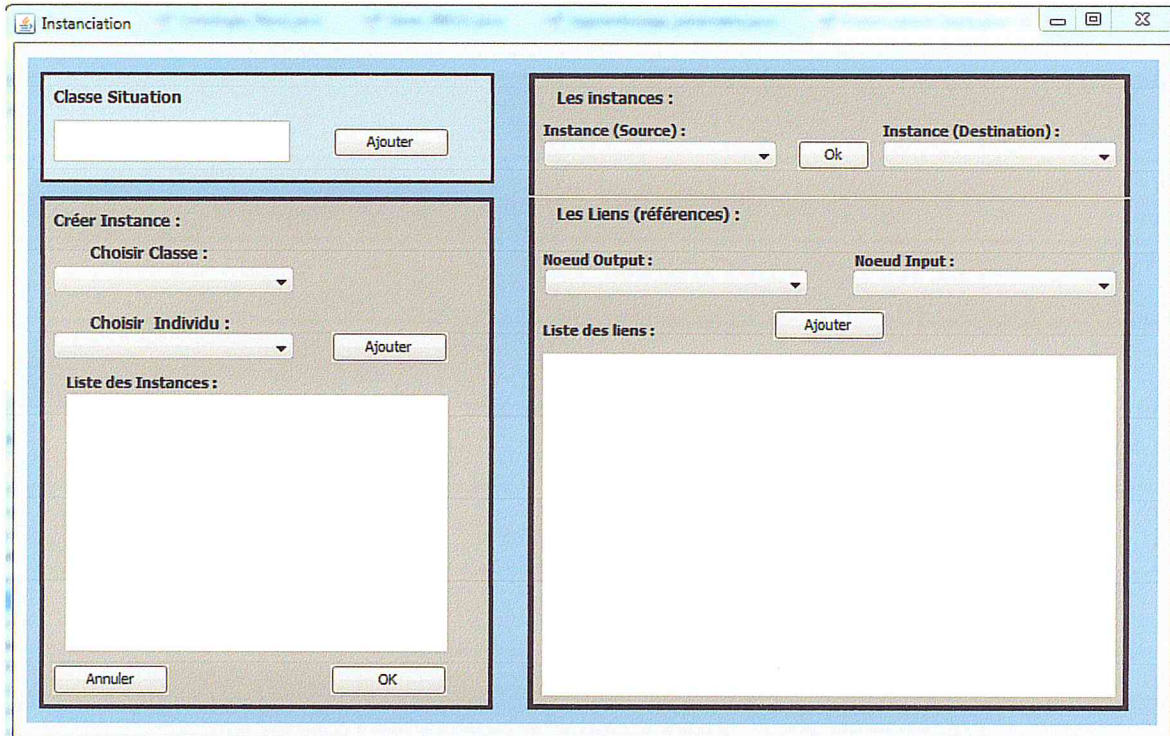


Figure 28 : Interface pour Définir Les Instanciation.

- D'abord on ajoute une classe à notre RBOO (classe particulière qui regroupe toutes les instances des classes prédéfinies). on doit saisir le nom de la classe, après on clique sur le bouton ajouter, pour que notre classe s'ajoute à l'ensemble des classes.
- Pour ajouter une instance, on doit spécifier la classe, après on choisit une instance à partir des individus de l'ontologie formelle qui la relie à la classe choisie, et afficher une liste de tous les individus ajoutés.

Vous pouvez ajouter plusieurs instances pour une seule classe.

- lorsqu'on termine la création des instances on aura la main pour ajouter la liaison entre ces instances. Et pour cela on doit choisir deux instances, où il existe au moins un Object propriété entre eux dans l'ontologie formelle, après cette étape on sélectionne les nœuds pour faire la liaison entre eux, et cela en choisissant pour la première instance un nœud Output parmi l'ensemble des nœuds Output de cette instance, et choisir pour la deuxième instance un nœud input, à condition que ces nœuds doivent avoir les mêmes états.

**Remarque:** Chaque nœud (output, input) n'a qu'une seule liaison.

### 3.2.5. Ontologie probabiliste

Lorsqu'on termine avec l'instanciation (structure globale), on va intégrer automatiquement le RBOO crée avec l'ontologie formelle pour obtenir à la fin une ontologie probabiliste.

### 3.2.6. Inférence

Cette interface nous aide à gérer l'inférence de notre RBOO. L'inférence consiste à entrer les observations pour certains nœuds (les évidences), c.à.d. pour chaque nœud observé on doit sélectionner un de ses états, ensuite exécuter un algorithme d'inférence afin de mettre à jour les tables de probabilités de chaque nœud.

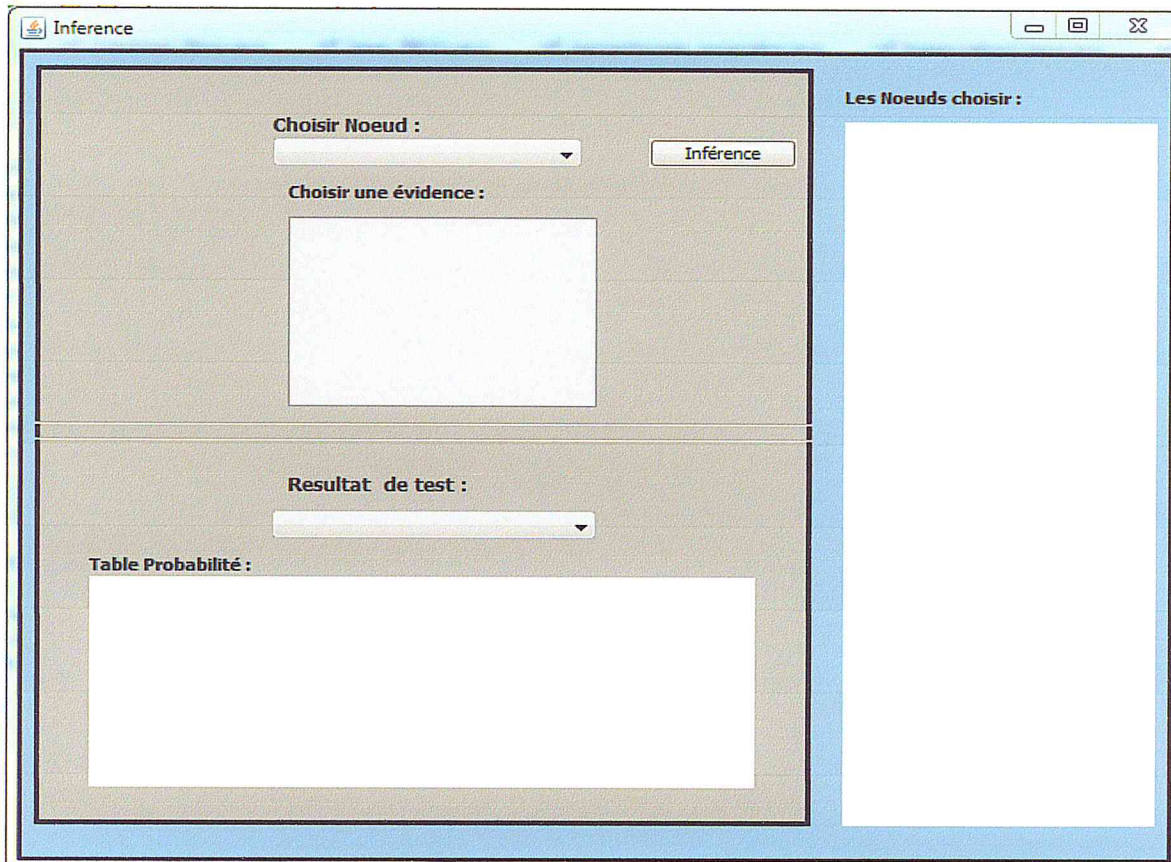


Figure 29 : Interface pour gérer l'inférence.

Chaque nœud a sa probabilité conditionnelle initiale.

- **Inférence** : On va choisir un nœud à partir de tous les nœuds prédéfinis (observés) en RBOO (instanciation de classe), et après on doit entrer les évidences (Sélectionner un état). Puis lorsqu'on clique sur le bouton un algorithme d'inférence doit s'exécuter en prenant en considération les évidences.

### 4. Conclusion

Nous avons vu dans ce chapitre les différents outils et technologies nécessaires pour la réalisation de notre système tels que l'environnement de développement, les langages de programmation, les bibliothèques open source utilisées...etc. Nous avons présenté clairement et en détail toutes les interfaces nécessaires, en utilisant des captures d'écran.

# **Chapitre 5 :**

## **Test du système**

### 1. Introduction

Dans cette dernière partie nous allons effectuer un test sur notre système implémenté. La méthode que nous avons utilisé pour tester notre système, c'est d'effectuer les étapes de notre modélisation pour créer une ontologie probabiliste utilisable pour soutenir le diagnostic du patient, qui présente des symptômes et les causes spécifiques pouvant augmenter la probabilité qu'il soit atteint du cancer du foie ou non.

On outre, notre système est générique, il est utilisable dans un autre domaine qui portent l'incertitude tel que l'industrie, mécanique, biologie,...etc.

### 2. Motivation étude de cas

L'intelligence artificielle a connu un succès largement supérieur dans divers domaines. Mais malgré tout cela, elle a souffert de quelques problèmes, y compris le problème de la représentation des connaissances. Et c'est pour cela que les experts et les chercheurs dans le développement de l'intelligence artificielle ont adopté l'ontologie comme solution idéale à ce problème. De part son rôle important, l'ontologie a connu une grande expansion dans divers domaines, y compris dans le domaine médical en contribuant à trouver des traitements à certaines maladies et en facilitant la prise d'une meilleure décision médicale.

Dans ce contexte, notre étude se focalisera dans le domaine médical et plus particulièrement le cancer du foie pour résoudre le manque de certitude et ainsi faciliter d'établir un bon diagnostic aux patients.

### 3. Cancer du foie

#### 3.1 Définition

Le cancer du foie correspond à la formation d'une tumeur dans les tissus du foie. Il existe différents types de cancers du foie selon la nature des cellules cancéreuses. Le carcinome hépatocellulaire (CHC) ou d'hépatocarcinome en est la forme la plus fréquente et représente 90 % de tous les cancers du foie. Le carcinome hépatocellulaire apparaît dans les hépatocytes, les principales cellules du foie [28].

D'autres formes plus rares du cancers primitifs du foie peuvent se développer à partir des cellules des canaux biliaires (cholangiocarcinome) ou beaucoup plus rarement, des vaisseaux sanguins (hémangioendothéliome épithélioïde).



### 3.2. La propagation du cancer du foie

Le cancer du foie représente le sixième cancer le plus fréquent à travers le monde. En Europe, on estime que 10 hommes sur 1000 et 2 femmes sur 1000 développent un cancer du foie à un moment donné de leur vie. À l'échelle de la planète, le cancer du foie est beaucoup plus fréquent en Asie du Sud-Est et en Afrique de l'Ouest. Cela est principalement dû au fait que l'infection par le virus de l'hépatite B, est plus fréquente dans ces deux régions. Aux États-Unis et en Europe du Sud, le virus de l'hépatite C est souvent une cause d'apparition du cancer du foie.

En 2008, environ 40 000 hommes et 20 000 femmes ont été diagnostiqués d'un cancer du foie en Europe.

L'âge moyen de diagnostic se situe entre 50 et 60 ans, mais en Asie et en Afrique, il est généralement compris entre 40 et 50 ans [28].

### 3.3. Facteurs de risque

Le cancer du foie est causé par une combinaison de différents facteurs de risque. Certains des facteurs associés au cancer du foie ne peuvent être modifiés [29].

Les facteurs de risque connus pour le cancer du foie sont :

- Les infections chroniques par le virus de l'hépatite B (VHB) ou le virus de l'hépatite C (VHC).
- L'abus d'alcool à long terme.
- La consommation du tabac.
- L'HÉMOCHROMATOSE : L'hémochromatose est une maladie liée à un excès de fer dans l'organisme.
- La stéatose hépatique.

Il y a des facteurs liés au patient : (sexe, l'âge, histoire de famille).

### 3.4. Symptômes possibles d'un cancer du foie

Les principaux symptômes pouvant être liés à un cancer du foie sont les suivants [28]:

- Une perte de poids inexplicquée
- De la fatigue
- Une perte d'appétit ou la sensation d'avoir trop mangé après un repas léger
- Des nausées ou des vomissements
- De la fièvre
- Une augmentation du volume du foie.

- Une augmentation du volume de la rate.
- Des douleurs à l'abdomen ou à proximité de l'omoplate droite
- Des démangeaisons

Tous ces symptômes peuvent être causés par d'autres maladies ou ne peuvent être observés qu'à un stade avancé du cancer du foie. Et doit donc faire l'objet d'examens supplémentaires.

### 3.5. Le diagnostic

Le diagnostic du cancer du foie est basé sur les examens suivants :

- Un examen clinique.
- Une prise de sang.
- Un examen radiologique.

## 4. Ontologie du cancer du foie

Les ontologies médicales fournissent aux praticiens une plate-forme pour une prise de décision adéquate dans leur diagnostic, en tenant compte des différents aspects liés aux patients, aux maladies, aux remèdes,...etc. Et permet aux praticiens de rechercher l'information, de la traiter et de communiquer entre eux afin de donner le meilleur diagnostic.

Dans notre projet nous avons utilisé une ontologie de domaine médicale : ontologie du cancer du foie (**ontologie liver cancer**).

A travers ce que nous avons parlé au-dessus et selon une documentation sur l'ontologie du cancer du foie [33]. Nous avons implémenté cette ontologie à l'aide de l'outil appelé protégé version 4.3 (Protégé permet le support d'OWL et prend en charge les ontologies protégées dans un format tel que les schémas RDF, OWL et XML.). Cette ontologie de la maladie du foie (liver cancer) proposée représente l'information dans les deux classes :

- Patient.
- Liver cancer.

La classe liver cancer contient deux sous-classes, à savoir:

- RiskFactors.
- Symptômes.

La figure suivante montre une présentation graphique de l'ontologie Liver Cancer:

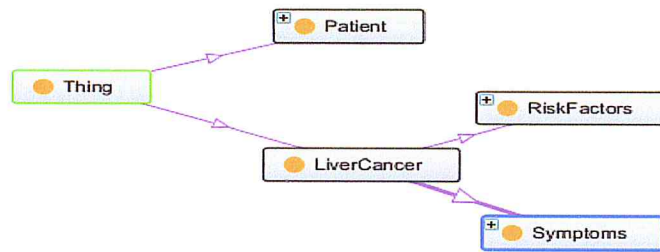


Figure 30 : la taxonomie de l'ontologie Liver Cancer [33].

**La description de l'ontologie Liver Cancer:**

Classe	Datat propretie	Object Propertie	individual
Patient	hasName :String hasAge :String FamilyHistory :Boolean hasGender :String hasWeight : Sring	hasCausedBy(RiskFactors) hasSymptoms(Symptoms)	P
Liver Cancer			
RiskFactors	hasAge :String FamilyHistory :Boolean hasGender :String Alcohol :Boolean CirrhoseHapatique : Boolean Diabetis : Boolean HepatiteB : Boolean HepatiteC : Boolean Infection : Boolean OverWeight : String		RF

Symptômes	EnlargerLiver : Boolean Fatigue : Boolean Jandince : Boolean JointPain : Boolean PainAbdomen : Boolean WeightLess : String		Sy
-----------	---	--	----

La figure suivante représente la vision générale sur l'ontologie de test (Liver Cancer) :

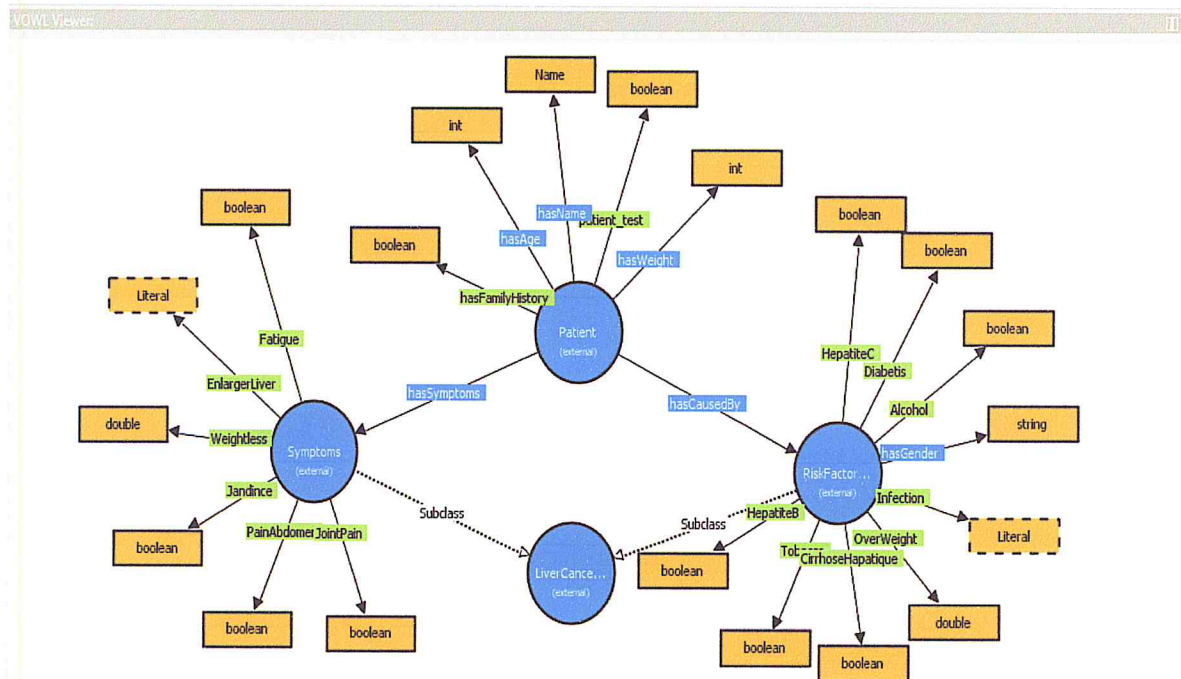


Figure 31 : vision générale sur l'ontologie Liver Cancer.

## 5. La création de l'ontologie probabiliste

Dans cette partie nous montrons les différentes étapes qu'on a suivi pour terminer la création de l'ontologie probabiliste.

### 5.1. Chargement de l'ontologie Liver Cancer

La première étape pour commencer notre test, est de télécharger notre ontologie de test liver cancer.

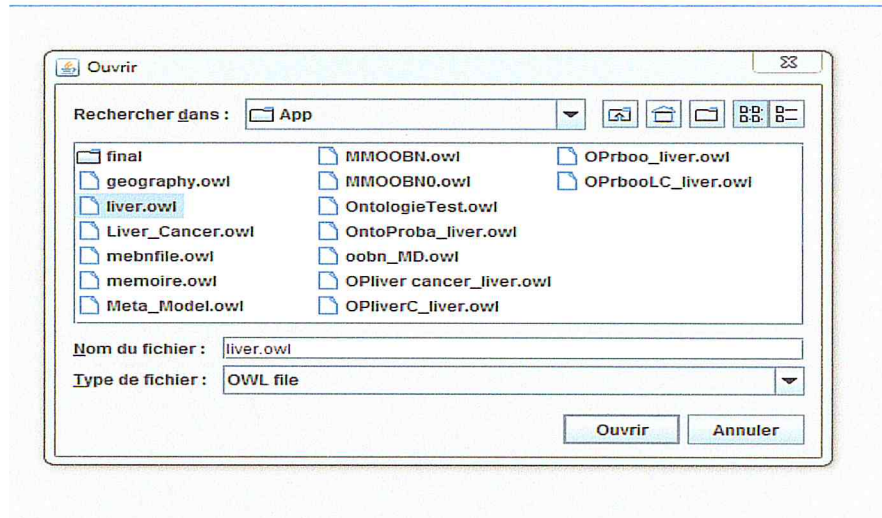


Figure 32 : chargement de l'ontologie liver cancer

## 5.2. Extraction des TBox et ABox

Dans cette étape on fait l'extraction des différents composants de notre ontologie : les concepts, les propriétés des données, les propriétés des objets et les individus.

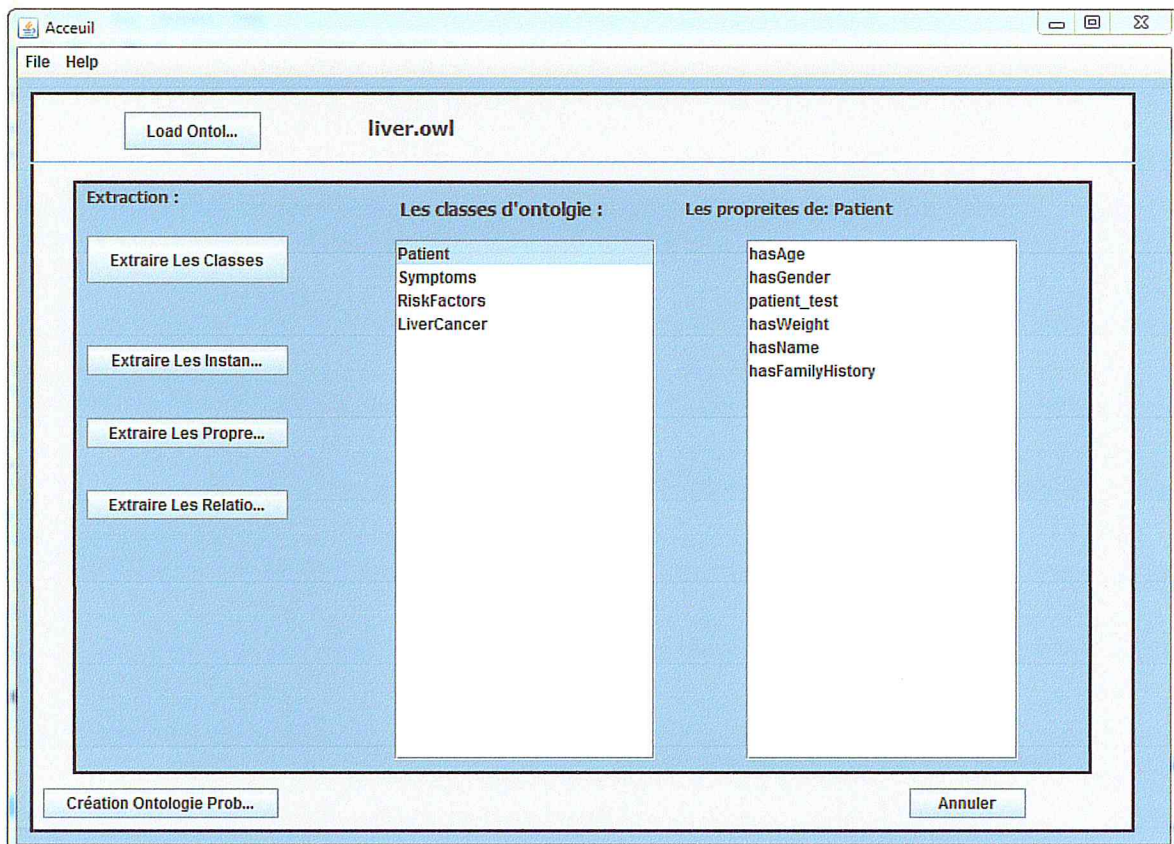


Figure 33 : Extraction des propriétés des données pour la classe Patient.

### 5.3. Créer le réseau bayésien orienté objet

Dans cette étape nous faisons la création du réseau bayésien orienté objet correspond à l'ontologie liver cancer. Selon quatre étapes:

Il faut d'abord donner un nom pour le réseau bayésien orienté objet à créer :

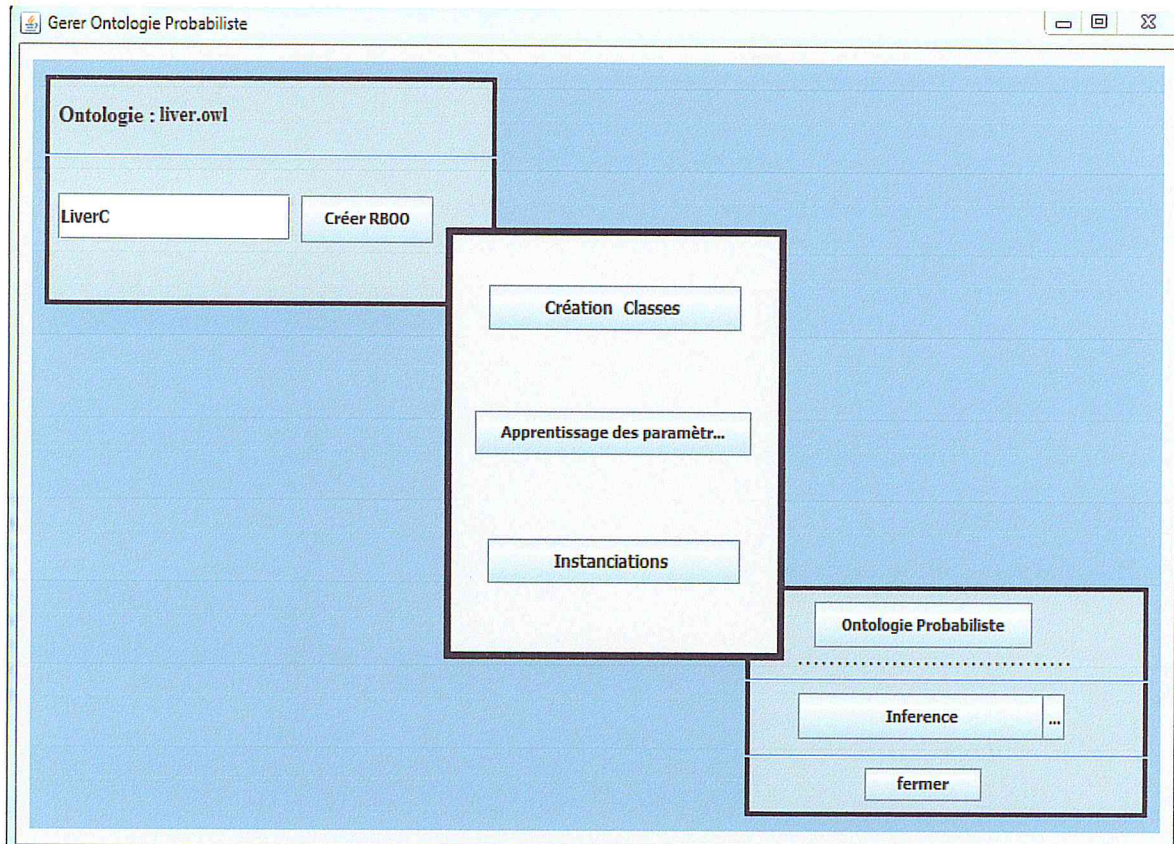


Figure 34: Donner un nom pour le réseau bayésien orienté objet.

### 5.4. Création des sous structures (les classes)

Dans cette étape nous allons créer les sous structures de notre réseau bayésien orienté objet, et pour cela il faut commencer par créer les classes (1), ces nœuds (2), et les liaisons entre les nœuds (3).

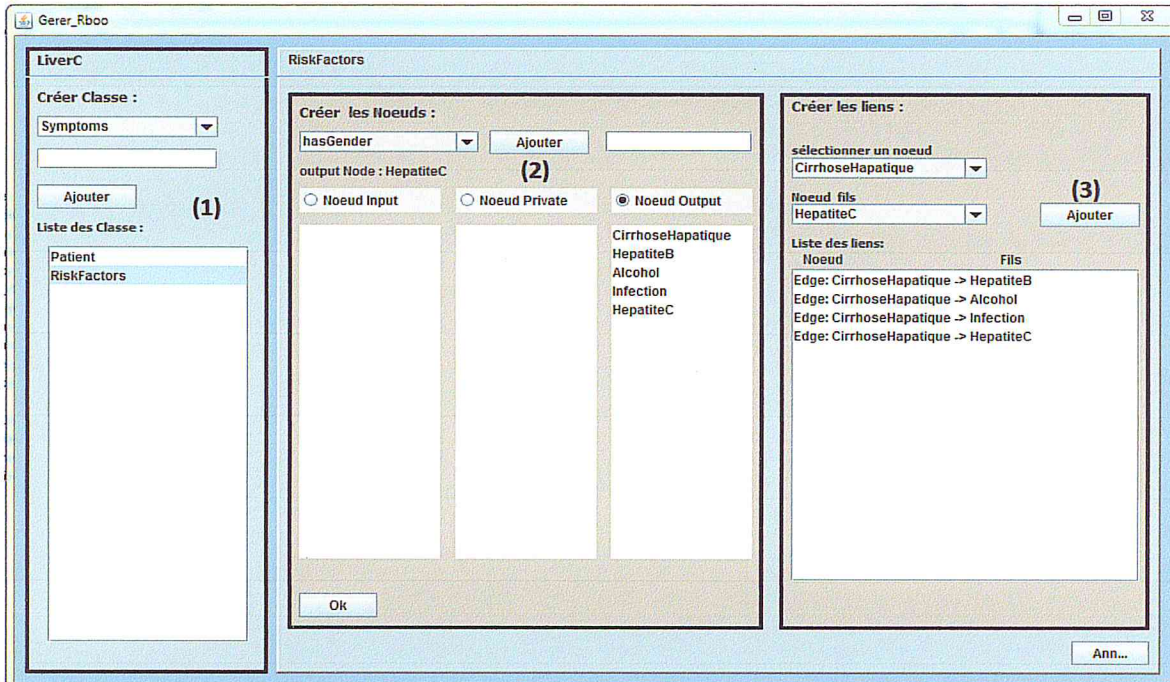


Figure 35 : création la classe RiskFactors (1), ces nœuds (2) et la liaison entre les nœuds(3).

### 5.5. Apprentissage des paramètres

Après la fin de la création des sous structures, il suffit de remplir toutes les tables de probabilité pour chaque nœud.

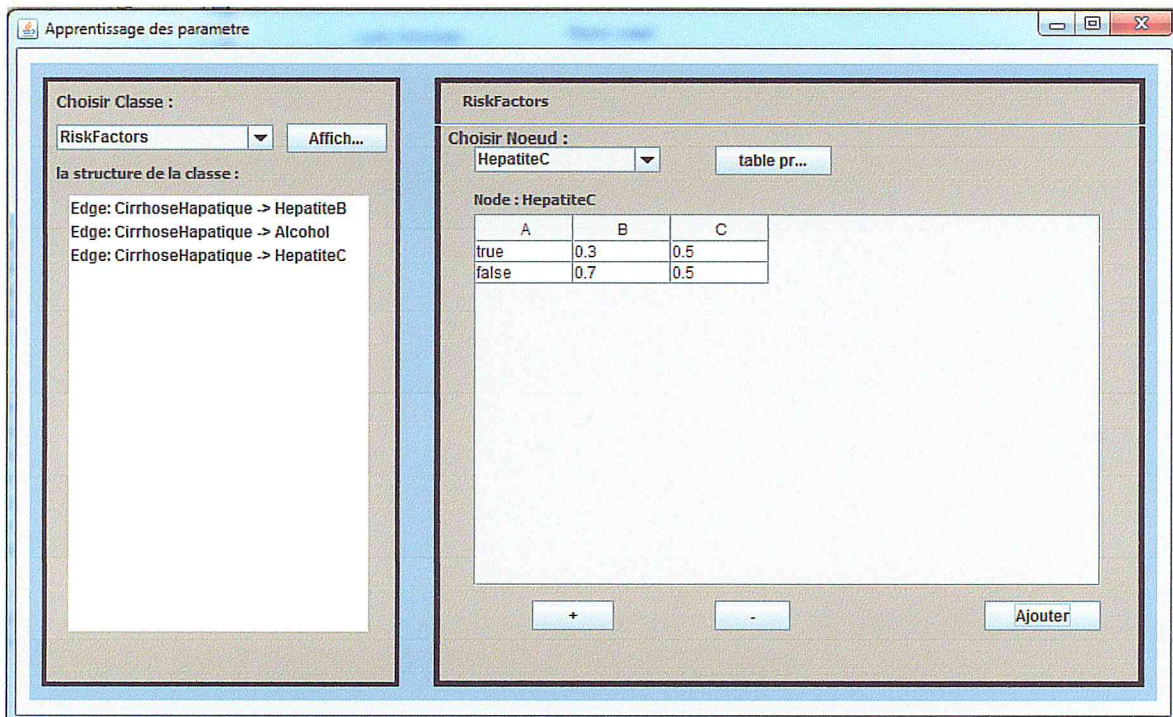


Figure 36 : Table de probabilité du nœud HepateC de la classe RiskFactors.

### 5.6. Création de la structure globale (instanciation)

Après la fin de l'apprentissage des paramètres, nous allons créer la structure globale de notre RBOO, et pour cela il faut commencer par créer la classe situation qui contient les instanciations des autres classes (1), créer les instances des classes de notre RBOO sauf classe situation (2), créer les liaisons entre les instances à partir des nœuds (input et output) (3).

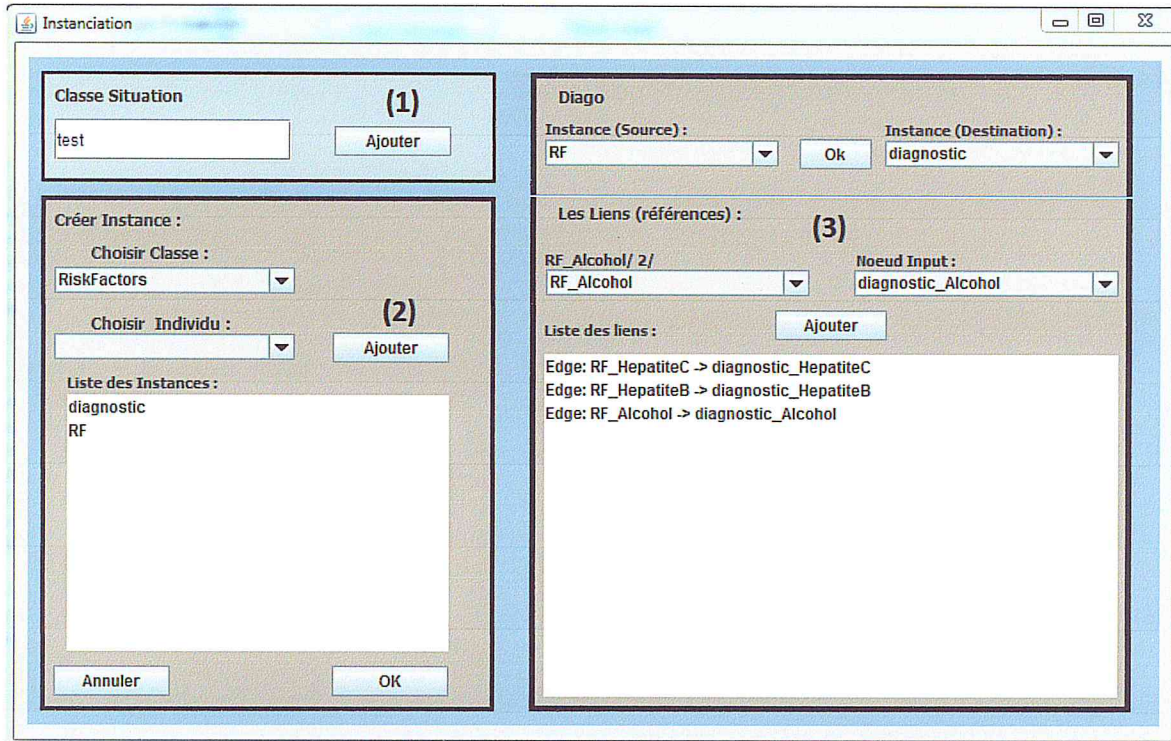


Figure 37 : création de la structure globale de notre RBOO.

### 5.7. L'intégration

Après avoir terminé de créer de notre réseau bayésien orienté objet, nous devons intégrer ce RBOO avec son ontologie liver cancer. A la fin nous obtenons une ontologie probabiliste.

### 5.8. L'inférence

Après l'instanciation de notre RBOO, nous allons effectuer l'inférence. Pour cela il faut commencer par choisir les évidences (états) observer de différents aspects liés aux patients et maladie prédéfinie en RBOO. Ensuite exécuter un algorithme d'inférence afin d'établir un bon diagnostic aux patients.



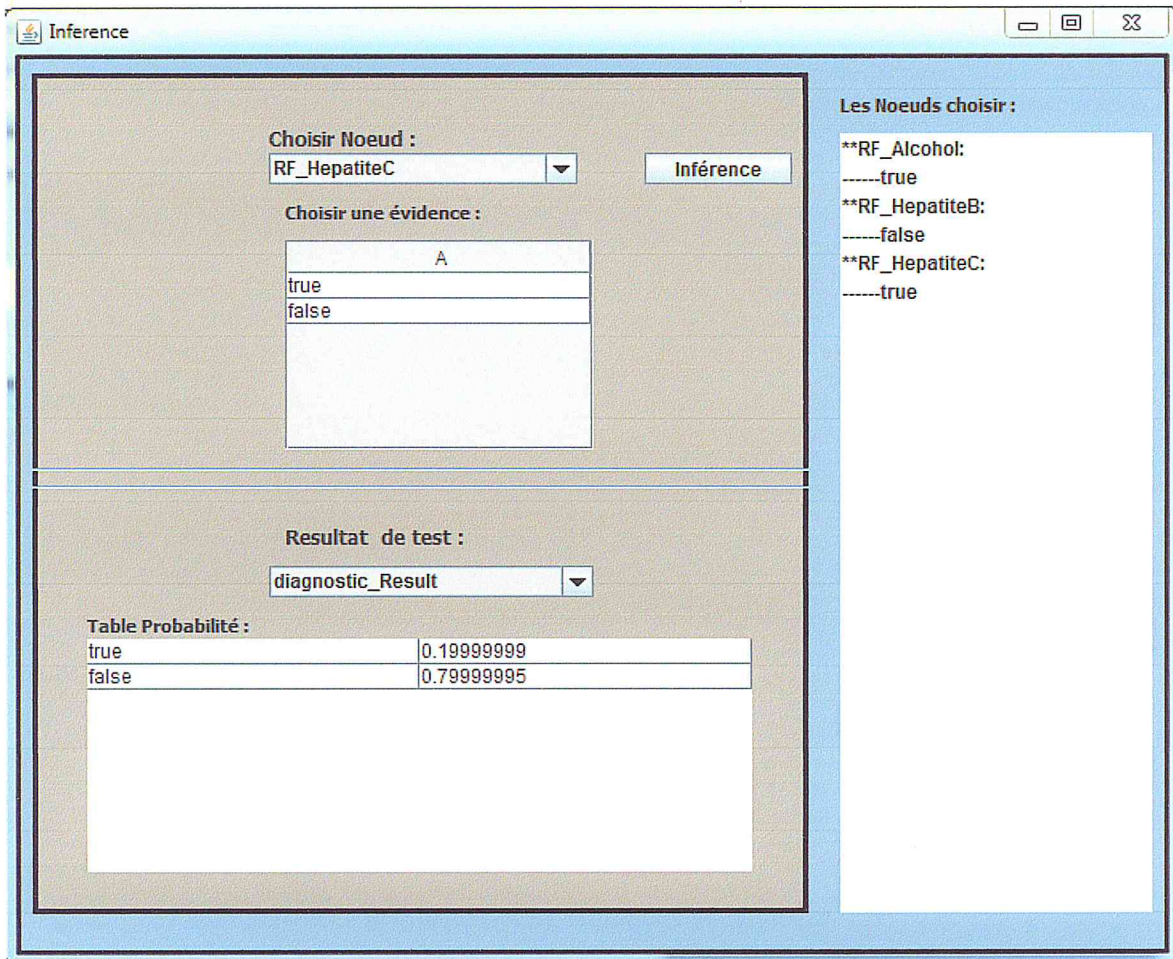


Figure 38 : l'inférence dans notre RBOO.

## 6. Résultat de test

Après l'effectuation des étapes de la création de l'ontologie probabiliste. On a obtenu une ontologie probabiliste du domaine étudié (**Liver Cancer**).

L'ontologie probabiliste obtenue est la combinaison de l'ontologie formelle avec l'ontologie qui représente la méta-ontologie de RBOO.

### 6.1. La hiérarchie de l'ontologie probabiliste obtenue

La hiérarchie des classes de l'ontologie probabiliste

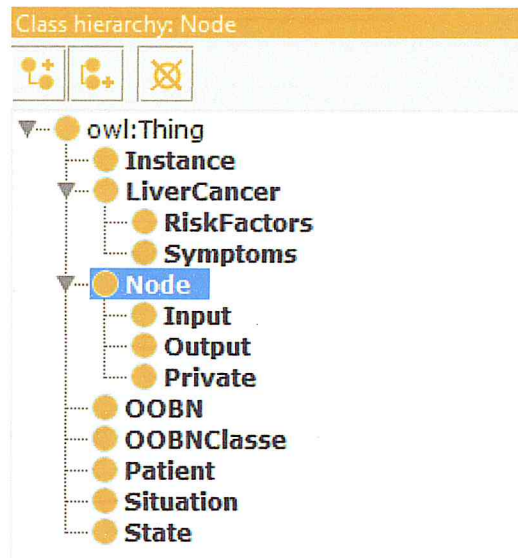


Figure 39 : Hiérarchie des classes

La hiérarchie des propriétés d'objet de L'ontologie probabiliste :

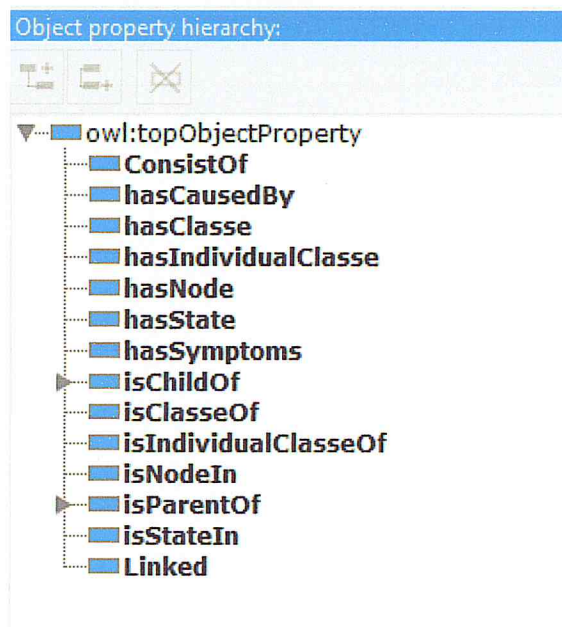


Figure 40 : La hiérarchie des propriétés d'objet.

La hiérarchie des propriétés des données de l'ontologie probabiliste

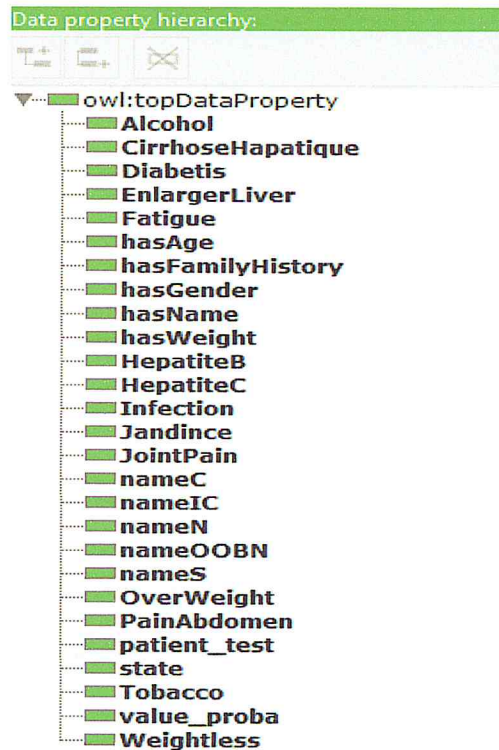


Figure 41 : La hiérarchie des propriétés des données

6.2. Les syntaxes de l'ontologie probabiliste

Après la combinaison nous intégrons le RBOO dans l'ontologie probabiliste .Ce qui suit On a représenté des exemples de syntaxe de l'intégration (ajoute).

- RBOO :

La syntaxe de l'ajout d'un réseau bayésien orienté objet (RBOO) :

```

<owl:NamedIndividual rdf:about="file:/C:/Users/mohamed/Desktop/App/OntoProba_liver.owl#liver cancer">
  <rdf:type rdf:resource="http://www.semanticweb.org/mohamed/ontologies/2017/4/untitled-ontology-45#OOBN"/>
  <untitled-ontology-45:hasClasse rdf:resource="file:/C:/Users/mohamed/Desktop/App/OntoProba_liver.owl#Diagnostic"/>
  <untitled-ontology-45:hasClasse rdf:resource="file:/C:/Users/mohamed/Desktop/App/OntoProba_liver.owl#Patient"/>
  <untitled-ontology-45:hasClasse rdf:resource="file:/C:/Users/mohamed/Desktop/App/OntoProba_liver.owl#RiskFactors"/>
  <untitled-ontology-45:hasClasse rdf:resource="file:/C:/Users/mohamed/Desktop/App/OntoProba_liver.owl#Symptoms"/>
  <untitled-ontology-45:hasClasse rdf:resource="file:/C:/Users/mohamed/Desktop/App/OntoProba_liver.owl#situation"/>
</owl:NamedIndividual>
  
```

Figure 42 : syntaxe du RBOO.

- **Classe :**

La syntaxe de l'ajout d'une Classe :

```
<owl:NamedIndividual rdf:about="file:/C:/Users/mohamed/Desktop/App/OntoProba_liver.owl#Patient">
  <rdf:type rdf:resource="http://www.semanticweb.org/mohamed/ontologies/2017/4/untitled-ontology-45#OOBNClasse"/>
  <untitled-ontology-45:hasIndividualClasse rdf:resource="file:/C:/Users/mohamed/Desktop/App/OntoProba_liver.owl#P1"/>
  <untitled-ontology-45:hasNode rdf:resource="file:/C:/Users/mohamed/Desktop/App/OntoProba_liver.owl#hasAge"/>
  <untitled-ontology-45:hasNode rdf:resource="file:/C:/Users/mohamed/Desktop/App/OntoProba_liver.owl#hasFamilyHistory"/>
  <untitled-ontology-45:hasNode rdf:resource="file:/C:/Users/mohamed/Desktop/App/OntoProba_liver.owl#hasGender"/>
  <untitled-ontology-45:hasNode rdf:resource="file:/C:/Users/mohamed/Desktop/App/OntoProba_liver.owl#hasWeight"/>
  <untitled-ontology-45:isClasseOf rdf:resource="file:/C:/Users/mohamed/Desktop/App/OntoProba_liver.owl#liver cancer"/>
  <untitled-ontology-45:nameC rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Patient</untitled-ontology-45:nameC>
</owl:NamedIndividual>
```

Figure 43 : Syntaxe de la classe.

- **Nœud :**

La syntaxe de l'ajout d'un Nœud :

```
<owl:NamedIndividual rdf:about="file:/C:/Users/mohamed/Desktop/App/OntoProba_liver.owl#HepatitisB">
  <rdf:type rdf:resource="http://www.semanticweb.org/mohamed/ontologies/2017/4/untitled-ontology-45#Input"/>
  <rdf:type rdf:resource="http://www.semanticweb.org/mohamed/ontologies/2017/4/untitled-ontology-45#Node"/>
  <rdf:type rdf:resource="http://www.semanticweb.org/mohamed/ontologies/2017/4/untitled-ontology-45#Output"/>
  <untitled-ontology-45:hasState rdf:resource="file:/C:/Users/mohamed/Desktop/App/OntoProba_liver.owl#no"/>
  <untitled-ontology-45:hasState rdf:resource="file:/C:/Users/mohamed/Desktop/App/OntoProba_liver.owl#yes"/>
  <untitled-ontology-45:isNodeIn rdf:resource="file:/C:/Users/mohamed/Desktop/App/OntoProba_liver.owl#RiskFactors"/>
  <untitled-ontology-45:isNodeIn rdf:resource="file:/C:/Users/mohamed/Desktop/App/OntoProba_liver.owl#diagno"/>
  <untitled-ontology-45:nameN rdf:datatype="http://www.w3.org/2001/XMLSchema#string">HepatitisB</untitled-ontology-45:nameN>
  <untitled-ontology-45:value_proba rdf:datatype="http://www.w3.org/2001/XMLSchema#string">0.1,0.9,1.0,0.0</untitled-ontology-45:value_proba>
  <untitled-ontology-45:value_proba rdf:datatype="http://www.w3.org/2001/XMLSchema#string">1.0,0.0</untitled-ontology-45:value_proba>
</owl:NamedIndividual>
```

Figure 44 : syntaxe du Nœud.

- **State :**

La syntaxe de l'ajout d'un state :

```
<owl:NamedIndividual rdf:about="file:/C:/Users/mohamed/Desktop/App/OntoProba_liver.owl#yes">
  <rdf:type rdf:resource="http://www.semanticweb.org/mohamed/ontologies/2017/4/untitled-ontology-45#State"/>
  <untitled-ontology-45:isStateIn rdf:resource="file:/C:/Users/mohamed/Desktop/App/OntoProba_liver.owl#Alcohol"/>
  <untitled-ontology-45:isStateIn rdf:resource="file:/C:/Users/mohamed/Desktop/App/OntoProba_liver.owl#CirrhoseHapatique"/>
  <untitled-ontology-45:isStateIn rdf:resource="file:/C:/Users/mohamed/Desktop/App/OntoProba_liver.owl#EnlargerLiver"/>
  <untitled-ontology-45:isStateIn rdf:resource="file:/C:/Users/mohamed/Desktop/App/OntoProba_liver.owl#Fatigue"/>
  <untitled-ontology-45:isStateIn rdf:resource="file:/C:/Users/mohamed/Desktop/App/OntoProba_liver.owl#HepatitisB"/>
  <untitled-ontology-45:isStateIn rdf:resource="file:/C:/Users/mohamed/Desktop/App/OntoProba_liver.owl#Weightless"/>
  <untitled-ontology-45:isStateIn rdf:resource="file:/C:/Users/mohamed/Desktop/App/OntoProba_liver.owl#fin_test"/>
  <untitled-ontology-45:isStateIn rdf:resource="file:/C:/Users/mohamed/Desktop/App/OntoProba_liver.owl#hasFamilyHistory"/>
  <untitled-ontology-45:state rdf:datatype="http://www.w3.org/2001/XMLSchema#string">yes</untitled-ontology-45:state>
</owl:NamedIndividual>
```

Figure 45 : syntaxe du State.

- **Situation :**

```
<owl:NamedIndividual rdf:about="file:/C:/Users/mohamed/Desktop/App/OntoProba_liver.owl#situation">
  <rdf:type rdf:resource="http://www.semanticweb.org/mohamed/ontologies/2017/4/untitled-ontology-45#COBNClasse"/>
  <rdf:type rdf:resource="http://www.semanticweb.org/mohamed/ontologies/2017/4/untitled-ontology-45#COBNGlobal"/>
  <untitled-ontology-45:isClassOf rdf:resource="file:/C:/Users/mohamed/Desktop/App/OntoProba_liver.owl#Liver cancer"/>
  <untitled-ontology-45:nameC rdf:datatype="http://www.w3.org/2001/XMLSchema#string">situation</untitled-ontology-45:nameC>
  <untitled-ontology-45:nameCG rdf:datatype="http://www.w3.org/2001/XMLSchema#string">situation</untitled-ontology-45:nameCG>
</owl:NamedIndividual>
```

Figure 46 : Syntaxe de Situation.

- **Instance :**

```
<owl:NamedIndividual rdf:about="file:/C:/Users/mohamed/Desktop/App/OntoProba_liver.owl#Sy">
  <rdf:type rdf:resource="http://www.semanticweb.org/mohamed/ontologies/2017/4/untitled-ontology-45#IndividualClasse"/>
  <untitled-ontology-45:Linked rdf:resource="file:/C:/Users/mohamed/Desktop/App/OntoProba_liver.owl#diagnostic"/>
  <untitled-ontology-45:LinkedBy rdf:resource="file:/C:/Users/mohamed/Desktop/App/OntoProba_liver.owl#EnlargerLiver"/>
  <untitled-ontology-45:LinkedBy rdf:resource="file:/C:/Users/mohamed/Desktop/App/OntoProba_liver.owl#Fatigue"/>
  <untitled-ontology-45:LinkedBy rdf:resource="file:/C:/Users/mohamed/Desktop/App/OntoProba_liver.owl#Weightless"/>
  <untitled-ontology-45:isIndividualClasseOf rdf:resource="file:/C:/Users/mohamed/Desktop/App/OntoProba_liver.owl#Symptoms"/>
  <untitled-ontology-45:nameIC rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Sy</untitled-ontology-45:nameIC>
</owl:NamedIndividual>
```

**Figure 47 : syntaxe de l'instance.**

## 7. Conclusion

Dans cette partie nous avons parlé du diagnostic des maladies du cancer du foie à l'aide d'un outil protégé. L'ontologie du cancer du foie a été construite grâce à la réutilisation des outils d'ontologie existants et des données médicales. L'ontologie du cancer du foie est utilisée pour fournir un bon diagnostic. Et nous avons fait un test à notre système avec cette ontologie et les résultats obtenus on les a notés en utilisant des captures d'écran.

### Conclusion générale :

L'ontologie est utilisée, depuis plusieurs années, dans l'Ingénierie des Connaissances (IC) et l'Intelligence Artificielle (IA) pour structurer les concepts d'un domaine. Elle est utile pour partager des connaissances, créer un consensus, construire des systèmes à base de connaissances. Cependant, les ontologies classiques ne fournissent pas un soutien adéquat à l'incertitude.

Pour faire face à de tels problèmes. Ce mémoire présente une méthode qui porte les procédures générales pour la modélisation de l'incertitude dans une ontologie à des domaines en utilisant les réseaux bayésiens orientés objets (RBOO).

L'ontologie probabiliste que nous avons créé, est basée sur les réseaux bayésiens orientés objet. Elle permet de modéliser l'incertitude.

Il est prouvé que les réseaux bayésiens orientés objet (RBOO) sont des méthodes fiables pour modéliser l'incertain et pour faire des prédictions en se basant sur des faits réels même en présence d'incertitudes autour de ces faits. Nous avons conclu que le RBOO peut aider dans le web sémantique afin de résoudre le problème de l'incertitude.

Nous avons proposé pour cette modélisation un processus qui porte les étapes suivantes :

- **L'importation de l'ontologie formelle OWL:** Choisir quelle ontologie on veut traiter.
- **Extraire les partie TBox et ABox :** Extraire les classes, les propriétés, et les individus.
- **La construction des sous-structures :** Construire le RBOO et ses classes par l'extraction des concepts d'une ontologie. Et aussi la création des nœuds avec leur types et on va définir la liaison entre eux.
- **La construction des paramètres :** L'apprentissage des paramètres consiste à remplir la table probabiliste de chaque nœud déjà défini.
- **La construction de la structure globale:** C'est la construction de la structure globale du réseau bayésien orienté objet.
- En plus, nous pouvons calculer l'inférence de notre RBOO créé.
- **L'intégration du RBOO dans l'ontologie:** C'est la construction d'une ontologie probabiliste.

Nous avons implémenté notre système qui est générique et nous l'avons testé avec une ontologie d'un domaine médical 'Liver-cancer', ainsi nous avons montré les différentes interfaces nécessaires pour la construction de notre ontologie probabiliste.



# **Annexe**

**Service de Chirurgie Oncologique du Centre Anti Cancer de Blida**

**Chirurgie Hépato-Biliaire et Pancréatique**

Dr M.R AITARAB

Dans la dernière partie nous avons effectué un test sur notre système implémenté. Est cela dans le service de chirurgie oncologique du centre anti cancer de Blida unité de chirurgie hépato-biliaire et pancréatique, ou nous avons recueillie les données de deux patients (01 cas présentant un CHC (**Cancer du foie**) / 01 témoin avec un foie normal)

Patient 1 : patient qui ne présentant pas de cancer hépatique avec un foie normale sur les examens morphologiques avec des marqueurs tumoraux négatifs

Patient 2 : patient qui présente une tumeur du foie gauche dont l'étude histologique de la biopsie écho-guidée est revenue en faveur d'un carcinome hépato cellulaire bien différencié CHC. Avec des marqueurs tumoraux très élevés (alpha fœto protéines 10x la normale).

## Data set

Classe	Propretés	States	Test cases	
<u>Patient</u>	name		Patient 1	Patient 2
	Age	Young		
		Adult	X	
		Old		X
	Family History	Yes		
		No	X	X
	Gender	Man	X	X
		Woman		
	Weight	(Kg)	74	56
<u>Risk Factors</u>	Age	Young		
		Adult	X	

	Old		X
Family History	Yes		
	No	X	X
Gender	Man		X
	Woman	X	
Alcohol	Yes		
	No	X	X
Cirrhose Hapatique	Yes		X
	No	X	
Diabetes	Yes	X	X
	No		
Hepatitis B	Yes		X
	No	X	
Hepatitis C	Yes		
	No	X	X
Infection	Yes		
	No	X	X
Over Weight	Yes		
	No	X	X
<u>Symptômes</u>	Enlarger Liver	Yes	X
		No	X
Fatigue	Yes	X	X
	No		
Jandince	Yes		X
	No	X	
Joint Pain	Yes		
	No	X	X
Pain Abdomen	Yes		X
	No	X	
Weight Less	Yes		X
	No	X	

diagnostique	Yes	X
	No	X

### **Bibliographies:**

- [1] T. Berners-Lee., J. Hendler and O. Lassila (2001). The Semantic Web. Scientific American Magazine, 284(5), 34-43.
- [2] T. Berners-Lee. (2003). Standards, Semantics and Survival. SIIA Upgrade: pp. 6–10.
- [3] J. Baget. E., Canaud. J. Euzenat and M.-S. Hacid. (2004). Les langages du Web Sémantique, série de la Revue Information - Interaction - Intelligence (I3), 4(1), Cépaduès, Toulouse, pp. 21-43.
- [4] L. Lassila, R. R. Swick (1999). Resource description framework (rdf) model and syntax specification. World Wide Web Consortium W3C Recommendation 22 February 1999 / W3C.
- [ 5] D. L. McGuinness, F. van Harmelen (2004). Owl web ontology language overview. World Wide Web Consortium, Recommendation REC-owlfeatures- 20040210.
- [6] T.R. Gruber. (1993). A translation approach to portable ontology specifications. Knowledge Acquisition, 5(2) :199–220,.
- [7] T.R. Gruber. (1995). Towards Principles for the Design of Ontologies Used for Knowledge Sharing. International Journal Human-Computer Studies, 43(5–6):907– 928.
- [8] M. Uschold, M. King.(1995). Towards a methodology for building ontologies, in Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing, IJCAI'95.
- [9] G. Van Heijst, A. Schreiber and B. Wielinga (1997). Using explicit ontologies in kbs development. International Journal of Human and Computer Studies - Knowledge Acquisition, 46(2/3), 183–292.
- [ 10] 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.
- [11] N. Guarino. Some organizing principles for a unified top-level ontology. In National Conference of the American Association on Artificial Intelligence (AAAI), p. 57–63, Stanford, United-States.
- [12] N. Guarino, A. Gangemi, C. Masolo and A. Oltrari (1995). Understanding top-level ontological distinctions. In Workshop on Basic Ontological Issues in Knowledge Sharing, The International Joint Conference on Artificial Intelligence (IJCAI).
- [13] D. Dubois and H. Prade. (2001).« La problématique scientifique du traitement de l'information ». In : Information-Interaction-Intelligence 1.2 . pages 79–98.

- [14] D. Dubois, H. Prade, H. Farreny, R. Martin-Clouaire and C. Testemale.(1988). *Théorie des possibilités : applications à la représentation des connaissances en informatique*. Tome 1. Masson Paris.
- [15] T. Lukasiewicz and U. Straccia. (2008). « Managing uncertainty and vagueness in description logics for the semantic web ». In : *Web Semantics : Science, Services and Agents on the World Wide Web 6.4* , pages 291–308.
- [16] D. Dubois and H. Prade.(2009). « Formal representations of uncertainty ». In : *Decision-Making Process : Concepts and Methods* , pages 85–156.
- [17] K. J. Laskey, and al... (2016). *W3C Uncertainty Reasoning for the World Wide Web Incubator Group* [en ligne], Adressee URL: <https://www.w3.org/2005/Incubator/urw3/group/draftReport.html>
- [18] P. Judea. (1988). *Probabilistic reasoning in intelligent systems : networks of plausible inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA,.
- [19] F. Nir., G. Dan., and G. Moises. (1997). Bayesian network classifiers. *Machine learning*, 29(2-3) :131–163.
- [20] D. Grossman and P. Domingos. (2004). Learning bayesian network classifiers by maximizing conditional likelihood. In *Proceedings of the twenty-first international conference on Machine learning*, page 46. ACM.
- [21] F. Nir. (1998). The bayesian structural em algorithm. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 129–138. Morgan Kaufmann Publishers Inc.,
- [22] R. Fung., D. B. Favero. (1994). « Backward Simulation in Bayesian Networks ». Dans *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 227–234, San Francisco, CA, USA, july.
- [23] A. Goldberg and D. Robson. (1983). *Smalltalk-80: The Language and its Implementation*. Addison-Wesley.
- [24] O. Bangso ., P.-H.Wuillemin. (2000). *Réseaux Bayésiens Object Oriented - Un cadre pour Spécifications détaillées des grands réseaux bayésiens avec structures répétitives*. Rapport. Département de l'informatique. Aalborg University.
- [25] O. Bangso., M. Flores. (2001). l'apprentissage des paramètres dans les réseaux bayésiens orientés objet. *Annales de mathématiques et intelligence artificielle*. (32) 221–243. Kluwer Academic Publishers. Pays-Bas.

- [26] Y. Xiang, D. Poole and M.P. Beddoes. (1 993). Multipliez les réseaux bayésiens sectionnés et les forêts de jonction pour les grands systèmes basés sur la connaissance. *Intelligence computationnelle*, 9 (2): 1 7 1 -220.
- [27] Y. XIANG. (1998).Verification of DAG Structures in Cooperative Belief Networks Based Multi-Agent Systems. *Networks*, n.31, p.183-191.
- [28] Dr.M. Annemie. (2014) .Cancer du foie: un guide pour les patients – Basé sur les recommandations de l’ESMO. *ESMO/FAC Patient Guide Séries*. v.1
- [29] Y. Yang., J. Calmet (2005). Ontobayes: An ontology-driven uncertainty model. In: (IAWTIC’05). *IEEE Computer Society (2005)* 457-464.
- [30] Z. Ding., Y. Peng. (2004). A probabilistic extension to ontology language OWL, in: *Proceedings HICSS*.
- [31] P.C.G. da Costa., K.B. Laskey and K.J. Laskey. (2005). PR-OWL: A Bayesian ontology language for the Semantic Web, in: *Proceedings URSW*, pp. 23–33.
- [32] B.I. Mouna., L. Philippe and B.A. Nahla. (2011).Ontology-based generation of object oriented bayesian networks. In Ann Nicholson, editor, *Proceedings of the Eighth UAI Bayesian Modeling Applications Workshop (UAI-AW 2011)*, volume 818, pages 9–17, Barcelona, Spain.
- [33] K. Parminder., K. Aditya. (2015). Diagnosis of Liver Cancer Ontology using SPARQL. *International Journal of Applied Engineering Research*. ISSN 0973-4562 Vol. 10 No.69.

