

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 :

ELBESSEGHI Abdessalem

OUTTAS Smail

En vue d'obtenir le diplôme de master

Domaine : mathématique et informatique

Filière : Informatique

Spécialité : Informatique

Option : Ingénierie de logiciel

***Thème : Utilisation des ontologies probabilistes pour l'aide au diagnostic
médical.***

Soutenu le : Septembre 2019

M : HAMMOUDA. Président

M : CHIKHI. Examineur

M : FAREH M. Promotrice

Promotion

2018/2019

Remerciement

Nous tenons tout d'abord à remercier Dieu le tout puissant et miséricordieux, qui nous a donné la force et la patience d'accomplir ce modeste travail.

En second lieu, Nous tenons à remercier notre promotrice **Mme : FAREH M.** enseignante à l'université USDB, pour son aide et de nous avoir soutenues durant toute la période du travail.

Nous remercions le doctorant **Mr RIALI I.** pour nous avoir aidés et suivis durant la réalisation de ce projet.

Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre travail en acceptant d'examiner et de l'enrichir par leurs propositions.

Nous tenons à remercier également **Dr ELBESSEGHI H.** Médecin résidente au service de médecine interne du CHU de Blida.

Enfin, nous adressons nos plus sincères remerciements à nos familles : Nos parents, nos sœurs, nos frères, et tous nos proches et amis, qui nous ont accompagnés, aidé, soutenu et encouragé tout le long de la réalisation de ce mémoire.

ElBesseghi Abdessalem

Outtas Smail

Résumé

Le domaine médical est basé sur les diagnostics où les connaissances sont incertaines. Malgré le succès des ontologies classiques et leur utilisation pour modéliser et faire des raisonnements avec les connaissances dans plusieurs domaines, elles ne fournissent pas un support adéquat pour faire face à la connaissance incertaine. Les ontologies probabilistes jouent un rôle important dans la représentation de l'incertitude.

Notre objectif est de concevoir un système d'aide au diagnostic médical en utilisant les ontologies probabilistes, nous suivons dans ce mémoire la méthodologie UMP-ST basé sur les réseaux bayésiens multi entité, pour modéliser l'ontologie probabiliste. Nous commençons par la construction de la structure qui contient l'ensemble des nœuds du réseau bayésien, puis nous réalisons l'apprentissage de paramètres, à partir d'un ensemble de données. Une fois l'ontologie probabiliste est construite, nous l'utilisons pour la prédiction du diagnostic. Nous appliquons ce processus sur le cas d'étude du diagnostic de l'insuffisance rénale chronique.

Mots clé : ontologie, ontologie probabiliste, réseaux bayésiens, réseaux bayésiens multi-entité, incertitude.

Abstract

The medical field is based on diagnoses where knowledge is uncertain. Despite the success of classical ontologies and their use for modelling and reasoning with knowledge in many fields, they do not provide adequate support for dealing with uncertain knowledge. Probabilistic ontologies play an important role in the representation of uncertainty.

Our goal is to design a medical diagnostic support system using probabilistic ontologies, we follow in this thesis the UMP-ST methodology based on Bayesian multi entity networks, to model the probabilistic ontology. We start with the construction of the structure that contains all the nodes of the Bayesian network, and then we realize the learning of parameters, from a set of data. Once the probabilistic ontology is constructed, we use it for the prediction of the diagnosis. We apply this process to the case study of the diagnosis of chronic kidney disease

Key words: ontology, probabilistic ontology, Bayesian networks, Bayesian multi-entity networks, uncertainty.

ملخص

يعتمد المجال الطبي على التشخيصات حيث تكون المعرفة غير مؤكدة. على الرغم من نجاح الأنطولوجيا الكلاسيكية واستخدامها للنمذجة والتفكير مع المعرفة في العديد من المجالات ، إلا أنها لا توفر الدعم الكافي للتعامل مع المعرفة غير المؤكدة. تلعب الأنطولوجيا الاحتمالية دورًا مهمًا في تمثيل حالة عدم اليقين. هدفنا هو تصميم نظام دعم التشخيص الطبي باستخدام الأنطولوجيا الاحتمالية ، نتبع في هذا البحث منهجية UMPST القائمة على شبكات بايز متعددة الكيانات ، لنمذجة علم الوجود الاحتمالي. نبدأ ببناء الهيكل الذي يحتوي على جميع عقد شبكة بايزي ، ثم ندرك تعلم المعلمات ، من مجموعة من البيانات. بمجرد بناء علم الوجود الاحتمالي ، نستخدمه للتنبؤ بالتشخيص. نحن نطبق هذه العملية على دراسة الحالة لتشخيص الفشل الكلوي المزمن. الكلمات المفتاحية: الأنطولوجيا ، الأنطولوجيا الاحتمالية ، الشبكات البايزية ، الشبكات متعددة الكيانات البايزية ، عدم اليقين

Table des matières

Résumé	3
Contexte	10
Problématique.....	10
Objectif.....	11
Chapitre 1 : Les ontologies classiques et les ontologies probabilistes.....	13
1. Introduction.....	14
2. Ontologie classique.....	14
2.1. Définition.....	14
2.2. Les composants d'une ontologie	15
2.3. Rôles des ontologies	15
2.4. Les langages des ontologies	16
2.5. Motivations de création des ontologies	18
3. L'incertitude.....	18
3.1. Définition.....	18
3.2. Gestion de l'incertitude.....	18
3.3. L'incertitude dans les ontologies	20
4. Ontologie Probabiliste	20
4.1. Définition.....	20
4.2. Les langages des ontologies probabilistes	20
4.3. Matrice comparative	22
4.4. La méthodologie UMP-ST	23
4.5. Objectifs de l'ontologie probabiliste :	24
5. Différence entre ontologie classique et ontologie probabiliste.....	24
6. Conclusion	25

Chapitre 2 : Les réseaux bayésiens	26
1. Introduction.....	27
2. Définition des réseaux bayésiens	27
3. Domaine d'application.....	29
4. Les réseaux bayésiens standards	30
4.1. Inférence dans les réseaux bayésiens.....	30
4.1.1. Inférence exacte.....	30
4.1.2. Inférence approchée	31
4.2. Apprentissage dans les réseaux bayésiens	31
4.2.1. Apprentissage de structure	31
4.2.2. Apprentissage des paramètres	32
4.3. La mise en œuvre du réseau Bayésien.....	34
4.4. Les limites des RB :.....	34
5. Les réseaux bayésiens multi-entités (MEBN).....	35
5.1. Fragments MEBN (MFrag)	35
5.2. Théories MEBN (MTheory).....	36
5.3. Inférence dans le réseau bayésien multi-entité(MEBN).....	36
5.4. Applications du MEBN	37
6. Conclusion	37
Chapitre 3 : Analyse et conception	38
1. Introduction.....	39
2. Motivation de l'étude de cas	39
3. Le Dataset	39
4. Schéma global.....	41
5. Création de l'ontologie Probabiliste	43
5.1. Les exigences.....	43
5.1.1. Les besoins	44

5.2. Analyse et conception.....	44
5.2.1. Les entités.....	45
5.2.2. Les règles.....	47
5.2.3. Les groupes	50
5.3. Implémentation.....	51
5.3.1. Mapping	51
5.3.2. Probabilités.....	56
5.3.3. Test :.....	58
6. Résultat final de l'ontologie probabiliste :.....	63
7. Conclusion :	63
Chapitre 4 : Implementation et test	64
1. Introduction :.....	65
2. Les différents outils de développement utilisé pour notre système :.....	65
3. La mise en œuvre du système :	67
4. La base de connaissance :	68
5. Les interfaces :.....	68
6. La validation croisée	72
7. Conclusion :	75
Conclusion générale	76
Références bibliographiques	77

Liste des figures

Figure 1. 1 : Aperçu des concepts de la théorie PROWL MEBN [Intan et al., 2015]	21
Figure 1. 2 : Cycle de modélisation de l'ontologie probabiliste (UMP-ST) [Carvalho et al., 2016].....	24
Figure 2. 1 : Exemple de la représentation par réseaux bayésien.....	27
Figure 2. 2 : Graphe bayésien.....	29
Figure 2. 3 L'algorithme EM pour le maximum de vraisemblance	33

Figure 2. 4 : Etapes de construction d'un réseau bayésien[Naïm et al., 2011].....	34
Figure 3. 1 : Schéma global.....	42
Figure 3. 3 : La structure du réseau bayésien.....	49
Figure 3. 4 : réseau bayésien.	50
Figure 3. 5 : Diagramme de classe.	51
Figure 3. 6 : MTheory Tree.	52
Figure 3. 7 : Variable ordinaire « P ».	52
Figure 3. 8 : Noeud résident "blood_pressur".	53
Figure 3. 9 Noeud de contexte "IsA()".	53
Figure 3. 10 noeud d'entre "age".	53
Figure 3. 11 : MFrag "Facteur_De_Risque".	54
Figure 3. 12 : MFrag "SymptomesMFrag".	54
Figure 3. 13: MFrag "PyelonephritisMFrag".	55
Figure 3. 14 : MFrag "DiabetMFrag".	55
Figure 3. 15 : MFrag "HypertensionMFrag".	55
Figure 3. 16 : MFrag "AnemieMFrag".	56
Figure 3. 17 : MFrag "DiagnosticMFrag".	56
Figure 3. 18 : LPD du MFrag isTouchedBy.....	57
Figure 3. 19: Résultat SSBN pour le noeud IsTouchedBy(Patient P).....	58
Figure 3. 20 : Instance de la classe Patient.....	60
Figure 3. 21 : SSBN test1.....	61
Figure 3. 22 : SSBN Test2.	62
Figure 4. 1 : Fichier PLM.....	68
Figure 4. 2 Interface d'accueil.	69
Figure 4. 3 Interface « Add Patient »	69
Figure 4. 4 Interface « Consult Patient ».....	70
Figure 4. 5 Interface « Delete Patient ».....	70
Figure 4. 6 Interface « Prediction ».....	71
Figure 4. 7 Message « WARNING »	71
Figure 4. 8 Interface « Prediction ».....	72
Figure 4. 9 Calculer la précision du système.....	73
Figure 4. 10 : Exemple d'un résultat de validation (K=2).....	74

Liste des tableaux

Tableau 1. 1 : La matrice comparative des approches [Intan et al., 2015].....	22
Tableau 2. 1 : La modélisation de la table de probabilité.	28
Tableau 2. 2 : Tableau de probabilité conditionnelle.	29
Tableau 3. 1 : Description du dataset [Soundarapadian et al., 2015].....	40
Tableau 3. 2 : Description du schéma global du système.	43
Tableau 3. 3 : Les attributs de l'entité Patients.	45
Tableau 3. 4 : Les attributs de l'entité Anemie.	45
Tableau 3. 5 : les attributs de l'entité Symptomes.	46
Tableau 3. 6 : les attributs de l'entité Facteur_De_Risque.....	46
Tableau 3. 7 : les attributs de l'entité Hypertension.....	46
Tableau 3. 8 : les attributs de l'entité Diabete.....	46
Tableau 3. 9 : les attributs de l'entité Pyélonéphrite.	47
Tableau 3. 10 : les relations entre les entités.....	47
Tableau 3. 11 : Catégorisation d'attributs.	48
Tableau 4. 1 Précision du système	74

Introduction générale

Contexte

Dans le domaine médical, les médecins sont régulièrement amenés à prendre plusieurs décisions, certaines de ces décisions sont prises facilement, lorsque le diagnostic est facile, le traitement choisi est efficace et les risques sont nuls, alors que dans d'autres cas, la bonne décision à prendre n'est pas évidente car généralement le domaine de la santé est rempli d'incertitude.

Plusieurs approches existent dans la littérature médicale pour aider le médecin à prendre des meilleures décisions, parmi ces méthodes existe les ontologies probabilistes qui sont utilisées afin de décrire en détail les connaissances sur un domaine et l'incertitude associée à ces connaissances d'une manière structurée et partageable, idéalement dans un format qui peut être lu et traité par un ordinateur. Ils élargissent également les possibilités d'ontologies standards en introduisant l'exigence d'une représentation adéquate des régularités statistiques et les preuves incertaines sur les entités dans un domaine d'application.

A ce jour, tous les langages développés pour la représentation des ontologies probabilistes se basent sur les réseaux bayésiens. Les réseaux bayésiens sont des techniques très performantes en matière d'aide à la décision. Ils sont issus d'un mariage entre la théorie bayésienne des probabilités conditionnelles et la théorie des graphes. Ces techniques ont prouvé leur efficacité dans plusieurs domaines, tels que le domaine d'aide à la décision et le domaine de modélisation de systèmes complexes.

Problématique

Dans le domaine médical les données ou bien les connaissances d'une maladie sont incertaines et parfois complexes, c'est pour cela que les médecins sont régulièrement amenés à prendre plusieurs décisions dont certaines sont faciles et d'autres qui sont non évidentes.

L'incertitude dans le diagnostic est fréquemment rencontrée dans la pratique médicale et provoque du stress chez les patients ainsi que les médecins. En effet, les facteurs contribuant à l'incertitude comprennent la variabilité biologique des patients, les préjugés des patients et des médecins, l'incertitude donne vie à de nombreux conflits sur l'interprétation des tests, ce qui conduit les médecins à faire parfois des erreurs cliniques ou un mauvais diagnostic.

Cette incertitude peut se manifester par exemple lors de la prédiction, qu'un patient est atteint par une maladie. Dans notre cas d'étude la maladie traitée est l'insuffisance rénale chronique, cela nous conduit à poser les questions suivantes :

- Comment modéliser ces connaissances incertaines dans le domaine médical ?
- Comment construire l'ontologie probabiliste, quels sont ses composants et quelle est sa structure ?
- La mise en œuvre du système commence par l'identification d'un ensemble des signes pertinents pour établir un diagnostic, mais comment établir ce système et quels sont les outils nécessaires ?

Objectif

Notre objectif est de concevoir et de réaliser un système d'aide au diagnostic médical de la maladie « Insuffisance rénale chronique », en se basant sur les ontologies probabilistes. Cette ontologie est construite à l'aide de la méthodologie UMP-ST (Uncertainty Modeling Process for Semantic Technology).

Afin de prédire le diagnostic médical à partir des connaissances incertaines, l'utilisation des réseaux bayésiens est nécessaire, pour notre étude de cas le MEBN (Multi Entity Bayesian Networks) est le réseau bayésien choisi, ce dernier permet de créer un outil pour implémenter une ontologie probabiliste « PR-OWL » (PProbabilistic Ontology Web Language) à l'aide d'un standard de représentation d'ontologie OWL (Ontology Web Language).

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

La première partie :

Contient deux chapitres, elle présente le contexte du travail, elle a pour but de présenter les ontologies classiques, les ontologies probabilistes, les réseaux bayésiens classiques, les réseaux bayésiens multi-entités.

Chapitre 1 : Les Ontologies Classique et Les Ontologies Probabilistes : Ce chapitre contient une présentation de deux sortes d'ontologie, classique et probabiliste, ses différents composants, leurs rôles ainsi que la définition de la seule méthodologie qui permet la création de l'ontologie probabiliste (UMP-ST).

Nous parlerons aussi de quelques langages de représentation de ces deux types d'ontologies, nous avons aussi fait une comparaison entre les langages des ontologies probabilistes, tout en introduisant la notion d'incertitude, leur modèle et classification.

Chapitre 2 : Les Réseaux Bayésiens: Ce chapitre contient les différentes définitions des réseaux bayésiens classiques et les réseaux bayésiens multi-entités.

La deuxième partie :

Après les généralités présentées dans la première partie. Cette partie est consacrée pour la conception et le développement du système.

Cette partie met l'accent sur la conception de notre système et de la construction d'une ontologie PR-OWL.

Chapitre 3 : Analyse et Conception : Ce chapitre contient la motivation de notre cas d'étude, le processus de construction d'une ontologie probabiliste sous le langage PR-OWL en suivant les différentes étapes de la méthodologie UMP-ST.

Chapitre 4 : Implémentation et Test : Ce chapitre est le résultat de la conception pour réaliser le système de la modélisation de l'ontologie probabiliste PR-OWL, à la fin de ce chapitre nous effectuons un test de validation sur notre application.

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

CHAPITRE 1

LES ONTOLOGIES CLASSIQUES ET LES ONTOLOGIES PROBABILISTES

1. Introduction

Les ontologies classiques permettent de représenter les connaissances sur un domaine particulier, Elles sont des représentations qui regroupent un ensemble de concepts et de relations.

Les ontologies classiques ont un principal défaut qui est l'incapacité de représenter et de raisonner sur l'incertitude, pour répondre à ce problème les ontologies probabilistes ont vu le jour.

Dans ce chapitre nous allons définir deux type d'ontologie, classique et probabiliste, ses différents composants, leurs rôles ainsi que la définition de la seule méthodologie qui permet la création de l'ontologie probabiliste (UMP-ST).

Nous parlerons aussi de quelques langages de représentation de ces deux types d'ontologies, nous avons aussi fait une comparaison entre les langages des ontologies probabilistes, tout en introduisant la notion d'incertitude, leur modèle et classification.

2. Ontologie classique

Une ontologie est l'ensemble structuré des termes et concepts représentant le sens d'un champ d'informations.

2.1. Définition

Dans le cadre de l'intelligence artificielle, Neeches et ses collègues [Neeches et al., 1991] furent les premiers à proposer une définition à savoir : « une ontologie définit les termes et les relations de base du vocabulaire d'un domaine ainsi que les règles qui indiquent comment combiner les termes et les relations de façon à pouvoir étendre le vocabulaire ».

Swartout et ses collègues [Swartout et al., 1997] la définissent comme suit : « une ontologie est un ensemble de termes structurés de façon hiérarchique, conçue afin de décrire un domaine et qui peut servir de charpente à une base de connaissances ».

La même notion est également développée par Gomez [Gomez et al., 2004] comme : « une ontologie fournit les moyens de décrire de façon explicite la conceptualisation des connaissances représentées dans une base de connaissances ».

2.2. Les composants d'une ontologie

Les connaissances dans les ontologies sont principalement formalisées en utilisant cinq types de composants :

- **Les concepts (ou classes) :** Un concept est un constituant de la pensée (un principe, une idée, une notion abstraite) sémantiquement évaluable et communicable. L'ensemble des propriétés d'un concept constitue sa compréhension ou son intention et l'ensemble des êtres qu'il englobe son extension.
- **Les relations (ou propriétés) :** Elles représentent des interactions entre les concepts, elles permettent de construire des représentations complexes de la connaissance du domaine, elles établissent des liens sémantiques binaires, organisables hiérarchiquement.
- **Les fonctions :** Elles présentent des cas particuliers de relations dans lesquelles le nième élément de la relation est unique pour les n-1 éléments précédents. Formellement, les fonctions sont définies telles que : $F : c_1 * c_2 * ... * c_{n-1} * c_n$.
- **Les axiomes (ou règles) :** Les axiomes sont des expressions qui sont toujours vraies. Ils ont pour but de définir dans un langage logique la description des concepts et des relations permettant de représenter leur sémantique. Ils représentent les intentions des concepts et des relations du domaine et, de manière générale, les connaissances n'ayant pas un caractère strictement terminologique.
- **Les instances (ou individus) :** Elles constituent la définition extensionnelle de l'ontologie, elles sont utilisées pour représenter des éléments dans un domaine donné [Staab et al., 2000].

2.3. Rôles des ontologies

Plusieurs chercheurs se sont intéressés à la finalité des exploitations des ontologies. Nous précisons dans ce qui suit les rôles et l'intérêt des ontologies au sein des systèmes à base de connaissances (SBC) et du Web Sémantique :

- **Les connaissances du domaine d'un SBC :** En particulier, elles servent de squelette à la représentation des connaissances du domaine dans la mesure où elles décrivent les objets, leurs propriétés et la façon dont ils peuvent se combiner pour constituer des connaissances complètes du domaine.

- **La communication :** Les ontologies peuvent intervenir dans la communication entre personnes, organisations et logiciels [Uschold et al., 1996]. En effet, les ontologies servent par exemple, à créer au sein d'un groupe ou d'une organisation un « vocabulaire conceptuel commun ». Dans ce cas, on est plutôt dans le cadre d'une ontologie informelle.
- **L'interopérabilité :** L'interopérabilité est une spécialisation de la communication qui permet de répertorier les concepts que des applications peuvent s'échanger même si elles sont distantes et développées sur des bases différentes.

2.4. Les langages des ontologies

- **KIF**

Est un langage basé sur les prédicats du premier ordre avec des extensions pour représenter des définitions et des méta-connaissances. Tant que la logique du premier ordre est un langage de bas niveau pour l'expression d'ontologies, l'outil Ontolingua [Farquhar et al., 1997] Permet aux utilisateurs de construire des ontologies KIF à un niveau plus élevé de la description par l'importation des définitions des ontologies prédéfinies.

- **KL-ONE**

Est un langage basé sur la logique de description [Brachman et al., 1985]. Il est une formalisation de représentation de la connaissance à base de cadres (« frame-based »). Ce système maintient la définition des concepts par un simple nommage, et l'indication de la correspondance des concepts dans une hiérarchie de généralisation/spécialisation. De nouveaux termes peuvent être définis par des opérations de conjonction des concepts. De nouveaux rôles peuvent être introduits pour représenter les relations qui peuvent exister entre des individus dans le domaine modélisé. Les définitions des concepts peuvent inclure des restrictions sur les valeurs possibles, sur les nombres de valeurs, ou sur le type de valeurs qu'un rôle peut avoir pour un concept.

- **RDF et RDF Schéma**

« Resource Description Framework » est un formalisme graphique pour représenter des métadonnées. Il est basé sur la notion de triplet (sujet, prédicat, objet). RDF utilise la syntaxe XML, mais il ne donne aucune signification spécifique pour le vocabulaire comme sous classe de,

ou le type. Les primitives de modélisation offertes par RDF sont très basiques. RDF Schéma (RDFS) [Brickley et al., 2004] est un langage qui étend RDF avec un vocabulaire de termes et les relations entre ces termes. RDFS est reconnu comme un langage d'ontologie qui définit :

Des classes et des propriétés. Les sous-classes, les superclasses, les sous-propriétés, et les super-propriétés. Le domaine de définition et le domaine image des propriétés.

- **DAML + OIL**

Beaucoup de travaux ont été faits dans le domaine de la représentation des connaissances parmi lesquels on peut citer les plus importants : SHOE, OntoBroker [Berardi et al., 2001], OIL [Hartel et al., 2005], ET encore DAML + OIL [Pham et al., 2008] qui a remplacé DAML – ONT. DAML + OIL est un langage construit sur des normes précédentes du W3C telles que RDF et RDFS, et étend ces langages avec des primitives de modélisation plus riches.

DAML+OIL a été conçu à partir du langage d'ontologie DAML-ONT [Naïm et al., 2007] en vue de combiner plusieurs composants du langage OIL [Hartel et al., 2005]. OIL « Ontology Inference Language » est une représentation basée sur le Web, et une couche d'inférence pour des ontologies.

- **OWL**

Développé par le groupe de travail sur le Web Sémantique du W3C, OWL peut être utilisé pour représenter explicitement les sens des termes des vocabulaires et les relations entre ces termes. OWL vise également à rendre les ressources sur le Web aisément accessibles aux processus automatisés [McGuinness et al., 2004], d'une part en les structurant d'une façon compréhensible et standardisée, et d'autre part en leur ajoutant des méta-informations. Pour cela, OWL a des moyens plus puissants pour exprimer la signification et la sémantique que XML, RDF, et RDFS [Brickley et al., 2004]. De plus, OWL tient compte de l'aspect diffus des sources de connaissances et permet à l'information d'être recueillie à partir de sources distribuées, notamment en permettant la mise en relation des ontologies et l'importation des informations provenant explicitement d'autres ontologies [McGuinness et al., 2004].

2.5. Motivations de création des ontologies

- Un besoin générique : faire tomber les barrières créées par des vocabulaires disparates.
- Partager la compréhension commune de la structure de l'information entre les personnes ou les fabricants de logiciels.
- Permettre la réutilisation du savoir sur un domaine.
- Un besoin existe de partager la signification de termes dans un domaine donné.
- Une aide à la communication entre agents humains et aussi entre organisations.

3. L'incertitude

La fiabilité de l'information est très souvent remise en cause. En effet, l'incertitude, l'imprécision et l'incomplétude sont des problèmes récurrents dans le traitement de l'information. L'incertitude est la forme d'imperfection la plus étudiée, cela est notamment dû au fait qu'elle peut être modélisée formellement.

3.1. Définition

Le terme "incertitude" désigne une information qui n'est pas établie avec certitude, qui peut ou non se produire. L'incertitude est un état où nous ne savons pas quel choix prendre.

Plusieurs travaux ont été consacrés à l'incertitude, considèrent plusieurs représentations Formelles se basant sur des logiques probabilistes, possibilistes et floues.

Les auteurs de [Didier et Henri, 2009] ont considéré l'aspect caractériel de l'information, ils identifient deux types d'incertitude :

- ✓ **Incertitude stochastique** : caractérise l'aspect aléatoire de l'information, ainsi que sa variabilité.
- ✓ **Incertitude épistémique** : liée à l'incomplétude de l'information et au manque de connaissance de l'auteur. [Kerdjoudj, 2015].

3.2. Gestion de l'incertitude

Avec l'émergence d'Internet et du Web en particulier, 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 [Didier et Henri, 2001] 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 une ambiguïté ou une incohérence.

Toutes ces imperfections liées à la langue naturelle ou encore à l'acquisition de l'information remettent en question l'existence même des éventuelles connaissances à extraire. Chacune de ces imperfections peut être traitée de manière individuelle.

La gestion de l'incertitude présente différentes perspectives d'utilisation telles que l'évolution des connaissances, l'aide à la décision, l'analyse de controverses, la vérification et les analyses de faits ou encore la cotation des sources [Kerdjoudj, 2015].

3.3. L'incertitude dans les ontologies

L'approche probabiliste est un domaine qui essaie de trouver des mécanismes efficaces pour modéliser le raisonnement, tenant compte de l'incertitude de certaines connaissances. Dans ce domaine, les modèles graphiques probabilistes permettent de fournir un outil compact et expressif pour modéliser l'incertitude et la complexité. Ceux-ci joignent dans la même représentation la théorie des probabilités et la théorie des graphes [Nizza et al., 2013].

4. Ontologie Probabiliste

Les ontologies probabilistes sont des ontologies qui traite l'incertitudes.

4.1. Définition

Une ontologie probabiliste est une représentation de la connaissance formelle explicite qui exprime des connaissances sur un domaine d'application. Ceci comprend :

- Types d'entités qui existent dans le domaine.
- Propriétés de ces entités.
- Les relations entre les entités.
- Les processus et les événements qui se passent avec ces entités.
- Régularités statistiques qui caractérisent le domaine.
- Connaissances peu concluantes, ambiguës, incomplètes et peu fiables liés aux entités du domaine.
- Incertitude sur toutes les formes de connaissance citées ci-dessus

D'où, le terme entité se réfère à tout concept (réel ou fictif, concret ou abstrait) qui peut être décrit et raisonner selon le domaine d'application [Costa et al., 2006].

4.2. Les langages des ontologies probabilistes

Ils existent plusieurs langages des ontologies probabilistes, nous citons les plus connu.

- **PR-OWL**

PR-OWL est une extension qui 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 : Netica, Hugin, Quiddity*Suite, JavaBayes, etc.) [Paulo et al., 2016].

La figure-1.1 décrit un aperçu de concepts de la théorie PR-OWL MEBN [Intan et al., 2015].

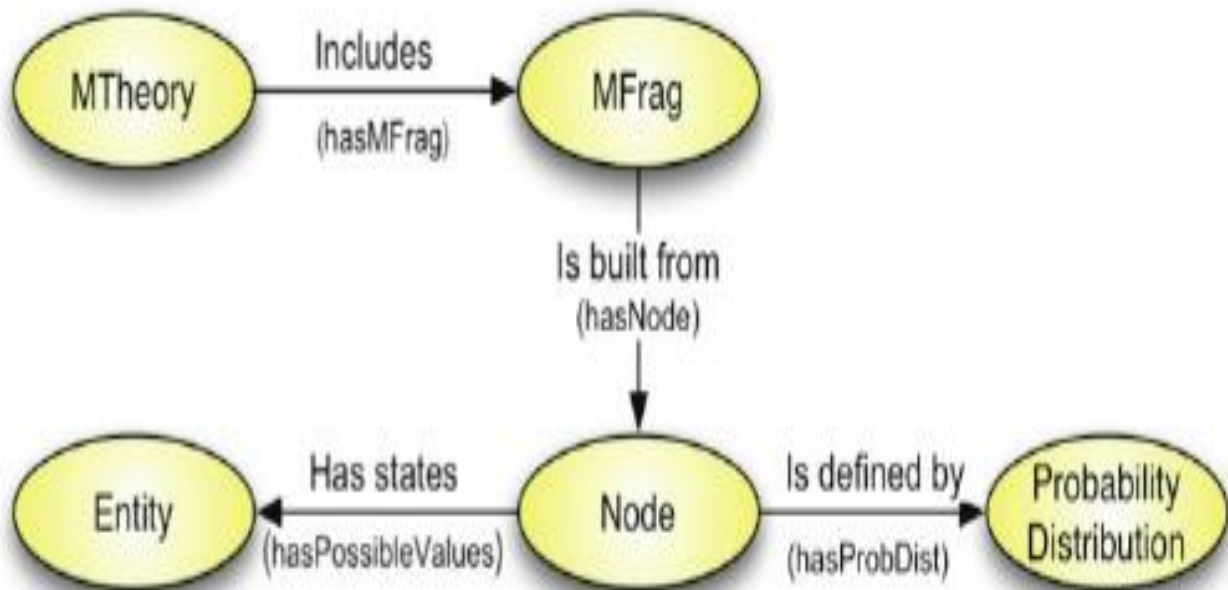


Figure 1. 1 : Aperçu des concepts de la théorie PROWL MEBN [Intan et al., 2015]

- **MEBN**

MEBN (Multy Entity Bayesian Network) est un système logique qui intègre la logique du premier ordre (First Order Logic (FOL)) avec la théorie des probabilités bayésiennes, il étend les réseaux bayésiens ordinaires pour permettre la représentation des modèles graphiques avec des sous-structures répétées. La connaissance est codée comme une collection de fragments de réseaux bayésiens (MFrag) qui peuvent être instanciés et combinés pour former des réseaux bayésiens très complexes spécifiques à chaque situation. Une théorie MEBN (MTheory) représente implicitement une distribution de probabilité conjointe sur les chiffres éventuellement non bornées d'hypothèses et utilise l'apprentissage bayésien pour affiner une base de connaissances quand les observations s'accumulent. MEBN fournit une base logique pour la collection émergente des langages à base de probabilités très expressives [Paulo et al., 2016].

- **BayesOWL**

Est un Framework probabiliste qui augmente et complète OWL pour représenter et raisonner avec l'incertitude basée sur les réseaux bayésiens (RB). Le modèle BayesOWL de base comprend un ensemble de règles de traduction structurelle pour convertir une ontologie OWL en un graphe acyclique dirigé (DAG) d'un RB et un mécanisme qui utilise des informations probabilistes disponibles pour construire une table de probabilité conditionnelle (CPT) pour chaque nœud du DAG. Pour aider à comprendre l'approche [Intan et al., 2015].

- **Dempster-Shafer Theory**

La théorie est venue à l'attention des chercheurs d'intelligence artificielle (AI) au début des années 1980, quand ils essayaient d'adapter la théorie des probabilités aux systèmes experts. Les degrés de croyance Dempster-Shafer ressemblent aux facteurs de certitude, et cette ressemblance a suggéré qu'ils pourraient combiner la rigueur de la théorie des probabilités avec la flexibilité des systèmes basés sur des règles [Intan et al., 2015].

4.3. Matrice comparative

Un aperçu est donné de l'analyse de chaque approche sur les métriques, leurs natures, leurs caractéristiques et leurs performances. Le **Tableau 1.1** résume les résultats analytiques.

Approaches \ Metrics	Complexity	Accuracy	Ease of Implementation	Reasoning Mechanism
BayesOWL	SIMPLE/ MODERATE	MEDIUM	EASY	AVAILABLE
MEBN	COMPLEX	HIGH	MEDIUM/HARD	AVAILABLE
PR-OWL	COMPLEX	HIGH	MEDIUM/HARD	AVAILABLE
Dempster-Shafer	MODERATE	MEDIUM	MEDIUM	AVAILABLE

Tableau 1. 1 : La matrice comparative des approches [Intan et al., 2015]

Après la définition des langages nous avons pu choisir le langage correspondant à notre projet qui est le PR-OWL pour pouvoir gérer les connaissances incertaines en utilisant les ontologies probabilistes.

Dans le cas d'aide au diagnostic médical nous choisiront un langage qui soit précis, accessible et pertinent. Le choix a été basé sur ces critères :

- PR-OWL est basé sur le MEBN.
- MEBN est meilleur que les RB standards par rapport au niveau d'expressivité.
- Niveau de précision de MEBN et PROWL est plus fort par rapport aux autres.
- L'existence des outils open source proposés par PR-OWL.
- La pertinence du résultat dans PR-OWL.

4.4. La méthodologie UMP-ST

Développer des ontologies probabilistes peut être grandement facilité par l'utilisation d'un cadre de modélisation tel que l'UMP-ST (un processus de modélisation pour la construction d'une ontologie probabiliste).

L'UMP-ST est un processus itératif et incrémental, basé sur le processus unifié (UP), pour la conception d'une ontologie probabiliste. Alors que l'UP sert de point de départ, UMP-ST s'inspire aux processus d'ingénierie de l'ontologie.

Cette méthodologie est applicable à tout problème nécessitant le développement d'une ontologie probabiliste, dans laquelle PR-OWL a été choisi comme langage ontologique en raison de son traitement complet de l'incertitude [Carvalho et al., 2016].

Le diagramme dans la figure -1.2- montre une vision générale des différentes étapes du processus de la méthodologie

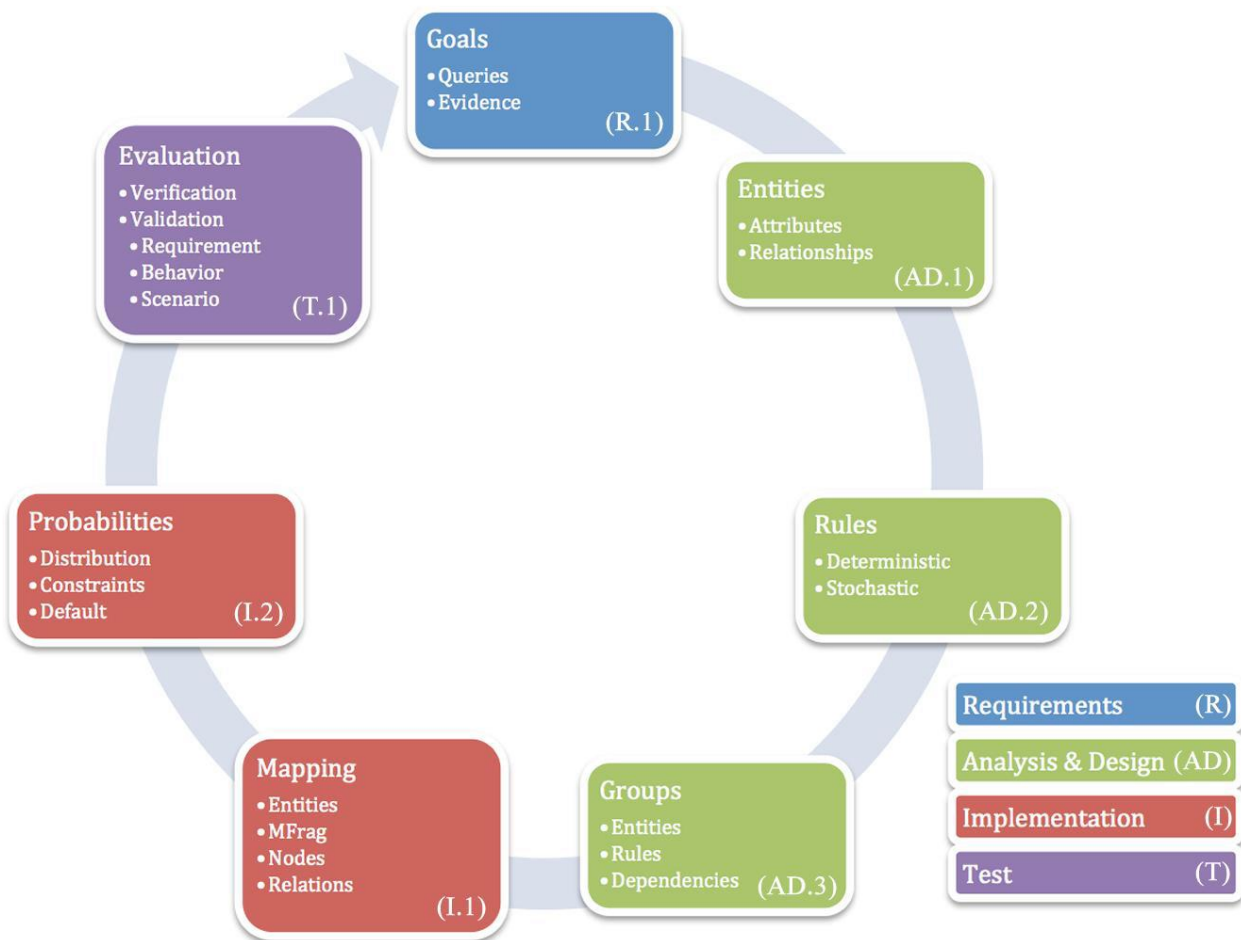


Figure 1. 2 : Cycle de modélisation de l'ontologie probabiliste (UMP-ST) [Carvalho et al., 2016]

4.5. Objectifs de l'ontologie probabiliste :

Les ontologies probabilistes sont utilisées dans le but de décrire en détail les connaissances sur un domaine et l'incertitude associée à ces connaissances d'une manière raisonnée, structurée et partageable, idéalement dans un format qui peut être lu et traité par un ordinateur. Ils élargissent également les possibilités d'ontologies standards en introduisant l'exigence d'une représentation adéquate des régularités statistiques et les preuves incertaines sur les entités dans un domaine d'application [Costa et al., 2006].

5. Différence entre ontologie classique et ontologie probabiliste

La principale différence entre les deux types d'ontologie est que l'ontologie classique n'a pas la capacité de représenter et de raisonner sur l'incertitude, pour cela l'utilisation des ontologies

probabilistes est nécessaire. Les modèles graphiques probabilistes permettent de fournir un outil compact et expressif pour modéliser l'incertitude et la complexité. Ceux-ci joignent dans la même représentation la théorie des probabilités et la théorie des graphes.

6. Conclusion

À travers ce premier chapitre nous avons présenté deux types d'ontologie classique et probabiliste avec leurs caractéristiques, leurs constructions, leurs composants et leurs langages puis, nous avons introduit la notion d'incertitude et l'incapacité de l'intégrer dans les ontologies classiques qui nous conduisent à la modéliser dans les ontologies probabilistes. Ce chapitre nous amène à choisir le langage qui vas au mieux avec nos besoins qui est le PROWL afin de représenter les connaissances incertaines dans le domaine médical.

CHAPITRE 2
LES RESEAUX BAYSIENS

1. Introduction

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. Dans ce chapitre nous allons définir les réseaux bayésiens classiques et les réseaux bayésiens multi-entités.

2. Définition des réseaux bayésiens

Les réseaux Bayésiens sont des modèles qui représentent des connaissances incertaines sur des phénomènes complexes. Un réseau Bayésien peut se décomposer en deux parties (voir Figure 2.1) :

- Une représentation graphique par un graphe orienté acyclique (Directed Acyclic Graph ou DAG). Les variables aléatoires sont représentées par des nœuds contenant leurs états, et les relations de dépendance entre ces variables par des arcs. Les nœuds sont reliés par la relation causale (relation de cause à effet),
- Une représentation probabiliste par un ensemble de tables de probabilités conditionnelles (CPT). A chaque nœud est associée une CPT de taille exponentielle au nombre de parents et leur nombre d'états.

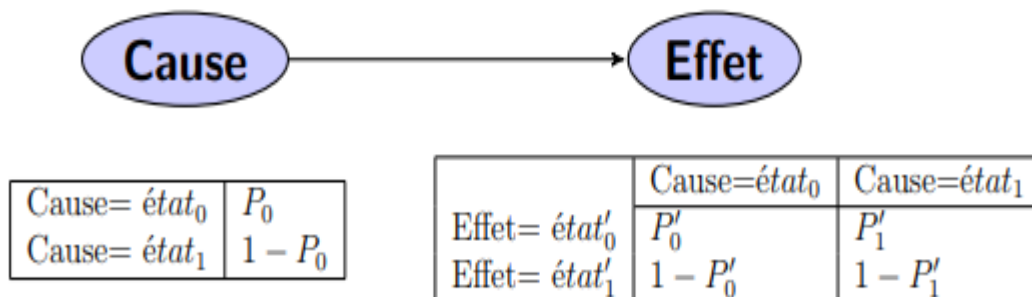


Figure 2. 1 : Exemple de la représentation par réseaux bayésien.

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 : $P\left(\frac{a}{b}\right) = p(b) \cdot p(b/a) / p(a)$. [Judea, 1988]

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

$$P(X_1, \dots, X_n) = \prod_i^n P(X_i | Pa(X_i)). [\text{Judea, 1988}]$$

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

- **Un exemple simple de réseau bayésien**

Ce matin-là, alors que le temps est clair et sec, M. Holmes sort de sa maison. Il s'aperçoit que la pelouse de son jardin est humide. Il se demande alors s'il a plu pendant la nuit, ou s'il a simplement oublié de débrancher son arroseur automatique. Il jette alors un coup d'œil à la pelouse de son voisin, M. Watson, et s'aperçoit qu'elle est humide. Il en déduit alors qu'il a probablement plu, et il décide de partir au travail sans vérifier [Naïm et al., 2007].

Dans ce texte, monsieur Holmes fait un raisonnement qui peut être décrit à l'aide d'un Réseau bayésien. L'herbe de mon jardin est mouillée (**J**) si et seulement si : il a plu cette nuit (**P**) ou j'ai oublié de débrancher mon arroseur (**A**).

Ce qui se traduit par la table de probabilité ci-dessous :

	A=vrai		A=faux	
	P=vrai	P=faux	P=vrai	P=faux
J=vrai	1	1	1	0
J=faux	0	0	0	1

$$P(J=vrai/A=vrai, P=vrai) = 1$$

Tableau 2. 1 : La modélisation de la table de probabilité.

L'herbe du jardin de mon voisin Watson est humide (**W**) si et seulement s'il a plu cette nuit. La table de probabilité associée :

	P=vrai	P=faux
W=vrai	1	0
W=faux	0	1

Tableau 2. 2 : Tableau de probabilité conditionnelle.

Cela conduit au graphe causal ci-dessous.

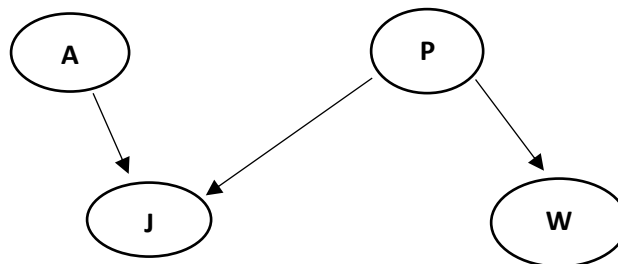


Figure 2. 2 : Graphe bayésien.

A : J'ai oublié de débrancher mon arroseur.

P : Il a plu pendant cette nuit.

J : L'herbe de mon jardin est humide.

W : L'herbe du jardin de mon voisin Watson est humide.

3. Domaine d'application

Un réseau bayésien sert à représenter la connaissance que l'on a d'un système (technique, informatique, Biologique, économique, ...) en vue de :

- Prévoir le comportement du système.
- Diagnostiquer (les causes d'un phénomène observé dans le système).
- Contrôler (le comportement du système).
- Simuler (le comportement du système).
- Analyser des données (relatives au système).
- Prendre des décisions (concernant le système).

- **Exemples dans différents domaines**

Les domaines d'utilisation principaux pour les RB sont le diagnostic (médical et industriel), analyse de risques, détection de spams, datamining, détection de fraudes, exploitation du retour d'expérience, modélisation et simulation de systèmes complexes, détection d'intrusions, TextMining, analyse de Biopuces, analyse de trajectoires de santé.

4. Les réseaux bayésiens standards

Un réseau bayésien est un modèle graphique probabiliste représentant un ensemble de variables aléatoires sous la forme d'un graphe orienté acyclique.

4.1. 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 :

4.1.1. 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 $E_1 \dots E_m$;
- **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 : $P(X|e)$ [Favero, 1994].

Parmi les algorithmes d'inférence exacte, on peut citer l'algorithme de propagation de messages, ainsi que les algorithmes de coupe ou de conditionnement qui exploitent la structure du réseau. Judea Pearl, le père de l'inférence dans les réseaux bayésiens, a développé un algorithme d'envoi de messages (message-passing) [Judea, 1988]. Le principe consiste, pour chaque nœud du réseau, à mettre à jour la distribution de probabilité de proche en proche avec ses voisins pour prendre en compte les observations sur un ou plusieurs nœuds. Cet algorithme s'applique aux réseaux bayésiens dont le graphe est un arbre ou un poly-arbre. Grâce à un algorithme coupe-cycle, il est possible d'étendre cette méthode à n'importe quelle structure de réseau bayésien.

4.1.2. Inférence approchée

Etant donné que l'inférence exacte est impraticable dans des grands réseaux multiplement connectés, il est essentiel d'envisager des méthodes d'inférence approchée. Cette section décrit d'algorithme d'échantillonnage aléatoire, également nommés 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.

4.2. Apprentissage dans les réseaux bayésiens

L'apprentissage d'un réseau bayésien à partir d'un jeu de données contenant un ensemble d'événements se fait en deux étapes :

- L'apprentissage de la structure, où le but est de trouver le meilleur graphe représentant la tâche à résoudre.
- L'apprentissage de paramètres : Une fois la structure connue, la seconde étape consiste à apprendre les bonnes distributions de probabilités conditionnelles des variables.

4.2.1. Apprentissage de 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 , Par exemple, $Score(G, D) = P(G/D) = P(D/G) \cdot P(G) \cdot P(D)$.

L'approche bayésienne de l'apprentissage de structure se résume à deux éléments essentiels : La fonction d'évaluation associée à une structure, et la procédure de recherche de la structure visant à maximiser la fonction d'évaluation.

4.2.2. Apprentissage des paramètres

L'apprentissage des paramètres pour une structure particulière consiste à estimer les distributions de probabilités à 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. Au contraire, si les données sont incomplètes, l'algorithme Expectation-Maximisation peut être utilisé. 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.

- **L'algorithme EM**

L'algorithme EM (Expectation-Maximisation) est un algorithme itératif dû à Dempster, Laird et Rubin (1977). Il s'agit d'une méthode d'estimation paramétrique s'inscrivant dans le cadre général du maximum de vraisemblance. Lorsque les seules données dont on dispose ne permettent pas l'estimation des paramètres, et/ou que l'expression de la vraisemblance est analytiquement impossible à maximiser, l'algorithme EM peut être une solution. De manière grossière et vague, il vise à fournir un estimateur lorsque cette impossibilité provient de la présence de données cachées ou manquantes ou plutôt, lorsque la connaissance de ces données rendrait possible l'estimation des paramètres. L'algorithme EM tire son nom du fait qu'à chaque itération il opère deux étapes distinctes

- La phase « Expectation », souvent désignée comme « l'étape E », procède comme son nom le laisse supposer à l'estimation des données inconnues, sachant les données observées et la valeur des paramètres déterminée à l'itération précédente.

- La phase « Maximisation », ou « étape M », procède donc à la maximisation de la vraisemblance, rendue désormais possible en utilisant l'estimation des données inconnues effectuée à l'étape précédente, et met à jour la valeur du ou des paramètre(s) pour la prochaine itération [Naïm, 1999].

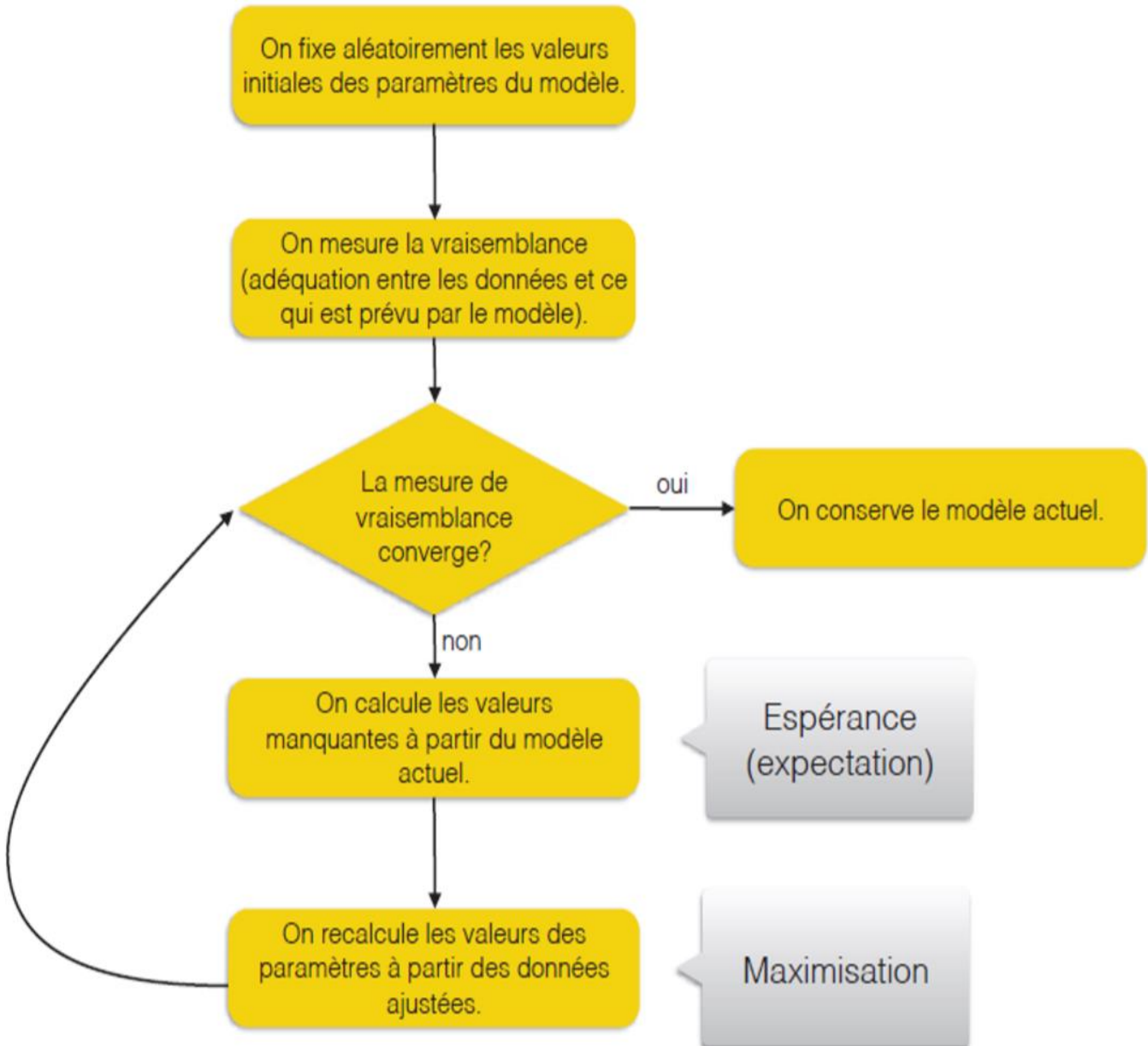


Figure 2. 3 L'algorithme EM pour le maximum de vraisemblance

4.3. La mise en œuvre du réseau Bayésien

La construction d'un réseau bayésien s'effectue en trois étapes essentielles, qui sont présentées sur la figure 2.3.

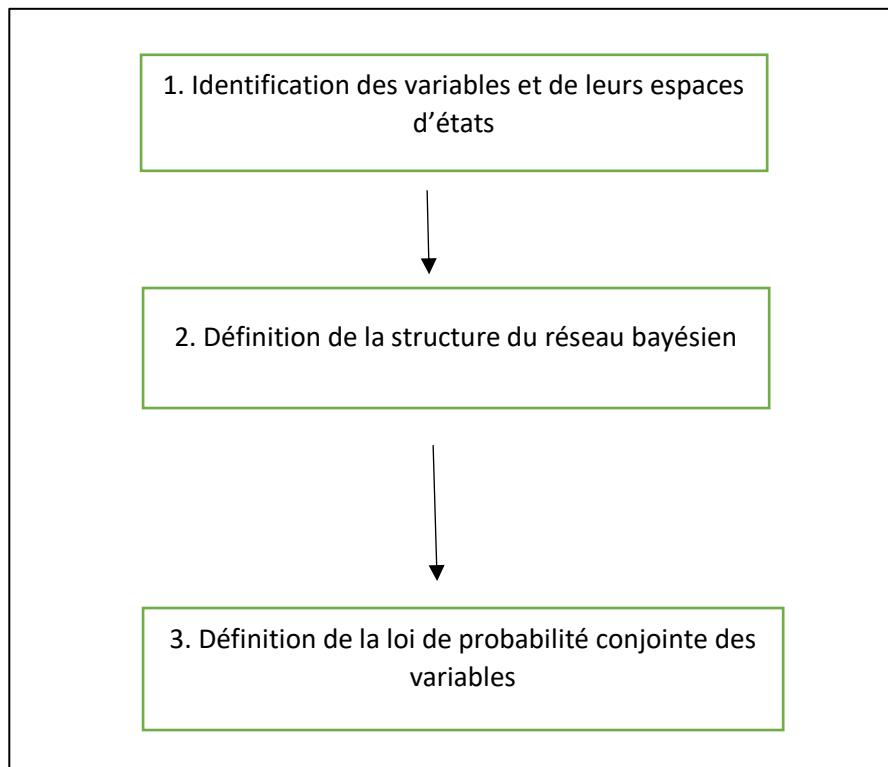


Figure 2. 4 : Etapes de construction d'un réseau bayésien[Naïm et al., 2011].

La première étape dite qualitative : consiste à la définition de l'ensemble des variables du système, avec précision de l'espace d'états de chaque variable.

La deuxième est l'étape probabiliste : elle introduit l'idée de distribution jointe définie sur Les variables à générer la base d'observations et choisir une structure de graphe qui sera compatible avec les variables.

La troisième est l'étape quantitative : elle consiste en l'évaluation numérique des distributions de probabilités conditionnelles.

4.4. Les limites des RB :

Depuis plusieurs décennies, les réseaux bayésien se sont avérés être l'outil essentiel entre les mains des chercheurs travaillant dans le domaine de l'intelligence artificielle. Pourtant, les réseaux bayésien ont certaines limites [Wang, 2004], les réseaux bayésien ont besoin d'extensions pour être plus expressives et fonctionnelles pour les besoins de raisonnement probabiliste de différents

domaines, l'utilisation des probabilités et leurs données par l'expert sont problématiques, problème des variables continues et la complexité des algorithmes.

5. Les réseaux bayésiens multi-entités (MEBN)

Les réseaux bayésiens multi-entités (MEBN) sont une théorie combinant l'expressivité des principes logiques de premier ordre et le raisonnement probabiliste des réseaux bayésiens. Le MEBN a été à ce jour la méthode de raisonnement logique probabiliste de premier ordre la plus expressive et a été utilisée méticuleusement pour surmonter plusieurs limitations des réseaux bayésiens.

Le réseau bayésien multi-entités (MEBN) emprunte le concept graphique de représentation de la distribution de probabilité conjointe pour un ensemble de variables aléatoires dans un domaine d'intérêt particulier, et vise à améliorer les réseaux bayésiens conventionnels en incorporant des relations sémantiques entre entités la logique classique de premier ordre en utilisant le pouvoir de gestion de l'incertitude de la théorie des probabilités [Mahoney et Laskey, 1998].

Les principaux éléments de base du MEBN sont les théories MEBN (MTheories), qui sont des outils puissants pour la modélisation des connaissances spécifiques aux domaines pour l'évaluation de la situation. Les théories MEBN sont composées de plusieurs fragments définis sur un ensemble de variables aléatoires sémantiquement liées (représentant une entité), appelées Fragments MEBN (MFrag). Ceci constitue la principale caractéristique de MEBN qui est sa modularité, c'est-à-dire que les MFrag peut être facilement (ajouté / retiré) du système modélisé, sans perdre la cohérence structurelle. De plus, les distributions de probabilités globales sont définies sur un ensemble de petits groupes d'hypothèses, satisfaisant les contraintes de consistance, avec des distributions de probabilité locales spécifiques.

Dans une perspective plus large, la distribution de probabilité conjointe sur les valeurs de vérité des ensembles d'entités sémantiquement connectées est la sortie d'un MTheory. Les sections suivantes fournissent une explication plus détaillée de MEBN.

5.1. Fragments MEBN (MFrag)

Un MFrag est défini par un tuple $F = (C, I, R, G, D)$ [Laskey, 2008], le MFrag propose des types de nœuds différents, à savoir les nœuds de contexte C , les nœuds d'entrée I , les nœuds résidents R , G représente un graphe et D un ensemble de distributions locales. Enrichis par la connaissance du domaine, les nœuds contextuels sont utilisés pour spécialiser la définition générale des phrases FOL dans un MFrag. D'un autre côté, les nœuds d'entrée sont responsables de toute information fournie au réseau afin de faciliter l'inférence.

Différentes sources d'informations telles que les lectures des capteurs peuvent être considérées comme les données fournies aux nœuds d'entrée. De plus, les nœuds résidents sont conditionnés par les valeurs fournies par le contexte et les nœuds d'entrée. De manière probabiliste, chacun de ces trois types de nœuds sont traités comme des variables aléatoires avec des distributions de probabilité prédéfinies déterminées par un expert du domaine. Il convient de noter que les ensembles C, R et I sont disjoints par paire, et G est un graphe acyclique dirigé (DAG) dont les nœuds appartiennent à $I \cup R$, et les nœuds racine sont des membres de I seulement. Enfin, les termes d'attribution de valeur de contexte en C sont utilisés pour appliquer les contraintes sous lesquelles les distributions locales s'appliquent.

5.2. Théories MEBN (MTheory)

Une collection de MFrag qui satisfait les contraintes de cohérence qui conduisent à l'existence d'une distribution de probabilité conjointe unique sur toutes les variables aléatoires de la collection est appelée MTheory [Laskey, 2008].

Un MTheory se compose de deux types de MFrag, à savoir, MFrag intégrés qui représentent le contenu logique de base, et MFrag spécifiques de domaine qui sont déterminés par une connaissance externe. Il est prouvé par Laskey dans [Laskey, 2008] que pour une Mtheory $T = \{F1, F2, \dots\}$, une distribution conjointe unique existe sur l'ensemble de ses variables aléatoires MFrag qui est cohérente avec la distribution locale assignée par les MFrag dans T.

5.3. Inférence dans le réseau bayésien multi-entité(MEBN)

Le MEBN utilise les réseaux bayésiens spécifiques à la situation (SSBN) comme moteur d'inférence. L'algorithme est basé sur des travaux de Laskey [Mahoney et Laskey, 1998] dans lesquels un SSBN minimal est approché en utilisant la combinaison de types particuliers de variables aléatoires extraites d'une requête donnée [Laskey, 2008]. Une fois le SSBN minimal construit, un algorithme bayésien standard d'inférence de réseau peut être déployé pour calculer la distribution marginale des variables aléatoires cibles (entités d'intérêt) en fonction des données probantes.

5.4. Applications du MEBN

- L'utilisation principale de MEBN est d'introduire l'incertitude dans les ontologies sous la forme d'ontologies probabilistes, MEBN a été rapidement adopté par les chercheurs pour modéliser des ontologies qui aideraient le raisonneur intégré dans les systèmes intelligents.
- Pour répondre au besoin de représentation des connaissances pour les robots autonomes qui prennent régulièrement place des humains dans des tâches triviales [Jain, et al., 2009].
- L'approche hybride consistait à utiliser PROWL comme un cadre représentatif pour représenter l'incertitude dans les ontologies [Mas, 2010].

Utiliser MEBN en aidant des décisions militaires d'attaque, basées sur des stratégies militaires et des bases de connaissances associées [Yun, et al., 2012] [Wang et al., 2012].

- Santé : Les applications du MEBN ont été développées dans le domaine du diagnostic médical pour gérer l'incertitude.

6. Conclusion

Dans ce chapitre nous avons présenté le réseau bayésien, ainsi que les réseau bayésien multi-entités (MEBN) qui permettent la modélisation et la représentation des connaissances incertaines et l'imprécision apportée par les informations d'entrée. Cela nous amène à utiliser la méthodologie UMP-ST pour modéliser les connaissances et pour la construction des ontologies probabilistes qui sera l'objectif du troisième chapitre.

CHAPITRE 3
ANALYSE ET CONCEPTION

1. Introduction

Le diagnostic médical est une tâche qui permet aux médecins de faire des prédictions sur une maladie selon les signes qui apparaissent sur un patient. Ces prédictions gèrent les connaissances incertaines qui doivent être modélisé dans une ontologie probabiliste, la construction de cette dernière se fait à l'aide de la méthodologie UMP-ST. Dans ce chapitre nous allons parler de notre motivation pour le choix de la maladie à traiter, le processus de construction d'une ontologie probabiliste, et les différentes étapes de la méthodologie UMP-ST.

2. Motivation de l'étude de cas

L'insuffisance rénale chronique (IRC) se définit par une diminution prolongée, souvent définitive, des fonctions rénales exocrines et endocrines. Elle s'exprime essentiellement par une diminution de la filtration glomérulaire (FG) avec augmentation de la créatininémie et de l'urée sanguine (urémie) par diminution de la clairance de la créatinine. Elle peut aboutir à l'insuffisance rénale terminale (IRT) qui nécessite une suppléance [épuration extra-rénale (EER)] par hémodialyse ou dialyse péritonéale et/ou par transplantation rénale [Jocelyne, 2005]. L'insuffisance rénale chronique a touché 753 millions de personnes dans le monde en 2016, dont 417 millions de femmes et 336 millions d'hommes [Bikbov, et al., 2018]. En 2015, elle a entraîné 1,2 million de décès, contre 409 000 en 1990 [Lancet, 2016].

Notre application a pour but d'aider les médecins pour la prévention contre l'insuffisance rénale chronique et la diminution de pertes humaine causée par cette maladie.

3. Le Dataset

Le dataset IRC contient les informations sur les symptômes et les facteurs de risque d'Insuffisance rénale chronique, les sources de ce dataset [Soundarapadian et al., 2015]

Nous avons choisi ce dataset pour son nombre d'instance assez bon comparés aux autres dataset disponibles gratuitement, ses attributs (11 de type numérique et 13 nominal), ses valeurs manquantes afin de pouvoir appliquer l'algorithme EM, et aussi il contient des données récentes.

Nous présentons dans le tableau 3.1 la description du dataset :

Caractéristiques du dataset	Multi varié	Nombre d'instances	400	Région	Inde
Caractéristiques Attributs	Réel	Nombre d'attributs	25	Date donnée	03/07/2015
Tâches associées	Classification	Valeurs manquantes?	Oui	Nombre de consultation Web	125759

Tableau 3. 1 : Description du dataset [Soundarapadian et al., 2015].

- **Informations sur les attributs du dataset**

- Age(numérique) : age in years
- Blood Pressure(numérique) : bp in mm/Hg
- Specific Gravity(nominal) : sg - (1.005,1.010,1.015,1.020,1.025)
- Albumin(nominal) : al - (0,1,2,3,4,5)
- Sugar(nominal) : su - (0,1,2,3,4,5)
- Red Blood Cells(nominal) : rbc - (normal, abnormal)
- Pus Cell (nominal) : pc - (normal, abnormal)
- Pus Cell clumps(nominal) : pcc - (present, notpresent)
- Bacteria(nominal) : ba - (present, notpresent)
- Blood Glucose Random(numérique) : bgr in mgs/dl
- Blood Urea(numérique) : bu in mgs/dl
- Serum Creatinine(numérique) : sc in mgs/dl
- Sodium(numérique) : sod in mEq/L
- Potassium(numérique) : pot in mEq/L
- Hemoglobin(numérique) : hemo in gms
- Packed Cell Volume(numérique) :
- White Blood Cell Count(numérique) : wc in cells/cumm
- Red Blood Cell Count(numérique) : rc in millions/cmm
- Hypertension(nominal) : htn - (yes, no)
- Diabetes Mellitus(nominal) : dm - (yes, no)
- Coronary Artery Disease(nominal) : cad - (yes, no)
- Appetite(nominal) : appet - (good, poor)
- Pedal Edema(nominal) : pe - (yes, no)

- Anemia(nominal) : ane - (yes, no)
- Class (nominal) : class - (ckd, notckd)

4. Schéma global

Notre objectif consiste à concevoir un système de prédiction sur la maladie de l'IRC dans une ontologie probabiliste puis l'aide au diagnostic médical.

Pour pouvoir utiliser notre système il faut passer par l'étape de la construction donc il y'a 2 grandes étapes :

- **Etape de construction**

Le but de cette étape est de construire notre ontologie probabiliste PR-OWL qui se base essentiellement sur l'MEBN.

La construction du MEBN nécessite l'apprentissage de structure pour la création des nœuds qui composent le MEBN (à savoir les nœuds de l'ontologie probabiliste), et aussi l'apprentissage de paramètres pour remplir les LPD de l'ontologie probabiliste.

- **Etape d'utilisation**

Le but de la partie d'utilisation est de permettre au médecin d'envoyer des requêtes sur les patients qui souhaite diagnostiquer ainsi un SSBN (Situation-Specific Bayesian Networks) est généré qui correspond à la requête demandée.

Le résultat de l'inférence est une probabilité qui sera interprété pour l'affection ou non de la maladie.

Nous présentons notre schéma global dans la **Figure 3.1**.

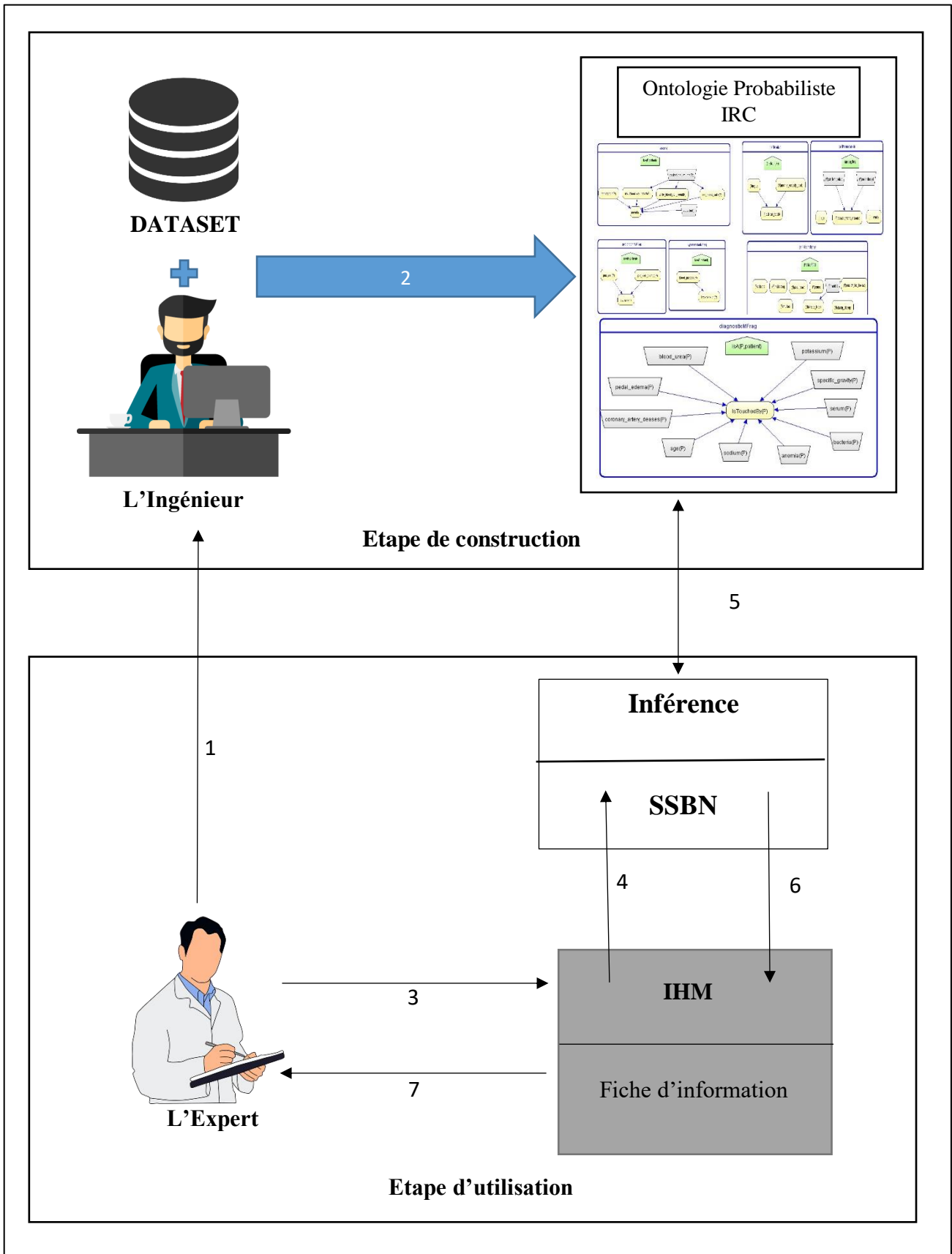


Figure 3. 1 : Schéma global.

Le tableau 3.2 contient des informations sur le schéma global.

N°	Description
1	Le médecin enrichit les connaissances de l'ingénieur avec un ensemble d'informations relatives à la maladie.
2	L'Ingénieur à l'aide des connaissances fournies par l'expert du domaine et les données obtenues du DATASET va créer l'ontologie Probabiliste.
3	Le médecin remplit la fiche d'information concernant le patient dont il veut le diagnostic et lance sa requête.
4	A partir des données de la requête du médecin l'inférence est lancée.
5	Pour réaliser l'inférence l'ontologie probabiliste est utilisée.
6	La réponse de la requête est communiquée au médecin sous forme de probabilité.
7	Le médecin reçoit et interprète le résultat du diagnostic.

Tableau 3. 2 : Description du schéma global du système.

5. Création de l'ontologie Probabiliste

L'objectif de la création de l'ontologie probabiliste est d'établir la prédiction qu'un patient est probablement touché par l'IRC, ce diagnostic provient des connaissances incertaines que l'ontologie classique ne pouvait pas traiter.

Pour la création de l'ontologie probabiliste nous avons suivis la méthodologie UMP-ST (représentée dans le 1^{er} Chapitre).

5.1. Les exigences

UMP-ST commence par la phase des exigences « Requirements », pour cette discipline, il est important de définir les questions que le modèle devrait répondre, c'est-à-dire les requêtes à poser au système en cours de conception. Pour chaque question, un ensemble d'informations (évidence) pouvant aider à répondre [Carvalho, et al., 2016].

5.1.1. Les besoins

Les besoins (Goals), veut dire les différents types de questions ou requêtes que l'utilisateur veut que le système réponde via le raisonnement de l'ontologie probabiliste (OP) [Carvalho, et al., 2016].

L'ensemble des besoins, des requêtes et des preuves (évidence qui sont tirées au prêt de l'expert du domaine) suivants concernant notre domaine de raisonnement pour qu'un patient soit touché par l'insuffisance rénale, sont :

- **Besoins** : Identifier si le patient est touché par l'insuffisance rénale ?
- **Requête 1** :
 - Le patient a-t-il les facteurs de risque de la maladie ?
- **Evidence** :
 - Le patient est Diabétique.
 - Le patient est Hypertendu.
 - Le patient a fait des Pyélonéphrites.
 - Le patient présente une maladie inflammatoire chronique (Hyper-albuminémie)
- **Requête 2** :
 - Le patient a-t-il les Symptômes de la maladie ?
- **Evidence** :
 - Le patient a l'anémie.
 - Il a une hyperkaliémie.
 - Il a des œdèmes des membres inférieurs.
 - Il a une perte d'appétit.
 - Il a une oligo-anurie.

5.2. Analyse et conception

Cette étape décrit les classes d'entités, leurs attributs, les relations entre elles et les règles qui sont applicables dans notre domaine. Cette étape est indépendante du langage utilisé pour implémenter le modèle [Carvalho, et al., 2016].

5.2.1. Les entités

C'est de décrire les classes d'entités, leurs attributs, les relations entre eux, pour que nous puissions établir les règles qui sont applicables dans notre domaine, elles sont tirées après de long discussions avec l'expert du domaine.

- **Les entités** : ce sont les classes de l'ontologie
 - Patient
 - Facteur_De_Risque.
 - Symptome.
 - Diabete
 - Hypertension
 - Anemie
 - PyelonephriteS

- **Propriétés des entités**

Les tableaux suivants porte les informations sur les différents attributs et relations des entités.

Attributs	Domaine	Range
Age	Patients	Int

Tableau 3. 3 : Les attributs de l'entité Patients.

Attributs	Domaine	Range
Anemia	Anemie	Boolean
Hemoglobin	Anemie	Float
Red Blood Cells	Anemie	Boolean
White Blood Cell Count	Anemie	Float
Red Blood Cell Count	Anemie	Float

Tableau 3. 4 : Les attributs de l'entité Anemie.

Attributes	Domaine	Range
Blood Urea	Symptomes	Float
Serum Creatinine	Symptomes	Float
Sodium	Symptomes	Float
Potassium	Symptomes	Float
Packed Cell Volume	Symptomes	Float
Pedal Edema	Symptomes	Boolean
Specific Gravity	Symptomes	Float
Appetite	Symptomes	Boolean

Tableau 3. 5 : les attributs de l'entité Symptomes.

Attributs	Domaine	Range
Albumin	Facteur_De_Risque	Int
Age	Facteur_De_Risque	Int
Coronary Artery Disease	Facteur_De_Risque	Boolean

Tableau 3. 6 : les attributs de l'entité Facteur_De_Risque.

Attributs	Domaine	Range
hypertension	Hypertension	Boolean
blood_pressure	Hypertension	Float

Tableau 3. 7 : les attributs de l'entité Hypertension.

Attributs	Domaine	Range
Diabetes Mellitus	Diabete	Boolean
Blood Glucose Random	Diabete	Float
Sugar	Diabete	Int

Tableau 3. 8 : les attributs de l'entité Diabete.

Attributs	Domaine	Range
Pus Cell	Pyélonéphrite	Boolean
Pus Cell clumps	Pyélonéphrite	Boolean
Bacteria	Pyélonéphrite	Boolean

Tableau 3. 9 : les attributs de l'entité Pyélonéphrite.

Attributs	Domaine	Range
a_facteur_de_risque	Patients	Facteur_De_Risque
a_facteur_de_risqueH	Patients	Hypertension
a_facteur_de_risqueD	Patients	Diabete
a_facteur_derisqueP	Patients	Pyelonephrite
a_symptomesA	Patients	Anemie
a_symptomes	Patients	Symptomes

Tableau 3. 10 : les relations entre les entités.

5.2.2. Les règles

Généralement, les règles probabilistes sont décrites initialement en utilisant des énoncés de probabilité qualitative. La mise en œuvre d'une ontologie probabiliste nécessite de spécifier des probabilités numériques. Les valeurs de probabilité peuvent être obtenues auprès d'experts du domaine ou tirées de l'observation.

Dans notre cas les valeurs de probabilité sont tirées du Dataset en 3 étapes : Afin d'utiliser le réseau bayésien multi entité (MEBN) qui est la base du langage PR-OWL nous devons avoir des données de type nominal. Le dataset choisi (IRC) contient des valeurs manquantes ce qui nous amène à utiliser l'algorithme EM pour donner une estimation maximale à ces valeurs, puis nous allons calculer les valeurs de probabilité de chaque nœud du MEBN.

➤ **Étapes 1** : Catégorisation des attributs numérique.

Après la collecte de données près de l'expert du domaine nous avons pu déterminer l'ensemble des intervalles pour chaque attribut.

Cette étape est nécessaire pour la construction du MEBN.

Attributs	Intervalles	Valeurs
age	[0-48] 48<	Young old
Blood Pressure	[0-88] 88<	Normal High
Blood Glucose Random	[0-126] 126<	Normal High
Blood Urea	[0-49] 49<	Normal High
Serum Creatinine	[0-12.5] 12.5<	Normal High
Sodium	[0-136] 136<	Low Normal
Potassium	[0-5.3] 5.3<	Normal High
Hemoglobin	[0-10.9] 10.9<	Low Normal
Packed Cell Volume	[0-29[[29-46.1] 46.1<	Low Normal High
White Blood Cell Count	[0-4000] 4000<	Low Normal
Red Blood Cell Count	[0-4.5] 4.5<	Low Normal

Tableau 3. 11 : Catégorisation d'attributs.

➤ **Etape 2 :** Construire la structure du réseau bayésien :

Après plusieurs discussions avec l'expert du domaine nous avons pu construire le réseau bayésien de notre système, la **Figure 3.2** représente la structure du réseau bayésien.

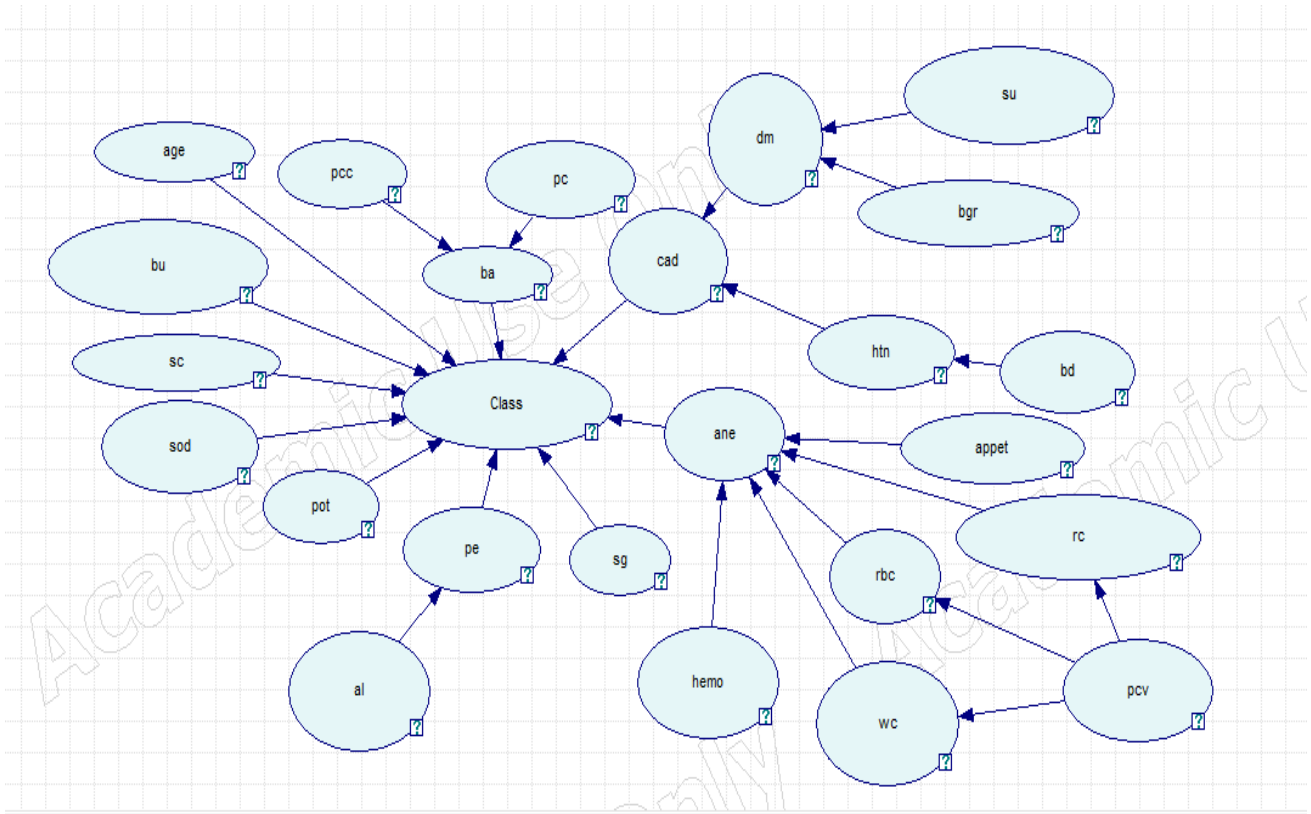


Figure 3. 2 : La structure du réseau bayésien

➤ **Etape 3 :** Application de l’algorithme EM et calcul de probabilité :

Nous avons appliqué l’algorithme EM sur notre dataset afin de donner une estimation maximale pour les valeurs de données manquante et préparer les paramètres de notre RB pour calculer les probabilités de chaque nœud.

La **Figure 3.3** montre les résultats obtenus des probabilités de chaque nœud.

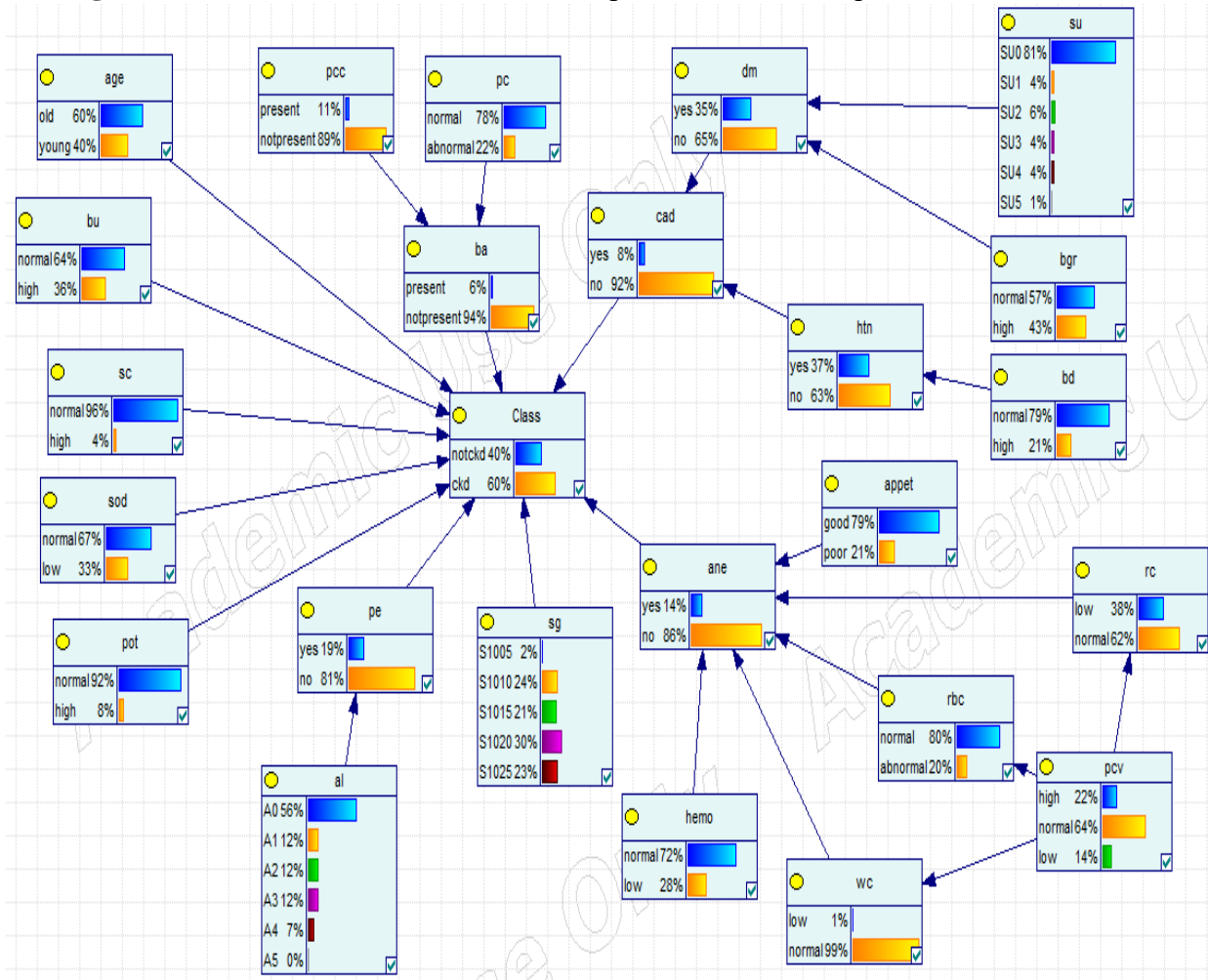


Figure 3. 3 : réseau bayésien.

5.2.3. Les groupes

Cette étape a pour objectif d'identifier les entités principales, les attributs et les relations pour notre ontologie probabiliste IRC.

Elle facilite la discipline de mise en œuvre. Plus le domaine est complexe, plus l'activité de regroupement est importante.

- **Diagramme de classe**

Le diagramme de classe est le meilleur choix pour répondre à nos besoins et faciliter la construction de notre ontologie probabiliste.

La **Figure 3.4** montre le diagramme de classe de notre système.

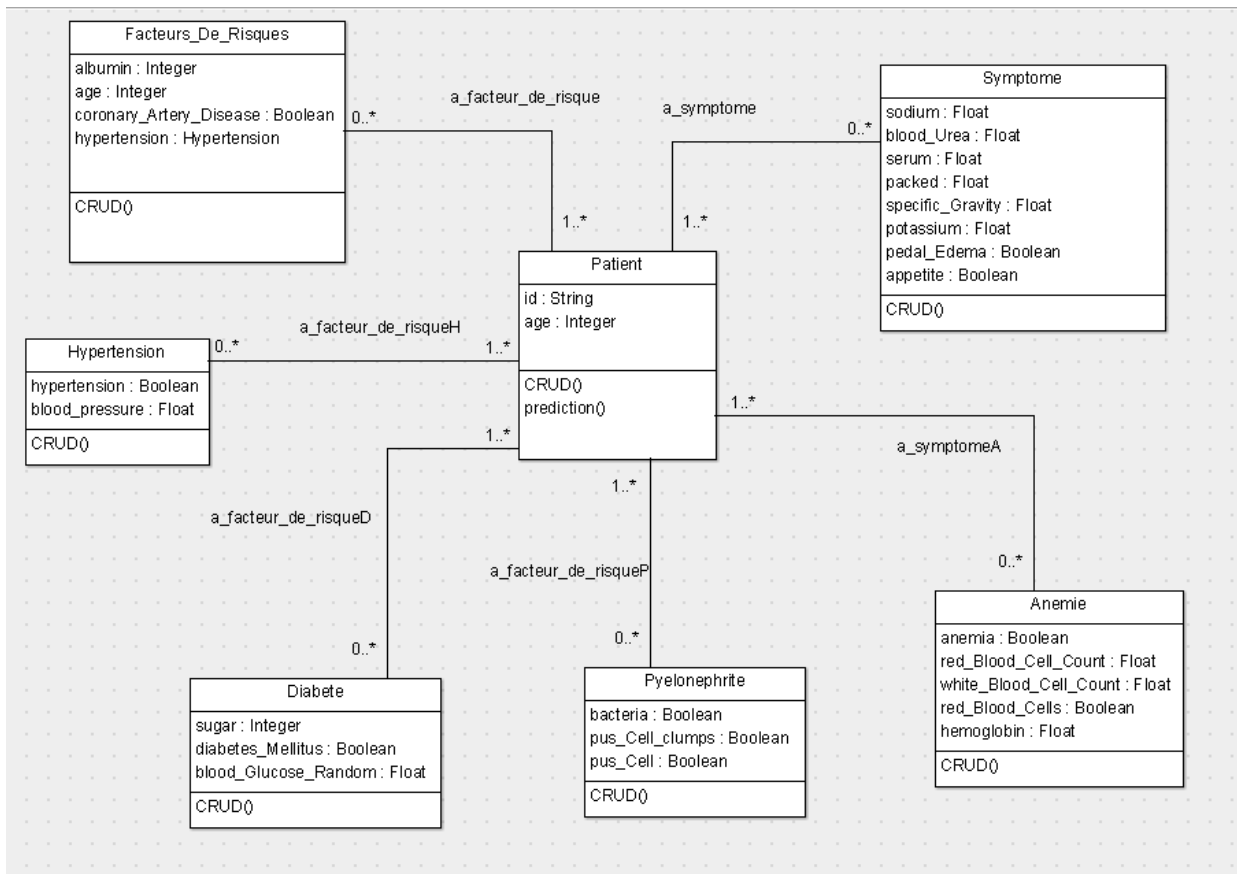


Figure 3. 4 : Diagramme de classe.

- Le patient a plusieurs **Symptome (ou pas)**, le patient peut avoir les symptômes **Anemie** ou non.
- Le patient a plusieurs **Facteurs_De_Risques (ou pas)**, ainsi que les facteurs de risques **Hypertension, Diabete** et **Pylonephrite** ou aucun.

5.3. Implémentation

Une fois que nous avons terminé la partie d'Analyse & conception, nous avons commencé à implémenter notre modèle dans un langage spécifique. Cette section décrit comment modéliser l'ontologie probabiliste IRC dans PR-OWL.

La première chose à faire est de commencer à implémenter les entités, leurs attributs et leurs relations avec PR-OWL, qui utilise essentiellement des termes MEBN. Cette partie dépend du langage utilisé.

5.3.1. Mapping

Le mapping nous permet d'utiliser PR-OWL pour raisonner de façon probabiliste sur des connaissances incertaines d'une ontologie existante basée sur les connaissances déjà disponibles [Carvalho, et al.2016].

Un modèle de probabilité est défini comme un ensemble de fragments MEBN (MFrag), chaque MFrag exprimant une incertitude quant à un petit nombre d'attributs et / ou de relations entre entités. Un ensemble de MFrag correctement définis pris ensemble constitue une théorie MEBN (MTheory), qui peut exprimer une distribution de probabilité conjointe sur des situations complexes impliquant de nombreuses entités du domaine. [Carvalho, .et al.2016].

- **Mtheory :**

Un MTheory est un ensemble de MFrag qui satisfait collectivement des contraintes de cohérence assurant l'existence d'une distribution unique de probabilité conjointe sur les instances des variables aléatoires représentées dans chacune des MFrag dans l'ensemble [Carvalho, .et al.2016].

La création de la MTheory est la première étape pour construire une ontologie probabiliste, il se fait juste après l'importation de l'ontologie qu'on veut modéliser, ce MTheory va contenir les différents MFrag dont notre ontologie probabiliste aura besoin.

- **Mfrag :**

Après la création de notre MTheory, nous commençons par la création des MFrag qui contiennent les nœuds (nœuds résidents, nœuds de contexte, nœuds d'entrées) qui sont les variables aléatoires. La **Figure 3.5** montre les MFrag de domaine :

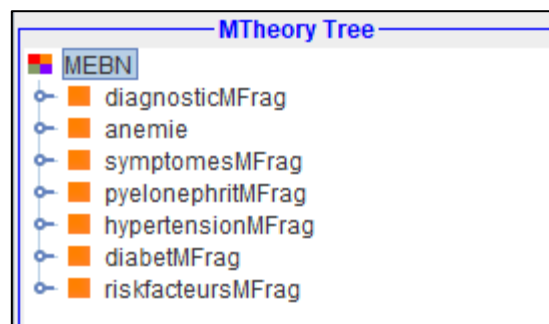


Figure 3. 5 : MTheory Tree.

- **Variables ordinaires**

Pour la déclaration des variables ordinaires on aura besoin des classes OWL que nous avons déjà extrait. Une variable ordinaire est déclarée avec la relation 'est-un' (*IsA*), c'est une instanciation d'une entité (classe OWL). Par exemple : *IsA* (*p*, *Patient*) tels que *p* est notre variable ordinaire et *Patient* est l'entité (la classe OWL).

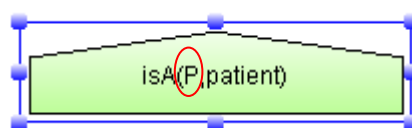


Figure 3. 6 : Variable ordinaire « P ».

- **ResidentNode (nœuds résidents)**

Ce sont les variables aléatoires réelles qui forment le cœur du sujet d'un MFragment. La logique MEBN exige que la distribution probabiliste locale de chaque nœud résident doit être unique et explicitement défini dans son MFragment original. Les valeurs possibles d'un nœud résident peuvent être une entité existante ou des attributs.

Voici un exemple de nœud résident dans le MFragment « HypertensionMFragment ».

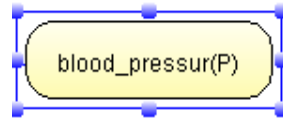


Figure 3. 7 : Nœud résident "blood_pressur".

- **ContextNode (nœud de contexte)**

Ce sont des variables aléatoires booléennes représentant des conditions qui doivent être satisfaites pour faire une distribution dans un MFragment valide. Les nœuds de contexte peuvent représenter plusieurs types de modèles d'incertitude, tels que l'incertitude sur les relations entre les entités. Si on peut inférer de la base de connaissances (c'est à dire l'ontologie) que la valeur d'un nœud de contexte est vraie, la distribution de probabilité du MFragment sera appliquée dans le modèle d'inférence. Si cette valeur est fausse, une distribution par défaut sera utilisée. Si la valeur est inconnue, le nœud de contexte devient pratiquement un parent de tous les nœuds résidents dans le même MFragment.



Figure 3. 8 Nœud de contexte "IsA()".

- **InputNode (nœud d'entre)**

Ces nœuds sont essentiellement des pointeurs faisant référence aux nœuds résidents d'autre MFragment. Les nœuds d'entrée fournissent également un mécanisme permettant la réutilisation des nœuds résidents entre les MFragment. Les nœuds d'entrée influencent sur la distribution de probabilité des nœuds résidents qui sont leurs enfants dans un autre MFragment, mais leurs propres distributions sont définies ailleurs (dans leurs propres MFragment d'origine). Par exemple : le nœud « age » dans le MFragment « diagnostiqueMFragment » [Figure3.16] fait référence au nœud résident du MFragment « Facteur_De_RisqueMFragment » [Figure3.10].

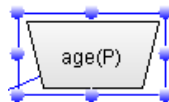


Figure 3. 9 nœud d'entre "age".

▪ Les Types des relations du réseau bayésien

Le type de relation entre les nœuds de la structure du réseau est de type causal, on distingue deux types de nœud dans une relation de causalité :

- Relation de causalité entre les nœuds d'entrées avec des nœuds résidents.
- Relation de causalité entre les nœuds résidents avec d'autres nœuds résidents.

Après les définitions des différents nœuds de MFrag nous allons montrer chaque MFrag en détail.

La **Figure 3.10** montre les différents nœuds du MFrag « Facteur_De_Risque ».

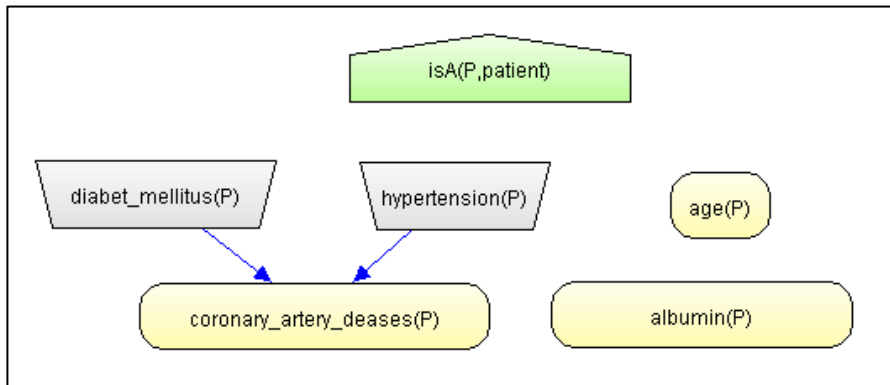


Figure 3. 10 : MFrag "Facteur_De_Risque".

La **Figure 3.11** montre les différents nœuds du MFrag « SymptomesMFrag ».

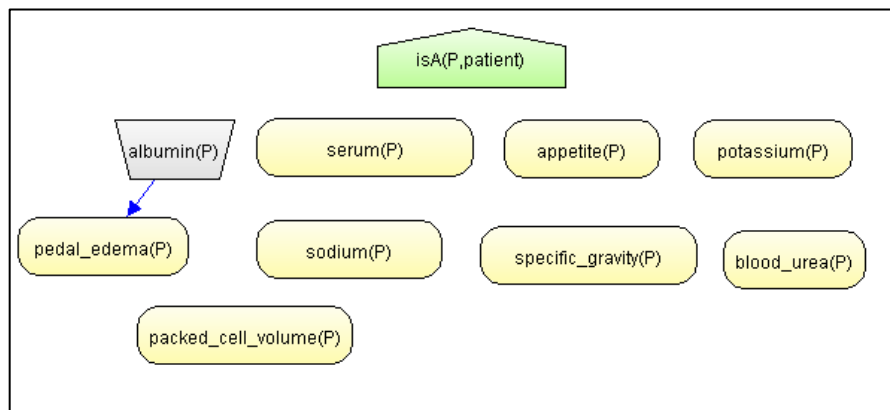


Figure 3. 11 : MFrag "SymptomesMFrag".

La **Figure 3.12** montre les différents nœuds du MFragment « PylonephritisMFragment ».

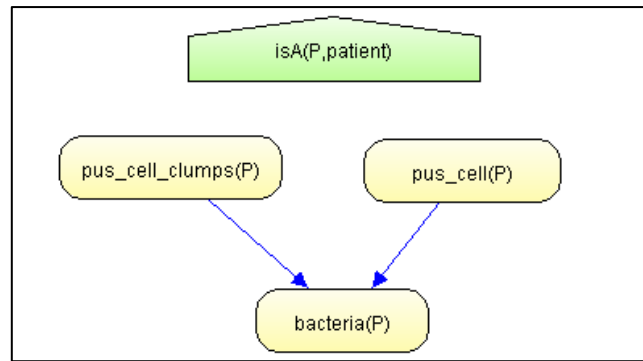


Figure 3. 12: MFragment "PylonephritisMFragment".

La **Figure 3.13** montre les différents nœuds du MFragment « DiabetMFragment ».

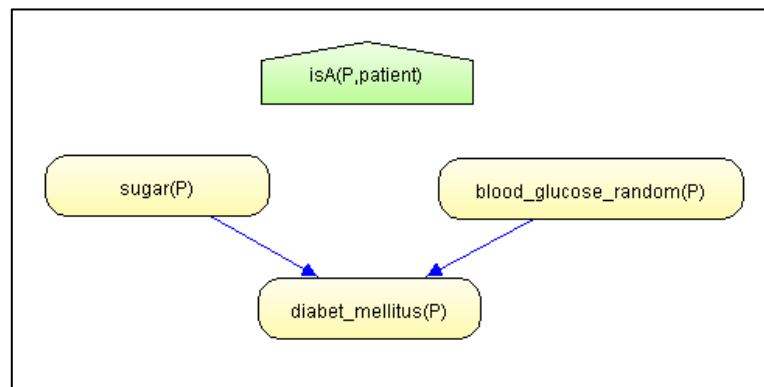


Figure 3. 13 : MFragment "DiabetMFragment".

La **Figure 3.14** montre les différents nœuds du MFragment « HypertensionMFragment ».

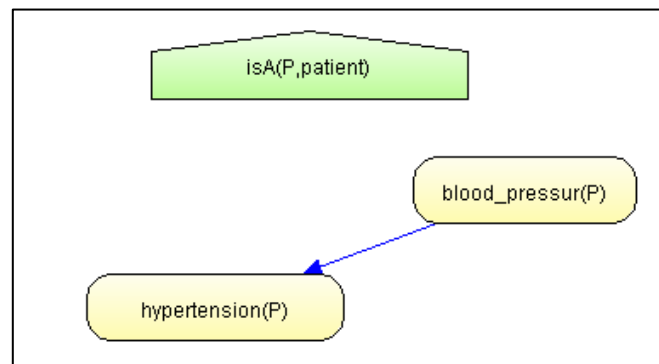


Figure 3. 14 : MFragment "HypertensionMFragment".

La **Figure 3.15** montre les différents nœuds du MFragment « AnemieMFragment ».

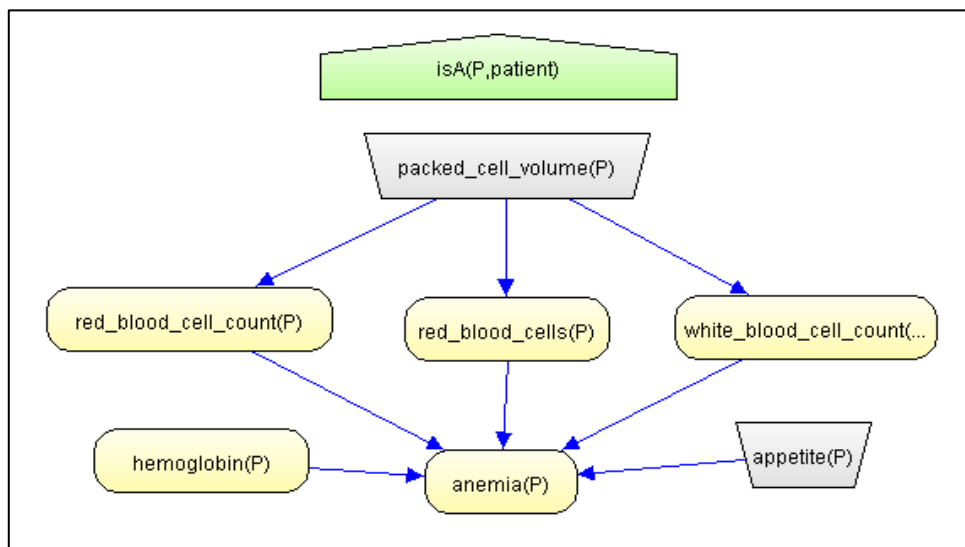


Figure 3. 15 : MFragment "AnemieMFragment".

La **Figure 3.16** montre les différents nœuds du MFragment « DiagnosticMFragment ».

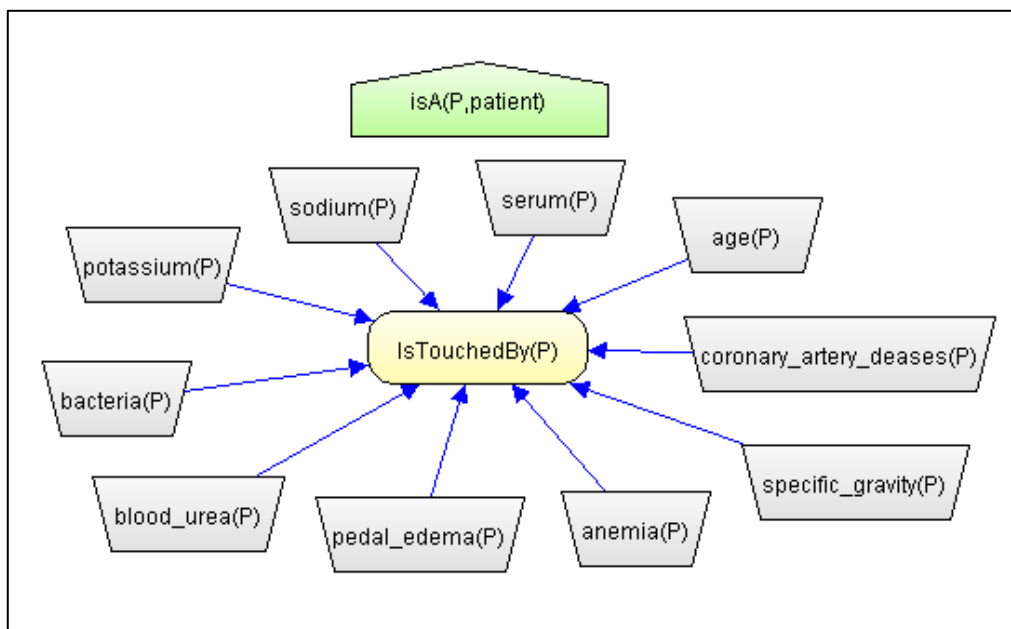


Figure 3. 16 : MFragment "DiagnosticMFragment".

5.3.2. Probabilités

La dernière étape dans la construction d'une ontologie probabiliste est de définir les distributions de probabilité locales (LPD) pour toutes les variables aléatoires (VR) résidentes à partir des règles définies au préalable.

Chaque MFragment définit les LPD pour ses nœuds résidentiels, Ces distributions sont conditionnées par la satisfaction des contraintes exprimées par les variables de contextes. Les distributions

locales peuvent dépendre des valeurs des variables d'entrées, dont les distributions sont définies dans leurs MFrag d'origine.

Voici LPD du MFrag IsTouchedBy qui répond à notre requête principale si le patient a l'IRC ou non (voire la **Figure 3.17**) :

The screenshot shows the LPD editor for the MFrag 'IsTouchedBy'. The interface is divided into three main sections:

- Nodes:** A list of variables used in the rules: specific_gravity, potassium, bacteria, sodium, coronary_artery_deases, blood_urea, age, serum, anemia, and pedal_edema.
- States:** A section for defining states, currently empty.
- Arguments:** A section for defining arguments, currently empty.

The main editing area contains a series of rules defined using logical expressions and probabilities. The rules are structured as follows:

```

if any P have ( anemia = no & coronary_artery_deases = no & bacteria = notpresent & specific_gravity = S1010 & pedal_edema = yes & potassium = normal & sodium = normal & serum = normal & blood_urea = normal & age = old) [
  no = 0.26,
  yes = 0.74,
  absurd = 0
]
else if any P have ( anemia = no & coronary_artery_deases = yes & bacteria = notpresent & specific_gravity = S1010 & pedal_edema = yes & potassium = normal & sodium = normal & serum = normal & blood_urea = normal & age = old) [
  no = 0.37,
  yes = 0.63,
  absurd = 0
]
else if any P have ( anemia = no & coronary_artery_deases = yes & bacteria = notpresent & specific_gravity = S1015 & pedal_edema = yes & potassium = normal & sodium = normal & serum = normal & blood_urea = normal & age = old) [
  no = 0.316,
  yes = 0.684,
  absurd = 0
]
else if any P have ( anemia = no & coronary_artery_deases = no & bacteria = notpresent & specific_gravity = S1015 & pedal_edema = yes & potassium = normal & sodium = normal & serum = normal & blood_urea = normal & age = old) [
  no = 0.24,
  yes = 0.76,
  absurd = 0
]
else if any P have ( anemia = no & coronary_artery_deases = yes & bacteria = notpresent & specific_gravity = S1020 & pedal_edema = yes & potassium = normal & sodium = normal & serum = normal & blood_urea = normal & age = old) [
  no = 0.36,
  yes = 0.64,
  absurd = 0
]
else if any P have ( anemia = no & coronary_artery_deases = no & bacteria = notpresent & specific_gravity = S1020 & pedal_edema = yes & potassium = normal & sodium = normal & serum = normal & blood_urea = normal & age = old) [
  no = 0.28,
  yes = 0.72,
  absurd = 0
]
else if any P have ( anemia = no & coronary_artery_deases = yes & bacteria = notpresent & specific_gravity = S1025 & pedal_edema = yes & potassium = normal & sodium = normal & serum = normal & blood_urea = normal & age = old) [
  no = 0.37,
  yes = 0.63,
  absurd = 0
]
else if any P have ( anemia = yes & coronary_artery_deases = no & bacteria = notpresent &

```

At the bottom of the editor, there are three buttons: Save, Compile, and Exit.

Figure 3. 17 : LPD du MFrag isTouchedBy.

La **Figure 3.18** présente les résultats d'une seule combinaison d'états pour les variables aléatoires des patients sans préciser ses évidences.

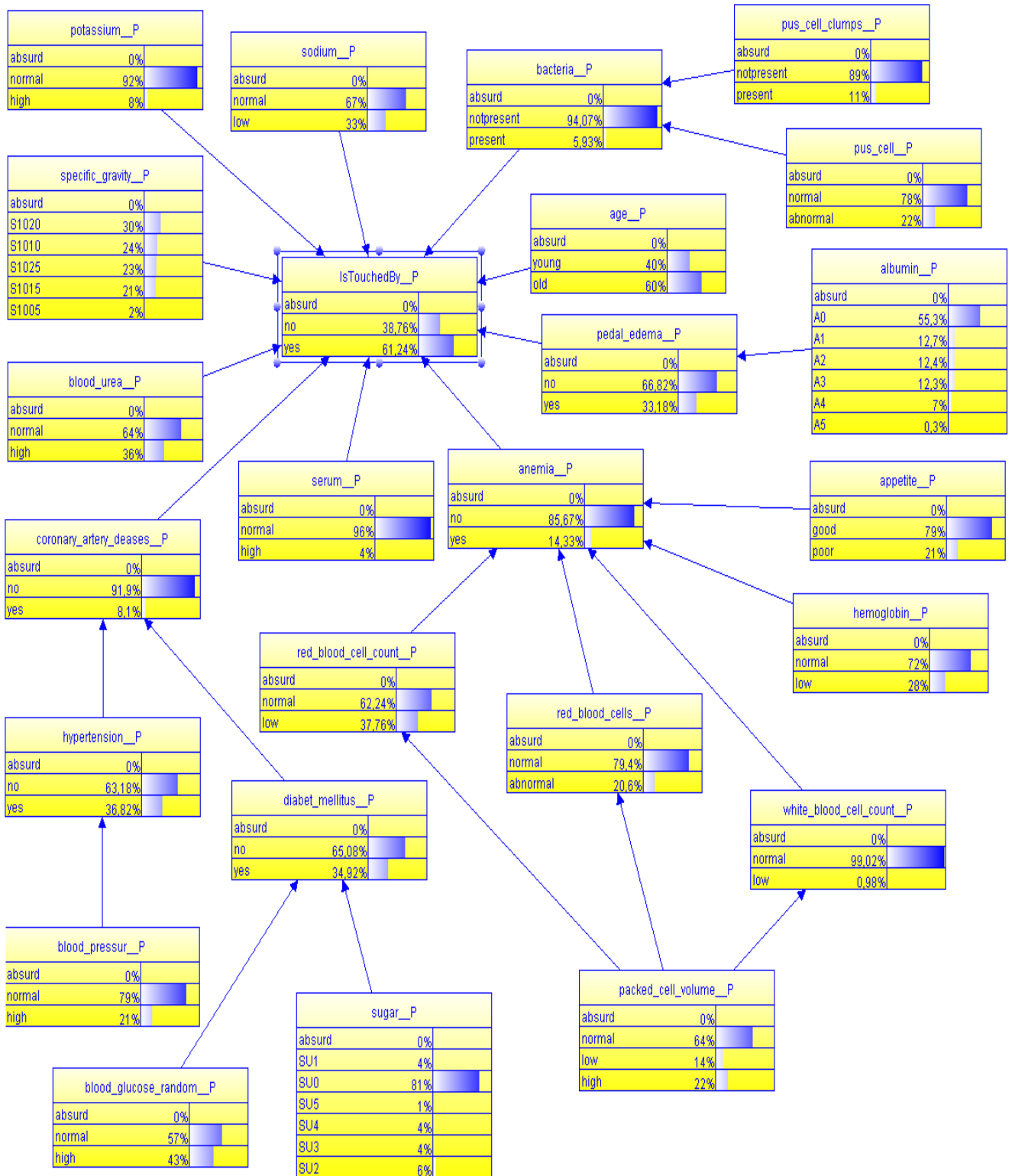


Figure 3. 18: Résultat SSBN pour le nœud IsTouchedBy(Patient P).

5.3.3. Test :

L'objectif des tests est de voir le bon fonctionnement de notre système et de trouver des défauts à notre système.

Avant de commencer à décrire les activités dans la discipline « Test », il est important de comprendre les différents types d'évaluation qui doivent être effectués. La littérature distingue deux types d'évaluation la vérification et la validation. D'une part, la vérification concerne la fourniture de toutes les fonctionnalités promises au client. Cela implique généralement l'examen des exigences, de la documentation, de la conception et du code. La vérification se fait souvent par des inspections et par des listes de contrôle. D'autre part, la validation concerne le bon comportement du système. La validation est le test réel du système et elle est effectuée après la vérification [Carvalho, .et al.2016].

Evaluation : Dans cette partie, nous nous concentrerons sur l'évaluation fondée sur des cas, qui définira des différents scénarios pour tester notre modèle. Nous voulons tester le comportement d'une partie du modèle, plus précisément, en vérifiant comment la variable centrée se comporte avec un ensemble de données distinct. Dans le cas de PR-OWL, nous pouvons analyser le comportement des variables aléatoires d'intérêt fournies par MFrag. Ce test MFrag est important pour saisir la cohérence locale du modèle [Carvalho, .et al.2016].

Vérification : La vérification concerne l'établissement de « le système a été construit correctement », c'est-à-dire que les éléments du système sont conformes à leurs spécifications de performance définies [Carvalho, .et al.2016].

Dans notre cas d'étude nous vérifierons essentiellement la façon dont la requête IsTouchedBy (Patient) se différencie entre les Tests. Nous définissons deux différents scénarios pour illustrer la mécanique de la définition et du test d'un scénario. Le premier est celui qui montre les évidences concernant les facteurs de risques sans influence et le second avec l'influence des facteurs de risques.

Validation : La validation vise à établir que le « bon système a été construit », c'est-à-dire qu'il réalise son utilisation prévue dans son environnement opérationnel [Carvalho, .et al.2016].

Lors de la définition d'un scénario, il est important de définir l'hypothèse à tester et quel est le résultat attendu, en plus de fournir les preuves qui seront utilisées.

- Pour commencer nous avons créé les nouvelles instances de la classe Patient.

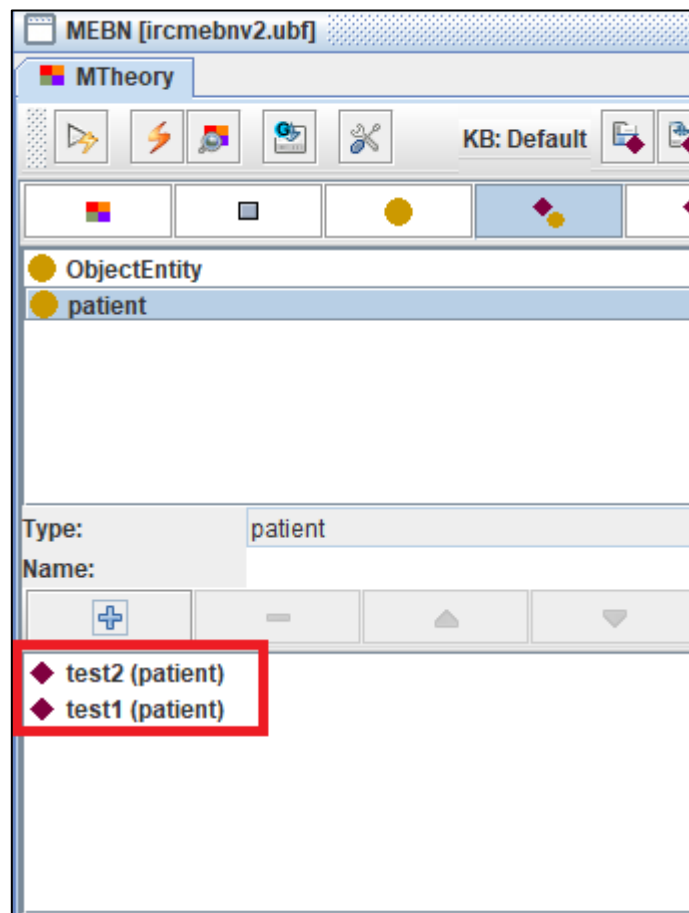


Figure 3. 19 : Instance de la classe Patient.

❖ **Test1** : Nous allons définir le 1er scénario à tester.

Hypothèse a testée : IsTouchedBy(Patient).

Résultat attendu : Le patient n'es pas touché par l'IRC.

Evidence (Facteur de risque(FR),Symptomes(SYM)) :

Le patient est jeune. (FR)

Le patient n'as pas d'anémie. (SYM)

Le patient n'es pas hypertendu. (FR)

Le patient n'as pas d'inflammation interne ou externe des membre inferieurs (pas de bactérie). (SYM)

Le patient a un taux de potassium normal. (SYM)

Le patient a un taux de sodium normal. (SYM)

- Après la création des instances nous allons commencer par charger les données du patient (test1) dans la base de connaissance puis nous générerons le SSBN (Inférence).

La **Figure 3.20** montre le résultat obtenu.

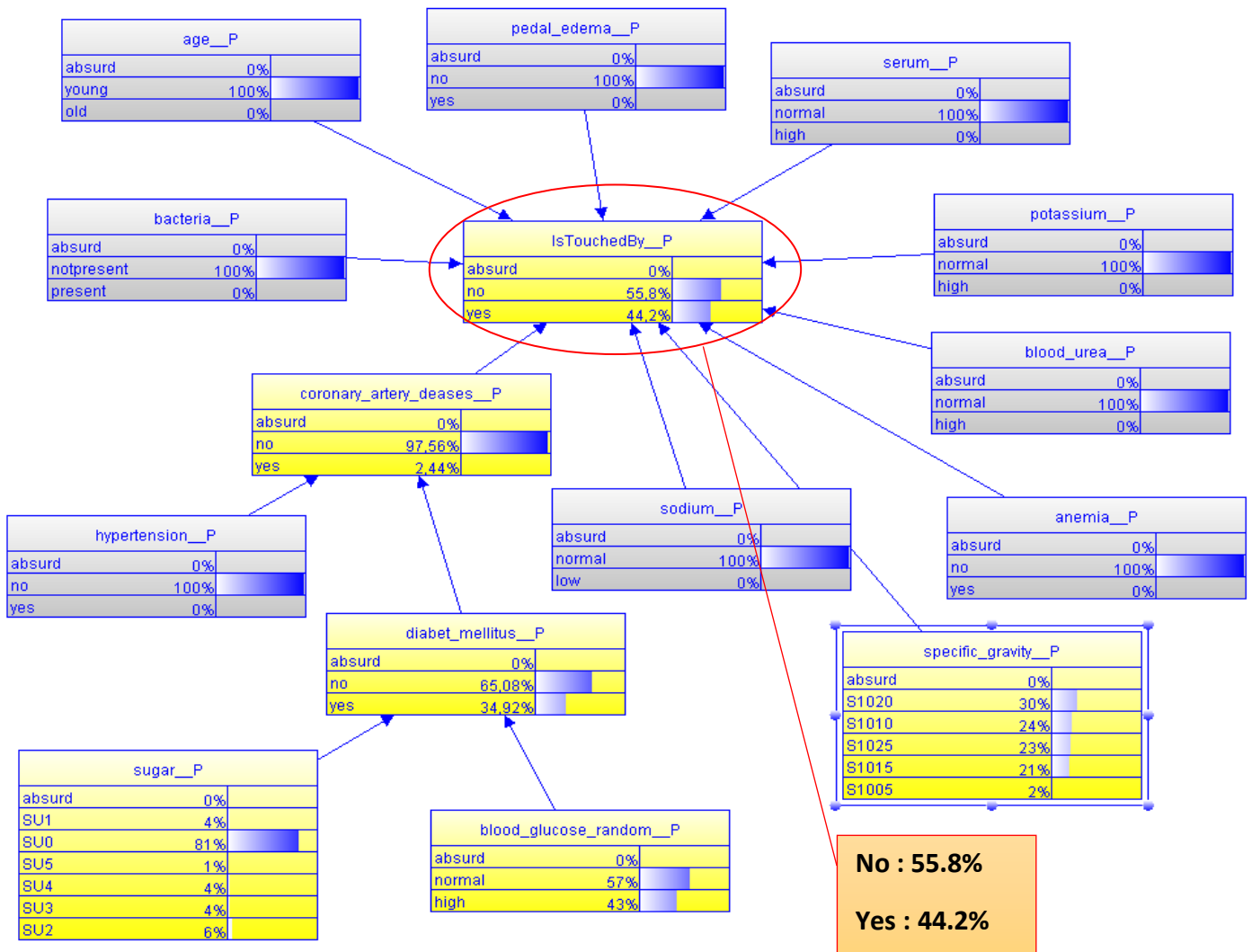


Figure 3. 20 : SSBN test1.

❖ **Test2** : Nous allons définir le 2em scénario à tester.

Hypothèse a testée : IsTouchedBy(Patient).

Résultat attendu : Le patient est touché par l'IRC.

Evidence :

Le patient est âgé. (FR)

Le patient est touché par l'anémie. (SYM)

Le patient est hypertendu. (FR)

Le patient a des inflammation interne ou externe des membre inferieurs (bactérie présente). (SYM)

Le patient a un taux de potassium élevé. (SYM)

Le patient a un taux de sodium bas. (SYM)

- Après la création des instances nous allons commencer par charger les données du patient (test2) dans la base de connaissance puis nous générerons le SSBN (Inférence).

La **Figure 3.21** montre le résultat obtenu.

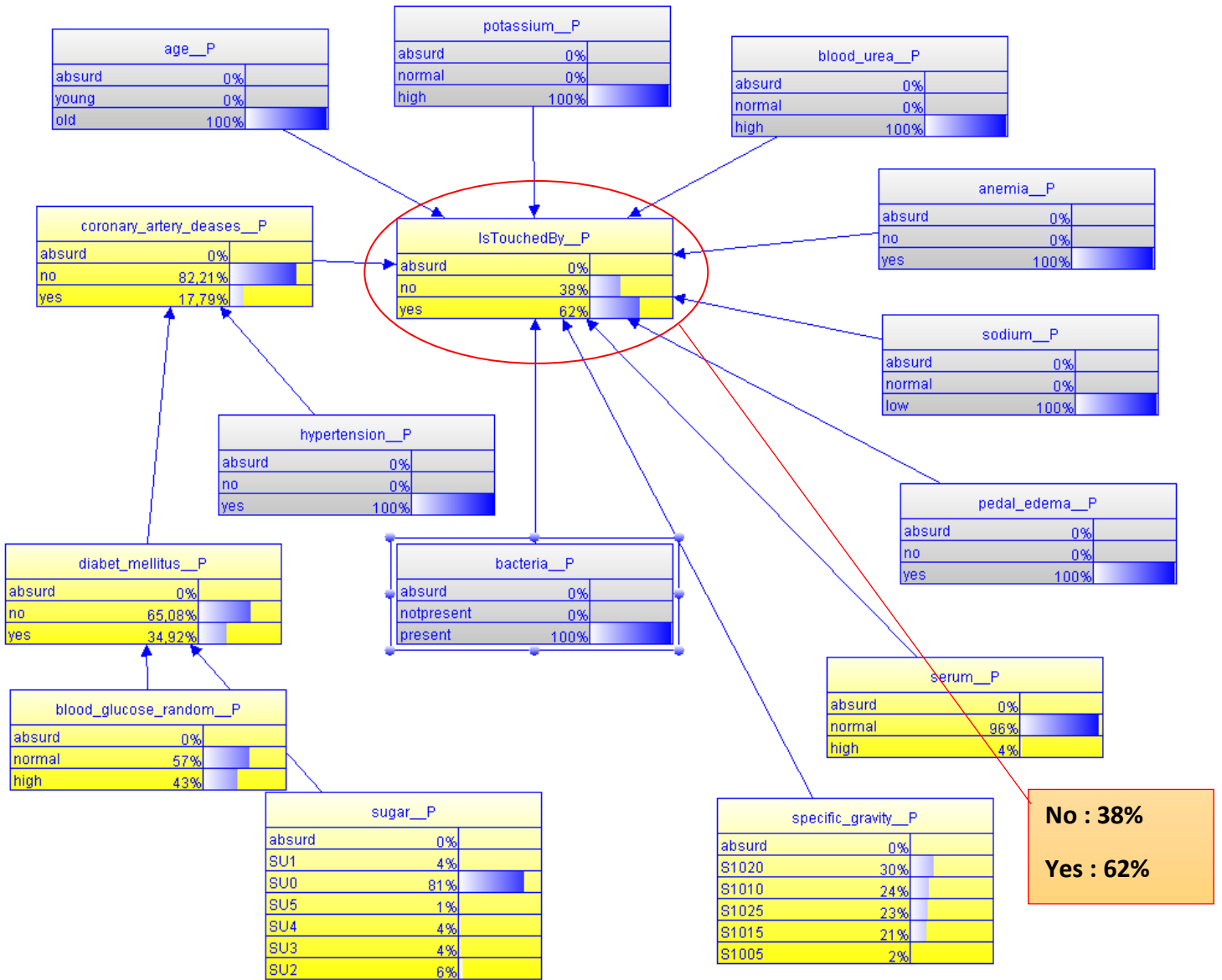


Figure 3. 21 : SSBN Test2.

- Les résultats obtenus des deux tests on était validé par l'expert du domaine.

6. Résultat final de l'ontologie probabiliste :

Après avoir suivie les étapes de la méthodologie UMP-ST, nous avons obtenu une ontologie probabiliste qui porte le MTheory avec ses différents MFrag.

7. Conclusion :

Dans ce chapitre nous avons montré le schéma de notre système ainsi que la construction de l'ontologie probabiliste IRC on suivant les étapes la méthodologie UMP-ST et arrivé à avoir une prédiction sur la maladie.

Dans le prochain chapitre nous allons passer à la réalisation de notre système d'aide au diagnostic médical.

CHAPITRE 4

Implémentation et test

1. Introduction :

Après la réalisation de toutes les étapes de la conception nous allons commencer à implémenter notre ontologie probabiliste, ainsi notre système qui permet de faire l'inférence probabiliste dans un langage spécifique dans PR-OWL en utilisant UnBBayes. Dans ce chapitre nous allons décrire de façon visuelle l'implémentation de notre système, en effectuant des captures d'écran des différentes interfaces du système.

2. Les différents outils de développement utilisé pour notre système :

Nous allons vous présenter les différents outils utilisés lors de l'implémentation de notre système.

- **Protégé :**

Protégé est un éditeur d'ontologies et de bases de connaissances produit par l'Université de Stanford. Protégé est un outil qui permet la construction d'ontologies de domaine, de formulaires de saisie de données personnalisés pour la saisie de données complètement gratuites.

Protégé permet la définition de classes, hiérarchies de classes, variables, restrictions de valeurs variables et les relations entre les classes et les propriétés de ces relations.

Protégé est livré avec des packages de visualisation tels que OntoViz, EZPal, etc.; tout cela aide l'utilisateur à visualiser les ontologies à l'aide de diagrammes. Stanford L'université fait un travail magnifique d'amélioration continue de Protégé.

Dans le cadre de sa dernière mise à jour, Protégé inclut maintenant une interface pour le langage de règle Web sémantique (SWRL), qui se situe au-dessus de OWL mathématiques, raisonnement temporel et ajoute des règles de raisonnement de type Prolog. Stanford a un tutoriel qui couvre la base de l'utilisation de Protégé avec le plug-in OWL [Kapoor et Sharma., 2010].

- **Protégé_OWL API**

C'est une bibliothèque Java à code source ouvert pour le langage OWL (Web Ontology Language) et RDF (S). L'API fournit des classes et des méthodes pour charger et enregistrer des fichiers OWL, pour interroger et manipuler des modèles de données OWL et pour effectuer un raisonnement basé sur les moteurs de logique de description. De plus, l'API est optimisée pour la mise en œuvre d'interfaces utilisateur graphiques. [Knublauch, 2010].

- **OWL API**

La nécessité d'avoir un outil fiable pour manipuler des ontologies, notamment dans un environnement dynamique, a poussé les informaticiens à développer OWL-API. Il s'agit d'un ensemble d'interfaces riches en fonctionnalité, permettant une manipulation flexible d'ontologies. De nombreux projets utilisent OWL-API comme outil de développement. A titre d'exemple, on peut citer Protégé-4, Swoop et OntoTrack. [Sami, 2016].

- **UnBBayes API**

UnBBayes est un framework de réseau probabiliste écrit en Java. Il a à la fois une interface graphique et une API avec inférence, échantillonnage, apprentissage et évaluation. Il supporte BN, ID, MSBN, OOBN, HBN, MEBN / PR-OWL, structure, paramètre et apprentissage progressif [Cardialfly, et al.,2011].

- **Java**

Java est un langage de programmation orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy, présenté officiellement le 23 mai 1995 au SunWorld.

La société Sun a été ensuite rachetée en 2009 par la société Oracle qui détient et maintient désormais Java. La particularité et l'objectif central de Java est que les logiciels écrits dans ce langage doivent être très facilement portables sur plusieurs systèmes d'exploitation tels que Unix, Windows, Mac OS ou GNU/Linux, avec peu ou pas de modifications, mais qui ont l'inconvénient d'être plus lourd à l'exécution (en mémoire et en temps processeur) à cause de sa machine virtuelle. Pour cela, divers plateformes et Framework associés visent à guider, sinon garantir, cette portabilité des applications développées en Java.

- **Eclipse IDE (Integrated Development Environment) :**

Eclipse est un environnement de développement extensible et polyvalent, initié par IBM en 2001, conçu autour d'une plateforme commune à laquelle s'agrègent des composants dérivatifs, le projet est ainsi constitué de nombreux sous-projets spécifiques aux technologies sous-jacentes. L'objet de la solution « Eclipse » est de fournir des outils favorisant la productivité, mais pas seulement celle qui concerne le codage logiciel. On y trouve des environnements de développement intégré mais également de conception, de modélisation, de tests...

Eclipse a beau être écrit en Java, il peut être utilisé pour développer sous n'importe quel langage de programmation. Eclipse est publié sous la licence EPL (Eclipse Public License), la fondation Eclipse en est désormais le porteur officiel.

- **GeNIe**

GeNIe Modeler est une interface utilisateur graphique (GUI) pour SMILE Engine et permet la construction de modèles interactifs et l'apprentissage. Il est écrit pour l'environnement Windows mais peut également être utilisé sous Mac OS et Linux sous les émulateurs Windows. Il a été testé de manière approfondie sur le terrain depuis 1998, a été largement accepté par les universités et l'industrie et compte des milliers d'utilisateurs dans le monde entier. Un critère de conception important pour tous les produits a été dès le début qu'ils devraient permettre une liberté de modélisation complète. Les modèles GeNIe ne plient pas la réalité aux outils de modélisation disponibles. Tout ce que le domaine exige, peut être modélisé dans notre logiciel. Parce qu'il n'existe pas d'algorithmes exacts pour certains types de modèles, ce logiciel est équipé d'une suite d'algorithmes d'échantillonnage stochastique approximatifs, capables de résoudre tous les modèles créés par les utilisateurs.

3. La mise en œuvre du système :

Après avoir effectué les étapes de la méthodologie UMP-ST nous avons obtenu un fichier avec une extension UBF qui porte le MTheory avec ses différents MFragments.

Le fichier PLM (base de connaissances) est généré automatiquement à partir du fichier UBF passé comme paramètre.

4. La base de connaissance :

La figure 4.1 montre un modèle d'une Base de connaissances (Fichier PLM).

```
(DEFMODULE "/PL-KERNEL-KB/PL-USER/GENERATIVE_MODULE_3/FINDINGS_MODULE_3"
:CASE-SENSITIVE? FALSE)

(IN-MODULE "/PL-KERNEL-KB/PL-USER/GENERATIVE_MODULE_3/FINDINGS_MODULE_3")

(IN-DIALECT :KIF)

(ASSERT (PATIENT_LABEL AABHA))
(ASSERT (PATIENT_LABEL AABHA1))
(ASSERT (PATIENT_LABEL ADITI))
(ASSERT (PATIENT_LABEL AABHA2))
(ASSERT (= (ALBUMIN AABHA) A0))
(ASSERT (= (ALBUMIN AABHA1) A0))
(ASSERT (= (ALBUMIN ADITI) A1))
(ASSERT (= (ALBUMIN AABHA2) A2))
(ASSERT (= (CORONARY_ARTERY_DEASES AABHA) YES))
(ASSERT (= (CORONARY_ARTERY_DEASES AABHA1) YES))
(ASSERT (= (CORONARY_ARTERY_DEASES ADITI) NO))
(ASSERT (= (CORONARY_ARTERY_DEASES AABHA2) NO))
(ASSERT (= (AGE AABHA) OLD))
(ASSERT (= (AGE AABHA1) YOUNG))
(ASSERT (= (AGE ADITI) YOUNG))
(ASSERT (= (AGE AABHA2) YOUNG))
(ASSERT (= (PUS_CELL_CLUMPS AABHA) PRESENT))
(ASSERT (= (PUS_CELL_CLUMPS AABHA1) PRESENT))
(ASSERT (= (PUS_CELL_CLUMPS ADITI) NOTPRESENT))
(ASSERT (= (PUS_CELL_CLUMPS AABHA2) NOTPRESENT))
(ASSERT (= (PUS_CELL AABHA) ABNORMAL))
(ASSERT (= (PUS_CELL AABHA1) ABNORMAL))
(ASSERT (= (PUS_CELL ADITI) NORMAL))
(ASSERT (= (PUS_CELL AABHA2) NORMAL))
(ASSERT (= (BACTERIA AABHA) NOTPRESENT))
(ASSERT (= (BACTERIA AABHA1) PRESENT))
(ASSERT (= (BACTERIA ADITI) PRESENT))
(ASSERT (= (BACTERIA AABHA2) PRESENT))
(ASSERT (= (PACKED_CELL_VOLUME AABHA) HIGH))
(ASSERT (= (PACKED_CELL_VOLUME AABHA1) HIGH))
(ASSERT (= (PACKED_CELL_VOLUME ADITI) HIGH))
(ASSERT (= (PACKED_CELL_VOLUME AABHA2) HIGH))
(ASSERT (= (SERUM AABHA) NORMAL))
(ASSERT (= (SERUM AABHA1) NORMAL))
(ASSERT (= (SERUM ADITI) NORMAL))
(ASSERT (= (SERUM AABHA2) NORMAL))
```

Figure 4. 1 : Fichier PLM.

5. Les interfaces :

Nous allons montrer les différentes interfaces de notre application.

❖ Interface d'accueil

C'est la 1^{ère} Interface de notre système elle contient les boutons des différentes fonctionnalités de notre application ainsi que la date et l'heure (Figure 4.2).



Figure 4. 2 Interface d'accueil.

❖ Interface « Add Patient » : Contient les informations sur le patient (Figure 4.3).

Add a patient

-Name :

Risk Factor :

Album...

C_A... Yes No

-Age ...

Blood Pres...

Hypertensi... Yes No

Sugar

Blood glucose ran...

Diabet mellitus : Yes No

Pus c... Normal Abnor...

Pus cell clu... Pres... Notpresent

Bacteri... Pres... Notpres...

Symptoms :

Potassiu...

Serum :

Yes No

Go... Poor

Packed cell volu...

Hymoglobin :

White blood cell co...

Red blood cell count :

Red blood cel... Nor... Abnor...

Anemi... Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor... Abnor...

Yes No

Nor...

❖ **Interface « Consult Patient »** : Contient les données du patient choisis (Figure 4.4)

The screenshot shows a window titled 'Consultation' with a 'General Consultation' section. It contains two columns of data entry fields. The left column is labeled 'Risk Factors' and the right column is labeled 'Symptoms'. Each field contains a value, and an 'OK' button is located at the bottom right.

Risk Factors :		Symptoms :	
-Name :	AABHA	Potassium :	normal
Albumin :	A0	Serum :	normal
C_A_D :	yes	Pedal edema :	yes
-Age :	old	Blood Urea :	high
Blood Pressur :	yes	Appetit :	good
Hypertension :	normal	Specific Gravity :	S1005
Sugar :	SU0	Sodium :	normal
Blood glucose ran... :	normal	Packed cell volume :	high
Diabet mellitus :	yes	Hymoglobin :	normal
Pus cell :	abnormal	White blood cell count :	low
Pus cell clumps :	notpresent	Red blood cell count :	normal
Bacteria :	notpresent	Red blood cells :	abnormal
		Anemia :	yes

Figure 4. 4 Interface « Consult Patient »

❖ **Interface « Delete Patient »** : Nous permet de supprimer un patient (Figure 4.5).

The screenshot shows a 'Delete Patient' interface with a background image of a doctor. A 'Message' dialog box is open in the center, displaying the text 'Delete succesful!' and an 'OK' button. Below the dialog, there is a 'Delete Patient' button, a dropdown menu showing 'AABHA1', and 'Delete' and 'Cancel' buttons. The top of the window shows 'File Help' and a timestamp: 'Saturday, September 21, 2019 4:41:04 PM WAT'.

Figure 4. 5 Interface « Delete Patient »

❖ **Interface « Prediction »** : Contient un message d'avertissement que l'opération prendra quelque instants (Figure 4.6), les résultats de la prédiction ainsi que l'aide au diagnostic en affichant les facteurs de risques et leurs valeurs de probabilités (Figure 4.7), et aussi le diagramme « Pie Chart » qui affiche le résultat de la prédiction sous forme de diagramme circulaire (Figure 4.8).

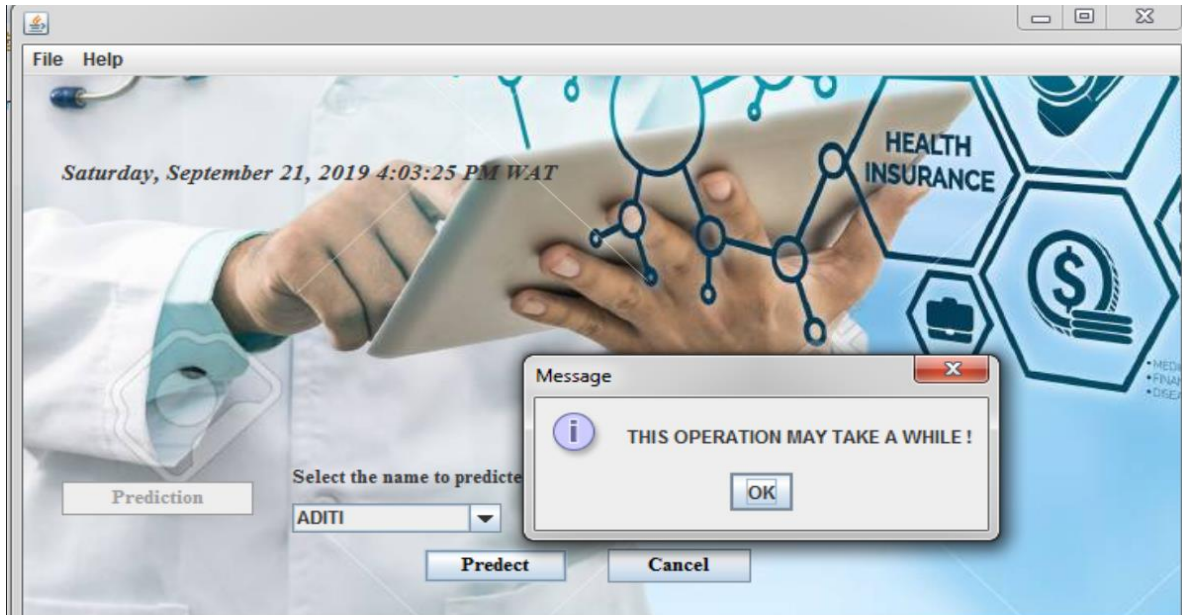


Figure 4. 6 Message « WARNING »

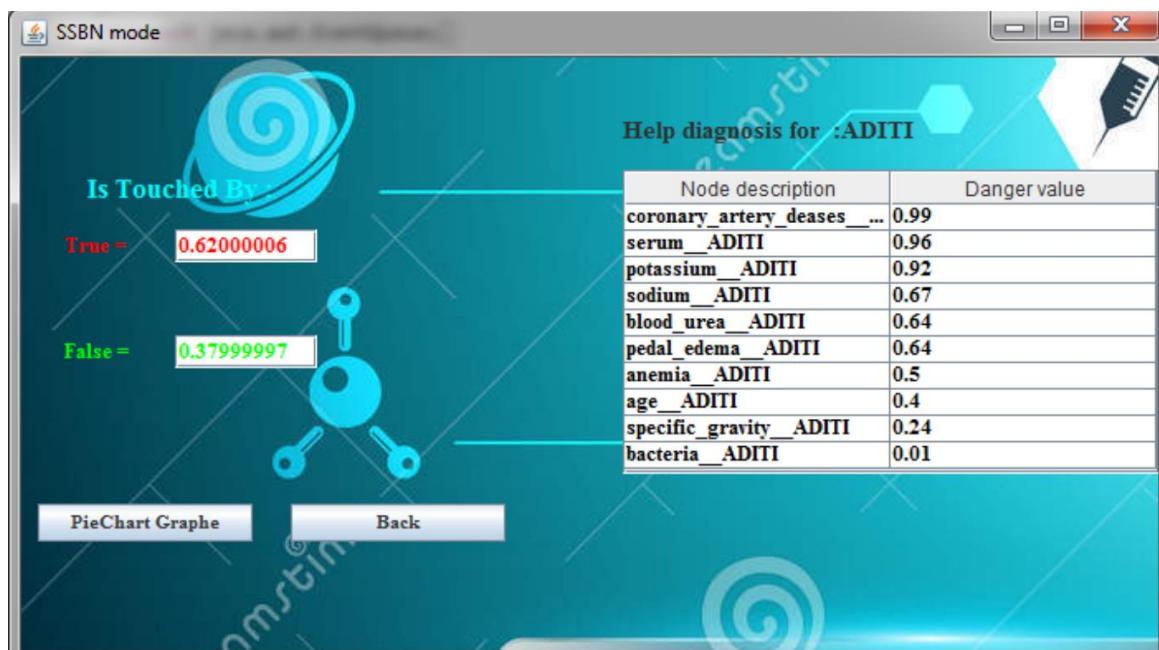


Figure 4. 7 Interface « Prediction »

❖ **Interface « PieChart »** : c'est la représentation graphique du résultat de la prédiction sous forme de diagramme circulaire (Figure 4.8).

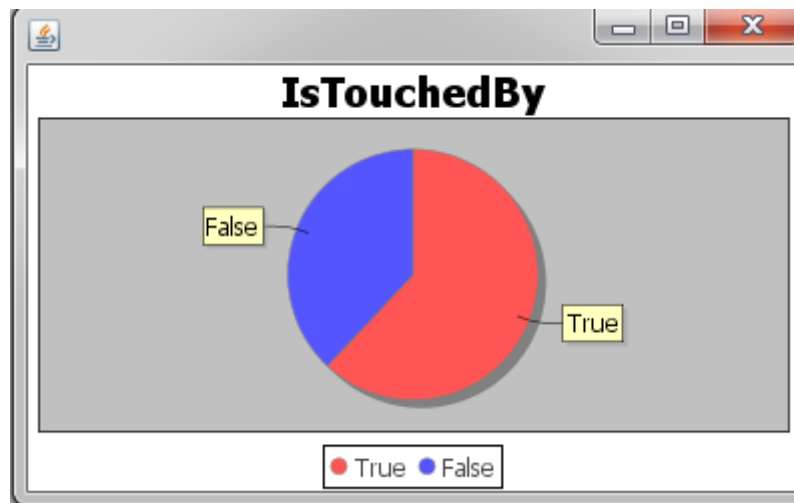


Figure 4. 8 Interface « Prediction »

6. La validation croisée

La validation croisée est un moyen pour prédire l'efficacité d'un modèle sur un ensemble de validation hypothétique lorsqu'un ensemble de validation indépendant et explicite n'est pas disponible, pour minimiser l'influence du choix du partitionnement de l'ensemble des exemples, la validation croisée subdivise l'ensemble d'entraînement initial en k sous ensemble disjoints D_1 , D_2 , ..., D_k de même taille. L'entraînement et le test sont effectués k fois. A l'itération i le sous-ensemble D_i est réservé pour le test et le reste des exemples sont utilisés pour entraîner le modèle. La précision finale du modèle est égale à la moyenne des k précisions de test.

Dans notre cas $K = [2, \dots, 10]$ et le calcul des précisions du système se fait suivant les étapes suivantes :

À chaque itération, le modèle est entraîné sur $k - 1$ parties, et testé sur la partie restante. On obtient à la fin k précisions, la précision du modèle est égale à leur moyenne.

La figure 4.9 montre comment la validation croisée calcule la précision du système.

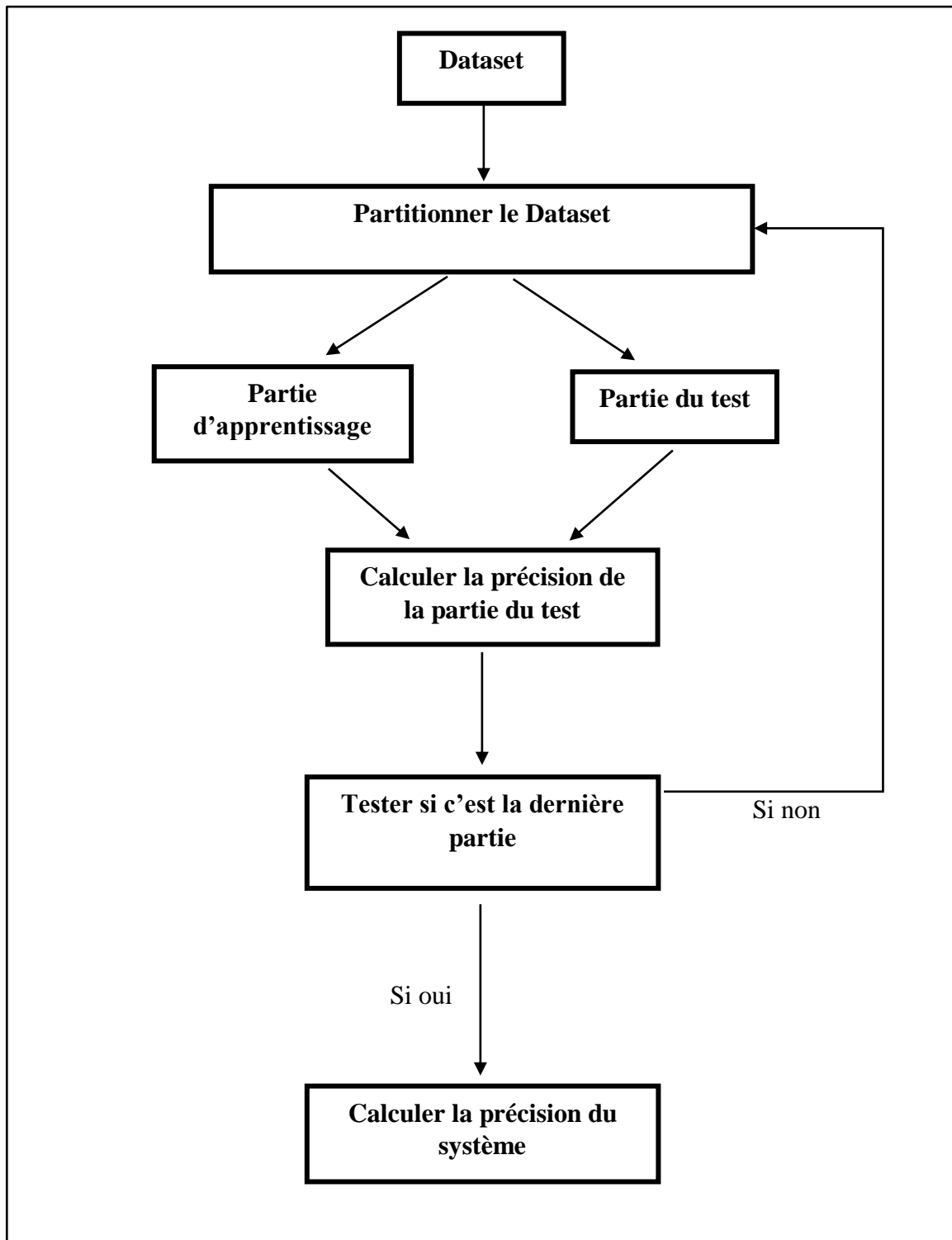


Figure 4. 9 Calculer la précision du système

➤ **Résultat de la validation croisée :**

Nous avons donné plusieurs valeurs à K, pour confirmer notre résultat.

Le calcul des performances a donné les résultats suivants (Tableau 4.1) :

Valeurs K	Précision
2	95 %
3	94.25%
4	94.5%
5	94.75%
6	94.75%
7	94.5%
8	94.25%
9	95%
10	94.75%

Tableau 4. 1 Précision du système

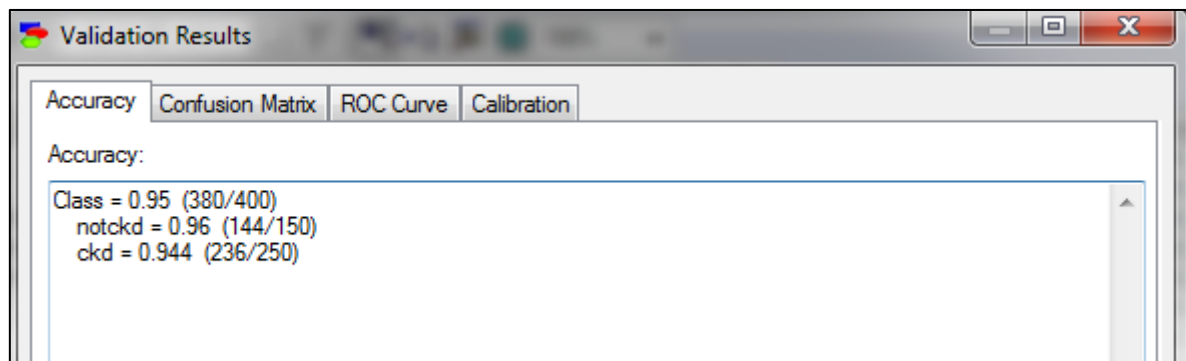


Figure 4. 10 : Exemple d'un résultat de validation (K=2)

- Dans l'état « oui » où le patient est touché par l'IRC 236 résultats obtenues de la validation croisée similaires avec des patients du dataset qui est égale à 250 « état=ckd », donc ça nous donne un pourcentage de 94.4 %.
- Dans l'état « non » où le patient est saint 144 résultats obtenues de la validation croisée similaires avec les états des patients du dataset qui égale à 150 « état= notckd », donc ça nous donne un pourcentage 96%.
- La somme de ces deux états nous donne 380 résultats obtenues similaires avec les patients du dataset qui est égale à 400 dans les deux cas ou les patients sont touchés (ckd) ou non (notckd), avec un pourcentage de 95%.

Ce résultat de validation croisé montre le taux de précision de notre système, un bon système c'est un système qui a une précision proche de 1.

Dans notre cas nous avons obtenus un bon résultat qui est entre 95% de précision.

7. Conclusion :

Dans ce chapitre nous avons implémenté notre système d'aide au diagnostic après avoir importé l'ontologie probabiliste et la base de connaissance, avec des captures de notre application et nous avons terminé par une validation croisée sur les résultats obtenus.

Conclusion générale

Dans ce mémoire, nous avons présenté un système conçu pour l'aide au diagnostic médical, il est prouvé que les ontologies probabilistes 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 derniers.

Le domaine médical est fondé sur les diagnostics où les connaissances sont incertaines, les ontologies classiques n'intègrent pas les connaissances incertaines, alors que les ontologies probabilistes jouent un rôle important dans la représentation de l'incertitude.

La création de notre ontologie probabiliste est réalisée à l'aide de la méthodologie UMP-ST en suivant toutes ses étapes.

La structure du MEBN a été modélisée à l'aide de l'expert du domaine, en énumérant les différents facteurs de risques ainsi que les liens de causalité entre eux, alors que l'apprentissage des paramètres du MEBN a été réalisé à partir de d'un ensemble de données (dataset) avec des données manquantes ce qui nous a amené à utiliser l'algorithme EM.

L'ontologie probabiliste PR-OWL est compatible avec OWL, cette ontologie basée sur les réseaux bayésiens multi entités est capable de prédire tout en tenant compte des incertitudes autour des données.

Notre cas d'étude sert à prédire, si un patient est touché par l'insuffisance rénale chronique, tout en sachant ses facteurs de risque et ses tests qui sont représentés sous la forme d'un SSBN contenant un ensemble de nœuds, et défini par des distributions de probabilité locales (LPD), en utilisant l'inférence.

Afin de réaliser notre conception, nous avons implémenté notre système, en implémentant l'ontologie probabiliste de l'insuffisance rénale, ainsi, nous avons montré les différentes interfaces de notre système. La validation des résultats de notre système en utilisant la validation croisée donne de bonne valeur de précision.

Nos perspectives c'est de rendre le système capable à traiter des données numériques, et de tester notre système sur une base de données réelle.

Références bibliographiques

[Berardi et al., 2001]: Berardi, D., Calvanese, D., & De Giacomo, G. (2001, September). Reasoning on UML class diagrams using description logic based systems. In *Proc. of the KI'2001 Workshop on Applications of Description Logics* (Vol. 44).

[Bikbov, et al., 2018] : Bikbov B, Perico N, Remuzzi G (23 May 2018). "Disparities in Chronic Kidney Disease Prevalence among Males and Females in 195 Countries: Analysis of the Global Burden of Disease 2016 Study". *Nephron*.

[Brachman et al., 1985]: Brachman, R. J., & Schmolze, J. G. (1985). An overview of the KL-ONE knowledge representation system. *Cognitive science*, 9(2), 171-216.

[Brickley et al., 2004]: Brickley, D., & Guha, R. V. (2004). RDF vocabulary description language 1.0: RDF schema.

[Brickley et al., 2004]: Brickley, D., & Guha, R. V. (2004). RDF vocabulary description language 1.0: RDF schema.

[Cardialfly, et al., 2011]: cardialfly, laecio_lima, mladeira, rommelnc , <https://sourceforge.net/projects/unbbayes/> ,2011, copnsulté le 19/09/2019.

[Carvalho et al., 2016]: Carvalho, R. N., Laskey, K. B., & Da Costa, P. C. (2016). Uncertainty modeling process for semantic technology. *PeerJ Computer Science*, 2, e77.

[Costa et al., 2006] : Costa, P. C., & Laskey, K. B. (2006). PR-OWL: A framework for probabilistic ontologies. *Frontiers in Artificial Intelligence and Applications*, 150, 237.

[Didier et Henri, 2001] : Didier Dubois et Henri Prade. « La problématique scientifique du traitement de l'information ». In : *Information-Interaction-Intelligence 1.2* (2001), pages 79–98.

[Didier et Henri, 2009] : Didier Dubois et Henri Prade. « Formal representations of uncertainty ». In: *Decision-Making Process: Concepts and Methods* (2009), pages 85–156. (Cité en page 33).

[Farquhar et al., 1997] : Farquhar, A., Fikes, R., & Rice, J. (1997). The ontolingua server: A tool for collaborative ontology construction. *International journal of human-computer studies*, 46(6), 707-727.

[Favero, 1994]: Favero, r. F. (1994). « backward simulation in bayesian networks ».

[Gomez et al., 2004]: Gómez-Pérez, A., Fernandez-Lopez, M., Corcho, O.: Ontological Engineering with Examples from the Areas of Knowledge Management, e-Commerce and the Semantic Web. First Edition Series: Advanced Information and Knowledge Processing 1st ed. 2004. 2nd printing, 2004, XII, 403 p. 159 illus., Hardcover (2004). ISBN: 978-1-85233-551-9

[Hartel et al., 2005] : Hartel, F. W., de Coronado, S., Dionne, R., Fragoso, G., & Golbeck, J. (2005). Modeling a description logic vocabulary for cancer research. *Journal of biomedical informatics*, 38(2).

[Intan et al., 2015] : Intan, R., Chi, C. H., Palit, H. N., & Santoso, L. W. (Eds.). (2015). *Intelligence in the Era of Big Data: 4th International Conference on Soft Computing, Intelligent Systems, and Information Technology, ICSIT 2015, Bali, Indonesia, March 11-14, 2015. Proceedings* (Vol. 516). Springer.

[Jain et al., 2009]: D. Jain, L. Mosenlechner, and M. Beetz, “Equipping robot control programs with firstorderprobabilisticreasoningcapabilities,” 2009 IEEE Int. Conf. Robot. Autom., pp. 3626–3631, 2009.

[Jocelyne, M., 2005] : Docteur Jocelyne MAURIZI-BALZAN, Professeur Philippe ZAOUÏ - Mars 2004 (Mise à jour juin 2005)

[Judea, 1988]: P. Judea. (1988). Probabilistic reasoning in intelligent systems : networks of plausible inference. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA,.

[Kapoor et Sharma., 2010]: Kapoor, B., & Sharma, S. (2010). A comparative study ontology building tools for semantic web applications. *International journal of Web & Semantic Technology (IJWesT)*, 1(3).

[Kerdjoudj, 2015]: KERDJOU DJ, Fadhela. *Gestion de l'incertitude dans le processus d'extraction de connaissances à partir de textes*. 2015. Thèse de doctorat. Paris Est.

[**Knublauch, 2010**]: Holger Knublauch, protégé wiki, https://protegewiki.stanford.edu/wiki/ProtegeOWL_API_Programmers_Guide, June 30 2010, consulté le 19/09/2019.

[**Lancet., 2016**] : GBD 2015 Mortality and Causes of Death, Collaborators. (8 October 2016). "Global, regional, and national life expectancy, all-cause mortality, and cause-specific mortality for 249 causes of death, 1980-2015: a systematic analysis for the Global Burden of Disease Study 2015". *Lancet*. 388 (10053): 1459–1544.

[**Laskey, 2008**]: K. B. Laskey, MEBN: A language for first-order Bayesian knowledge bases, *Artificial Intelligence*, 172(2008):140-178.

[**Mahoney et Laskey, 1998**] :S. M. Mahoney and K. B. Laskey, "Constructing situation specific belief networks," in *In Proc. 14th Conf. on Uncertainty in Artificial Intelligence*, 1998, pp. 370–378.

[**Mas, 2010**]: M. D. Mas, "Ontology Temporal Evolution for Multi-Entity Bayesian Networks under Exogenous and Endogenous Semantic Updating," *CoRR*, vol. abs/1009.2, pp. 1–20, 2010.

[**McGuinness et al., 2004**]: McGuinness, D. L., & Van Harmelen, F. (2004). OWL web ontology language overview. *W3C recommendation*, 10(10), 2004.

[**Naïm et al., 2007**] : P. Naïm, P-H. Willemin, P. Leray, O. Pourret, and A. Becker. Réseaux bayésiens. 423 pages, Eyrolles, Paris, 2007.

[**Naïm et al., 2011**] : Patrick Naïm, Pierre-Henri Willemin, Philippe Leray, Olivier Pourret, Anna Becker – Eyrolles, Paris, 2011.

[**Naïm, 1999**]: P. Naïm, A. Becker. Les reseaux bayesiens. 199 pages, Eyrolles, Paris, 1999

[**Neeches et al., 1991**] : Neches, R., Fikes, R. E., Finin, T., Gruber, T., Patil, R., Senator, T., & Swartout, W. R. (1991). Enabling technology for knowledge sharing. *AI magazine*, 12(3), 36.

[**Nizza et al., 2013**] : De Nizza, D., Ortiz, J., Meurisse, H., & Schobbens, P. Y. (2013, July). Formalisation et Construction d'une Ontologie dans le Domaine des Infections Orthopédiques. In *IC-24èmes*

[Paulo et al., 2016]: Paulo C. G. Costa & Kathryn B. Laskey, PR-OWL: A Bayesian extension to the OWL Ontology Language, 01/04/2016. <http://www.pr-owl.org/mebn/index.php>.

[Pham et al., 2008] : Pham, T. A. L. (2008). *De l'optimisation à la décomposition de l'ontologie dans la logique de description* (Doctoral dissertation, Nice).

[Sami, 2016]: Samir AMIR, ContentSide , <https://www.contentside.com/owl-api-une-api-java-pour-manipuler-des-ontologies/> ,(2016) consulté le 19/09/2019.

[Soundarapadian et al. , 2015] : Dr.P.Soundarapandian.M.D.,D.M , Apollo Hospitals, Managiri, Madurai Main Road, Karaikudi, Tamilnadu, India, 03/07/2015.

[Staab et al., 2000] : Staab, S., & Maedche, A. (2000, August). Ontology engineering beyond the modeling of concepts and relations. In *Proceedings of the ECAI-2000 Workshop on Ontologies and Problem-Solving Methods*. Berlin.

[Swartout et al., 1997]: Swartout, William & Patil, Ramesh & Knight, Kevin & Russ, Tom. (1997). Toward Distributed Use of Large-Scale Ontologies t.

[Uschold et al., 1996] : Uschold, M., & Gruninger, M. (1996). Ontologies: Principles, methods and applications. *The knowledge engineering review*, 11(2).

[Wang et al., 2012] Wang, Hao-Ran, et al. "Tactical Air Target Intention Recognition Based on MultiEntitiesBayesian Network." *HuoliyuZhihuiKongzhi* 37.10 (2012): 133-138.

[Wang, 2004]: Wang, Pei. "The limitation of Bayesianism." *Artificial Intelligence* 158.1 (2004): 97-106

[Yun, et al., 2012]: Z. Yun, Z. Cheng, L. Ting, Z. Weiming, and L. Zhong, "A COG Analysis Model of System-of-Systems (SoS) Based on Multi-EntityBayesian Networks (MEBN)," 2012.

f)