

UNIVERSITÉ DE BLIDA 1

Faculté des Sciences

Département d'Informatique

THÈSE DE DOCTORAT

Option : Sciences Informatiques et de Données.

**MODÉLISATION DE LA CONNAISSANCE INCERTAINE
DANS LE WEB SÉMANTIQUE.**

Par

Riali Ishak

devant le jury composé de :

N. Boustia	Professeur, U. de Blida 1	Présidente
H. Bouarfa	Professeur, U. de Blida 1	Rapporteur
M. Fareh	Maître de conférences, U. de Blida 1	Co-Directrice de thèse
A. R. Ghomari	Professeur, E.S.I, Alger	Examineur
Dr. L. Zemmouchi	Maître de conférences, E.N.S.T, Alger	Examinatrice

Blida, le 29 septembre 2020

Je dédie mes travaux de thèse :

À mes chers parents

À toute ma famille

À tous mes amis...

REMERCIEMENTS

"La reconnaissance est la mémoire du cœur."

En premier lieu, je rends louange à Allah le tout Puissant de m'avoir donné le courage, la volonté et de m'avoir éclairé le chemin du savoir ainsi que de m'avoir entouré par des personnes positives qui m'ont beaucoup aidé afin d'accomplir mon travail.

Je tiens à exprimer ma gratitude et mes plus vifs remerciements à ma directrice de thèse Mme Abed Hafida, Professeur en informatique à USD-Blida 1, pour son aide, son écoute, sa disponibilité et ses précieux conseils qu'elle m'a prodigué tout au long de ce parcours de recherche. L'autonomie qu'elle m'a laissé, m'ont permis de mener à bien mon travail.

Mes remerciements s'adresseront aussi à ma co-directrice de thèse Mme Fareh Messaouda, Maître de conférences en informatique à USD-Blida 1, pour toute sa disponibilité, sa patience, son écoute, son esprit critique, enthousiasme et confiance dont elle m'a gratifié et ses précieux conseils avisés qui ont nourri mes réflexions tout au long de l'élaboration de mon travail. Qu'elle trouve dans ces lignes si courtes l'expression de ma très grande gratitude.

Je tiens à remercier Pr. N. Boustia, pour m'avoir fait l'honneur d'accepter de présider mon jury de thèse.

J'exprime toute ma gratitude aux membres du jury Pr. A. R. Ghomari, Dr. L. Zemmouchi qui nous feront l'honneur d'évaluer mon travail.

J'exprime toute ma reconnaissance à mes parents Mohammed et Fatma, à mes soeurs et mes frères. Grâce à vos encouragements, votre soutien que j'ai pu avancer.

Mes vifs remerciements à mes ami(e)s doctorant(e)s, notamment, les soeurs jumelles Benks-sirat Amina et Selma, Mouzai Meriem et Chaba Mouna Mustapha qui m'ont soutenu tout au long de la réalisation de mon projet et qui ont fait de ma vie en tant que doctorant un énorme plaisir.

Mes gratitudes s'adressent aussi à mes enseignants du département d'informatique qui n'ont pas lésiné à nous transmettre leur savoir et leur savoir-faire.

A la fin mais non moins profonds, je présente mes reconnaissances et sincères remerciements à mes amis et à toutes les personnes qui ont contribué de près ou loin à l'élaboration de mon travail ...

Résumé

Ces dernières années, l'utilisation des ontologies a reçu une attention considérable pour conceptualiser un domaine d'application ou pour assurer l'interopérabilité sémantique entre des systèmes hétérogènes, notamment dans le web sémantique. Cependant, les ontologies classiques restent incapables et inefficaces pour le traitement des problèmes dont les connaissances sont incertaines.

Notre travail de thèse s'inscrit dans une perspective de modélisation et du raisonnement sur des connaissances incertaines représentées dans des ontologies riches en termes d'incertitude. En effet, le but ultime est de développer une nouvelle approche qui permet de traiter des connaissances probabilistes et des connaissances floues encapsulées dans une ontologie d'un domaine d'application.

Pour atteindre nos objectifs : (1) nous avons proposé une extension des réseaux MEBN (Multi Entity Bayesian Network) en nous basant sur la logique floue que nous avons par la suite nommée FzMEBN (FzMEBN : FuZzy Multi Entity Bayesian Network), afin d'enrichir leurs capacités de modélisation et de raisonnement probabiliste. (2) nous avons introduit une nouvelle approche pour résoudre le problème de l'incertitude du Web sémantique que nous avons baptisée FPO (FPO : Fuzzy Probabilistic Ontology). (3) nous avons proposé une extension basée sur la logique floue du PR-OWL 2 (PRobabilistic ontology Web language) classique en tant que langage Web d'ontologie probabiliste floue en utilisant FzMEBN.

Afin de valider notre travail, nos contributions ont été examinées et leur efficacité a été confirmée pour la modélisation et le raisonnement sur des connaissances médicales qui présentent des connaissances floues liées à des connaissances probabilistes.

Mots clés :

Web sémantique, Connaissance incertaine, Ontologies probabilistes, logique floue.

Abstract

The use of ontologies has received recently considerable attention to represent the knowledge of a domain of application and to ensure semantic interoperability between heterogeneous systems, especially in the semantic Web. However, classical ontologies are ineffective when dealing with problems, in which the knowledge is uncertain.

This dissertation falls in a perspective of modeling and reasoning in rich uncertainty ontologies. Indeed, the ultimate goal is to develop a new approach that allows dealing with probabilistic knowledge and fuzzy knowledge at the same time, which are involved in the ontology of a field of application. To achieve this :

(1) We proposed a fuzzy extension to the MEBN networks that we named FzMEBN (Fz-MEBN : FuZzy Multi Entity Bayesian Network), in order to enrich their modeling and reasoning capabilities so as to cope with fuzzy knowledge and probabilistic knowledge simultaneously. (2) We introduced a new approach to deal with the uncertainty of the Semantic Web that we call Fuzzy Probabilistic Ontology (FPO). (3) We proposed a fuzzy extension of the classical PR-OWL 2 as a fuzzy probabilistic ontology Web language using FzMEBN.

In order to validate our work, our contributions have been tested and their effectiveness confirmed for modeling and reasoning on medical knowledge with fuzzy knowledge involved in probabilistic knowledge.

Key words :

Semantic Web, Uncertain Knowledge, Probabilistic Ontologies, Fuzzy Logic.

ملخص

تلقي استخدام الأنطولوجيات في السنوات الأخيرة اهتمامًا كبيرًا لنمذجة المعارف في مختلف المجالات التطبيقية أو لضمان التشغيل الدلالي بين الأنظمة الغير متجانسة ، خاصة في الويب الدلالي. ومع ذلك ، تظل الأنطولوجيات الكلاسيكية غير قادرة وغير فعالة لمعالجة المشاكل التي تتضمن معارف غير مؤكدة.

عملنا في هذه الأطروحة يندرج في اطار النمذجة والاستدلال على المعرفة غير المؤكدة في الأنطولوجيات الغنية بالمعارف الغير مؤكدة. في الواقع ، هدفنا الرئيسي هو تطوير نهج جديد يتيح التعامل مع المعرفة الاحتمالية والمعرفة الغامضة المتضمنة في انطولوجية مجال تطبيق معين.

لتحقيق أهدافنا :

- (1) اقترحنا امتدادًا لشبكات (MEBN (Multi Entity Bayesian Network استنادًا إلى المنطق الضبابي الذي أطلقنا عليه اسم (FzMEBN: Fuzzy Multi Entity Bayesian Network) ، من أجل إثراء قدراتها في مجال النمذجة والتفكير الاحتمالي.
- (2) اقترحنا مقاربة جديدة لمواجهة عدم اليقين في الويب الدلالي الذي نسميه FPO (FPO: Fuzzy Probabilistic Ontology).
- (3) اقترحنا امتدادًا لـ PR-OWL 2 (probabilistic ontology web language) كلغة احتمالية غامضة لتمثيل المعارف على الويب باستخدام FzMEBN

من أجل التحقق من صحة عملنا ، تمت فحص مساهماتنا وتأكيد فعاليتها للنمذجة والاستدلال على المعرفة الطبية ، التي تتضمن غالبًا معارف غامضة و معارف احتمالية في ان واحد

كلمات البحث :

الويب الدلالي ، المعرفة غير المؤكدة ، الأنطولوجيات الاحتمالية ، المنطق الضبابي

Table des matières

REMERCIEMENTS	iii
TABLE DES FIGURES	xiii
Introduction générale	1
Contexte de travail	1
Positionnement de la problématique	2
Objectifs et contributions de la thèse	4
Organisation de la thèse et guide de lecture	5
1 Incertitude dans le Web sémantique : contexte et challenges	9
1.1 Introduction	10
1.2 Notion d'ontologie	10
1.2.1 Définitions dans une perspective informatique	10
1.2.2 Composants d'ontologie	11
1.2.3 Logiques de description (DLs)	12
1.3 Architecture du Web sémantique	14
1.3.1 Couche des URIs (Uniform Resource Identifier)	15
1.3.2 Langage XML (Extensible Markup Language)	15
1.3.3 Langage RDF (Ressources Description Framework)	15
1.3.4 Langage RDF Schéma (RDFS)	17
1.3.5 Langage OWL(Ontology Web Language)	17
1.3.6 Couches règles, logique, preuve et confiance	17
1.4 Connaissance incertaine	18
1.4.1 Classification de l'incertitude comme une information imparfaite	18
1.4.2 Classification de l'incertitude par le W3C	19
1.4.3 Incertitude dans les composants de l'ontologie	22
1.5 Bilan sur le Web sémantique et l'incertitude	24
1.6 Conclusion	25

2	Approches formelles pour la gestion de l’incertain	26
2.1	Introduction	27
2.2	Logique floue	27
2.2.1	Variables linguistiques	28
2.2.2	Ensembles flous	28
2.2.3	Sémantique de la logique floue	31
2.3	Théorie des probabilités	31
2.3.1	Probabilités conditionnelles	31
2.3.2	Sémantique de la probabilité	32
2.4	Réseaux Bayésiens Standards	33
2.4.1	Construction d’un réseau bayésien	34
2.4.1.1	Apprentissage de la structure	34
2.4.1.2	Apprentissage des paramètres	35
2.4.2	Inférence dans les réseaux bayésiens	36
2.4.2.1	Inférence exacte	37
2.4.2.2	Inférence approchée	37
2.5	Réseaux Bayésiens Multi-Entités (MEBNs)	38
2.5.1	Modélisation des connaissances avec MEBN	38
2.5.2	Inférence dans MEBN	41
2.6	Réseaux Bayésiens Flous	44
2.6.1	Méthode de pondération	47
2.6.2	Distribution probabiliste floue	49
2.6.3	Méthode de l’évidence virtuelle	49
2.6.4	Bilan sur les méthodes d’inférence probabiliste floue	51
2.7	Discussion	51
2.8	Conclusion	53
3	Modélisation et raisonnement sur les connaissances incertaines	54
3.1	Introduction	54
3.2	Représentation des connaissances probabilistes	55
3.2.1	Logiques des descriptions probabilistes	56
3.2.2	Extensions probabilistes du langage RDF	56
3.2.3	Extensions probabilistes du langage OWL	57
3.2.3.1	Travaux de recherche de Ding et al., 2006	57
3.2.3.2	Travaux de recherche de Yang et Calmet, 2005	57
3.2.3.3	Travaux de recherche de Costa et al., 2005 ; Carvalho et al., 2017	58
3.2.3.4	Travaux de recherche de Carvalho et al., 2016	58
3.2.3.5	Travaux de recherche de Zheng et al., 2016	59

3.2.3.6	Travaux de recherche de Ishak et al., 2011	59
3.2.3.7	Travaux de recherche de Fenz ., 2012	59
3.2.3.8	Travaux de recherche de Mouenis et al., 2014	60
3.2.3.9	Travaux de recherche de Emna et al., 2016a; Emna et al., 2016b	60
3.2.3.10	Travaux de recherche de Mohammed et al., 2016	60
3.2.3.11	Travaux de recherche de Emna et al., 2018	61
3.3	Représentation des connaissances floues	61
3.3.1	Définitions des ontologies floues	61
3.3.2	Logiques de description floues	62
3.3.3	Extensions floues de OWL	62
3.3.4	Extension floue de RDF	63
3.3.5	Raisonnement sur des connaissances floues	63
3.4	Synthèse et comparaison	64
3.4.1	Discussion sur les ontologies probabilistes	66
3.4.2	Discussion sur les ontologies floues	69
3.5	Conclusion	72
4	Extension des Réseaux Bayésiens Multi-Entités basée sur la logique floue	73
4.1	Introduction	73
4.2	Motivation de la problématique	74
4.3	Réseaux Bayésiens Multi Entités Flous	76
4.3.1	Modélisation de l'incertitude via FzMEBN	76
4.3.2	Observations dans FzMEBN	78
4.3.2.1	Observation certaine	79
4.3.2.2	Observation floue	80
4.3.3	Inférence dans FzMEBN	84
4.3.3.1	Étape 1 : Générer un SSBN Flou	85
4.3.3.2	Étape 2 : Déterminer les vecteurs d'observations floues. . . .	86
4.3.3.3	Étape 3 : Ajouter les nœuds virtuels	88
4.3.3.4	Étape 4 : Lancer l'inférence probabiliste floue	89
4.3.4	Exemple Illustratif	89
4.4	Synthèse	92
4.5	Conclusion	94
5	Approche pour la gestion de la connaissance floue dans les ontologies probabilistes	95
5.1	Introduction	95
5.2	Contexte et motivation	96
5.3	Ontologie probabiliste floue	98

5.4	FuZzy-PRWOL2	99
5.4.1	Sémantique de FuZzy-PR-OWL2	100
5.4.2	Raisonnement dans l'ontologie probabiliste floue	108
5.4.3	Exemple Illustratif	110
5.5	Comparaison avec d'autres approches	115
5.6	Conclusion	118
6	Tests et validation : étude de cas dans le domaine médical	119
6.1	Introduction	119
6.2	Présentation du domaine d'application	120
6.3	Construction de l'ontologie probabiliste T2DRF	121
6.3.1	Étape 1 : Spécification des besoins	122
6.3.2	Étape 2 : Analyse et la conception	123
6.3.2.1	Entités	123
6.3.2.2	Règles	125
6.3.3	Étape 3 : Implémentation	126
6.4	Fuzzification de l'ontologie probabiliste	129
6.5	Tests	130
6.5.1	Vérification	131
6.5.2	Validation	133
6.6	Conclusion	135
	Conclusion Générale	136
	Billan	136
	Perspectives	137
A	Annexe A : Langage PR-OWL2	139
A.1	Variables aléatoires (RandomVariable)	139
A.1.1	Variables aléatoires booléennes	140
A.2	Arguments de mappage (MappingArgument)	142
A.3	Distribution des probabilités	142
A.4	Classe MEBN Expression	143
A.5	Classe Exemplar	144
A.6	Classe OrdinaryVariable	144
A.7	Arguments	144

Table des figures

1	Exemple d'un système d'orientation automatique des patients	3
2	Le synopsis de la thèse	8
1.1	Base de connaissances représentée en DLs	13
1.2	Architecture du Web sémantique	15
1.3	Un triple RDF.	16
1.4	Représentation graphique des triples RDF.	16
1.5	Les types d'imperfection	19
1.6	L'ontologie de haut niveau de l'incertitude définie par URW3-XG	20
1.7	Les types d'incertitude.	21
1.8	Taxonomie de l'incertitude liée à la connaissance ontologique	23
2.1	Illustration graphique d'un ensemble classique et d'un ensemble flou	29
2.2	Fonctions d'appartenance usuelles	30
2.3	Exemple d'un réseau bayésien	34
2.4	Exemple d'une MTheorie	40
2.5	Une Distribution Locale du noeud hasDiagnosis	41
2.6	Le SSBN généré pour Patient1	43
2.7	Exemple d'un RBF.	46
2.8	Représentation graphique des fonctions d'appartenance.	47
2.9	La nouvelle structure du RBF avec le nœud virtuel VHb.	50
3.1	L'ontologie de haut-niveau de OntoBayes	58
3.2	Tableau récapitulatif de la comparaison.	65
3.3	Architecture globale du système développé	68
3.4	Architecture du système d'aide à la décision d'identification des terroristes	69
3.5	Capture d'écran du questionnaire et les résultats temporaires de la décision	70
3.6	Le moteur d'exécution de requêtes sémantiques floues	71
3.7	Un formulaire de saisie de requête sémantique floue	71
4.1	Illustration d'un FzMFrag.	77
4.2	Illustration des observations sur un nœud résident R.	79
4.3	Représentation graphique de l'observation o_1	80

4.4	Illustration d'une observation floue.	81
4.5	Illustration d'une observation floue.	82
4.6	Représentation floue du nœud TA_{p1}	83
4.7	Le processus général de l'inférence dans FzMEBN.	85
4.8	Un FzMfrag pour l'anémie.	89
4.9	Les fonctions d'appartenance attachées au nœud "blood _ cell_count(p,t)" . . .	90
4.10	Les fonctions d'appartenance attachées au nœud "blood _ cell_count(p,t)" . . .	90
4.11	Illustration du processus d'inférence.	92
4.12	Différence entre MEBN et FzMEBN.	94
5.1	Positionnement de notre approche.	97
5.2	Le processus proposé pour la fuzzification des ontologies probabilistes.	100
5.3	L'ontologie de haut niveau de FuZzy-PR-OWL2.	102
5.4	Les éléments nécessaires pour définir une MThéorie.	103
5.5	Illustration graphique des liens entre les noeuds résidents flous et les noeuds résidents.	104
5.6	Le code OWL de la classe <i>FuzzyResidentNode</i>	104
5.7	Le code en OWL de la propriété d'objets « <i>IsFuzzyResidentInstanceOf</i> ».	104
5.8	Illustration graphique des liens entre les noeuds résidents flous et les noeuds inputs flous.	105
5.9	Le code en OWL de la classe <i>FuzzyInputNode</i>	105
5.10	Le code en OWL de la propriété d'objets « <i>IsFuzzyInputInstanceOf</i> ».	105
5.11	Les fonctions d'appartenance utilisées dans FuZzy-PR-OWL2.	106
5.12	Le code en OWL qui décrit la classe <i>Membershipfunction</i>	106
5.13	Le code en OWL qui décrit les classes <i>Left_Shoulde</i> , <i>Right_Shoulder</i> et <i>Linear</i> .	107
5.14	Le code en OWL qui décrit la classe <i>Triangular</i>	107
5.15	Le code en OWL qui décrit la classe <i>Traperzoidal</i>	108
5.16	Le code OWL de la propriété d'objet <i>isAttachedWith</i>	108
5.17	La représentation graphique de FzDMFrag-Anemia	110
5.18	Le code en OWL qui représente le noeud <i>hasHemoglobine</i>	111
5.19	Les états du noeud <i>hasHemoglobine</i>	111
5.20	La LPD du nœud <i>Domain_Res.hasHémoglobine</i>	111
5.21	Le code en OWL de l'instance <i>Fz.hasHemoglobine</i>	112
5.22	Le code en OWL qui décrit le lien entre l'instance <i>Fz.hasHemoglobine</i> et l'instance <i>Domain.Res.hasHemoglobine</i>	112
5.23	Les fonctions d'appartenance <i>Low</i> et <i>High</i>	112
5.24	Les propriétés des données de la fonction d'appartenance <i>Low</i>	113
5.25	Les propriétés des données de la fonction d'appartenance <i>High</i>	113

5.26	Le code en OWL qui décrit la relation entre l'instance Fz.hasHemoglobine et les fonctions d'appartenances <i>Low</i> et <i>High</i>	114
5.27	Observation sur l'entité <i>Patient_1</i>	114
5.28	Le SSBN flou pour le <i>Patient_1</i>	115
6.1	Cycle de la modélisation ontologique probabiliste avec la méthodologie UMP-ST	122
6.2	Processus de la modélisation d'incertitude via la méthodologie UMP-ST	122
6.3	Diagramme de classes pour l'ontologie T2DRF.	125
6.4	Le processus d'extraction des probabilités.	125
6.5	Les entités de l'ontologie T2DRF.	127
6.6	Créer une variable aléatoire en PR-OWL 2 à partir d'une propriété OWL.	127
6.7	LPD du noeud hasGlucose.	128
6.8	LPD du noeud PhysicalInactivity.	128
6.9	La MTheorie T2DRF.	130
6.10	Le SSBN flou généré pour la observations de la patiente_1.	132
6.11	Les résultats obtenus par une inférence probabiliste et une inférence probabiliste floue.	134
A.1	Les principaux éléments pour définir une variable aléatoire	140
A.2	La définition de la variable RV.estAmie.	141
A.3	La classe BooleanRandomVariable	141
A.4	la définition des arguments de la variable RV.estAmie	142
A.5	Les classes nécessaires pour la définition d'une distribution de probabilités	143
A.6	Les principaux éléments pour définir une MExpression	144

Liste des tableaux

2.1	Les observations stockées dans la base de connaissances.	43
2.2	Comparaison entre la théorie des probabilités et la logique floue.	45
2.3	La fuzzification de la variable Hb.	46
4.1	Les probabilités marginales du noeud TAp_1	82
4.2	La fuzzification du noeud TAp_1	83
4.3	La table des probabilités conditionnelles V1	91
4.4	La table des probabilités conditionnelles V2	91
5.1	Comparaison de notre approche avec les approches existantes.	117
6.1	Les attributs de l'entité patiente.	124
6.2	Les attributs de l'entité Diabetes_screening.	124
6.3	Les attributs de l'entité Activity.	124
6.4	Les relations entre les entités.	124
6.5	La fuzzification des noeuds flous.	129
6.6	Les informations de la patiente_1	131
6.7	Illustration des paramètres d'évaluation.	134

List of Algorithms

1	L'algorithme EM	36
2	Génération d'un SSBN -L'algorithme d'inférence de laskey-	42
3	Générer un SSBN Flou	86
4	Déterminer les vecteurs d'observations floues	87
5	Normalisation	87
6	Ajouter les nœuds virtuels	88
7	Remplissage	88

Nomenclature

RB	R éseau B ayésien
PR-OWL	P robabilistic O ntology W eb L anguage
W3C	W orld W ide W eb C onsortium
OBN	O riented O bject B yesian N etwork
ODM	O ntology D efinition M eta- M odel for OWL2
PODM	P robabilistic O ntology D efinition M eta- M odel for OWL2
OWL	O ntology W eb L anguage
URI	U niform R essource I dentifier
XML	E xtensible M arkup L anguage
RDF	R essources D escription F ramwork
UMP-ST	U ncertainty M odeling P rocess for S emantic T echnology
LPD	L ocal P robability D istribution
URW3-XG	U ncertainty R easoning for the W orld W ide W eb I ncubator G roup
T2DRF	T ype2 D iabetes R isk F actors

Introduction Générale

Contexte de travail

Avec l'émergence des technologies du Web sémantique, leur utilisation est devenue indispensable et une nécessité pour les systèmes basés Web intelligents et communicants. Cela grâce aux nombreux services qu'elles offrent afin de représenter et faire des raisonnements sur les connaissances de manière automatique. En effet, le Web sémantique est une extension du Web actuel dont l'objectif est de structurer et de représenter sémantiquement l'information dans le Web sous un format sophistiqué et traitable par les machines, permettant ainsi aux agents logiciels de l'utiliser de manière intelligente et efficace.

Donner une sémantique aux différentes ressources publiées dans le Web ne peut être effectué que par des annotations supplémentaires. Dans ce contexte, les ontologies ont été définies comme un standard par le W3C¹ (World Wide Web Consortium), pour décrire et annoter sémantiquement les différentes ressources qui circulent dans le Web ainsi que pour représenter les liens sémantiques entre ces dernières. Les ontologies sont considérées comme l'élément de base pour la construction du Web sémantique, elles assurent ainsi l'interopérabilité entre les différents systèmes dans des environnements hétérogènes.

Une ontologie en informatique est un module qui comporte un vocabulaire riche de concepts et de relations définis explicitement. Cela afin d'établir une conceptualisation formelle d'un domaine spécifique du monde réel et de la réalité perçue. Ce vocabulaire est doté d'un ensemble de faits afin d'apporter l'aide aux différents agents pour faire leurs activités en tenant compte de la sémantique d'entités du monde.

Aujourd'hui, avec l'explosion du Big Data ; les systèmes basés Web intelligents devront faire face à plusieurs défis afin de faire des traitements automatiques, fiables et efficaces. L'incertitude est l'un des problèmes les plus fréquents dans le traitement de l'information et plus la masse de données augmente plus l'incertitude augmente. Ainsi, plusieurs facteurs contribuent à la propagation de l'incertitude dans le Web sémantique, entre autres, une automatisation des systèmes d'extraction des connaissances à partir des sources de données qui ne sont pas toujours fiables, l'incomplétude des connaissances de l'agent sur l'état du système où il pratique ses activités, une mauvaise compréhension du domaine et des jugements effectués par des agents du Web sémantique, etc.

1. Est un organisme de standardisation des technologies liées au Web

Cependant, les technologies du Web sémantique ne sont pas capables de modéliser formellement la connaissance incertaine. Ceci est dû au fait que les ontologies classiques sont basées sur une logique booléenne et déterministe. Plusieurs recherches ont été amenées afin de remédier aux défauts de l'ontologie classique et combler ses insuffisances. Certaines de ces études sont basées sur la théorie des probabilités [1, 2, 3, 4, 5, 6, 7, 8], notamment les réseaux bayésiens pour faire face à l'incertitude, en particulier, la connaissance probabiliste dans les ontologies, donnant ainsi naissance à un nouveau modèle théorique nommé "*les ontologies probabilistes*". D'autres études [9, 10, 11], font appelle à la logique floue afin de traiter le flou qui se trouve au niveau des composants de l'ontologie et ils ont donné naissance à ce qu'on appelle "*les ontologies floues*".

Positionnement de la problématique

Actuellement, il n'existe aucun modèle formel pour gérer l'incertitude dans les ontologies standardisées par le W3C. Néanmoins, l'intégration de certaines théories a prouvé son efficacité pour la gestion de l'incertitude dans les ontologies. En effet, les ontologies probabilistes sont basées sur la sémantique des réseaux bayésiens (RBs) et leurs extensions pour faire face à la connaissance probabiliste attachée aux composants de l'ontologie. Les RBs permettent de modéliser des structures complexes par des graphes relationnels simplifiés afin de représenter la connaissance probabiliste sous un format bien lisible et exploitable par les systèmes basés Web intelligents et communicants.

Au cours de ces dernières années, une attention remarquable a été portée à la représentation et au raisonnement avec incertitude dans le Web sémantique. En effet, le sous groupe URW3-XG (URW3-XG : **U**ncertainty **R**easoning for the **W**orld **W**ide **W**eb **I**ncubator **G**roup)² du W3C a défini une ontologie de haut-niveau qui formalise les classes et les propriétés permettant de gérer des connaissances incertaines dans les ontologies [12]. Il est indiqué dans cette ontologie que l'incertitude peut apparaître sous plusieurs types tels que l'ambiguïté, le caractère aléatoire, le flou, l'incohérence et l'incomplétude. En outre, elle représente les modèles mathématiques convenables pour le traitement de l'incertitude dans le Web sémantique, y compris les réseaux bayésiens (RBs), la logique floue, les modèles hybrides, etc.

Par ailleurs, les auteurs dans [13, 14, 15] ont indiqué que dans les ontologies, l'incertitude de l'information peut être associée à deux catégories de concepts : la catégorie des concepts bien définis pouvant être traités par les approches probabilistes (connaissances probabilistes). La deuxième catégorie concernant les concepts mal définis, l'incertitude liée à cette classe est due à l'imprécision ou à l'approximation. Ainsi, la logique floue est le modèle approprié pour traiter le flou.

2. Un groupe créé par W3C qui s'occupe au traitement de l'incertitude qui affecte les connaissances véhiculées dans le Web sémantique.

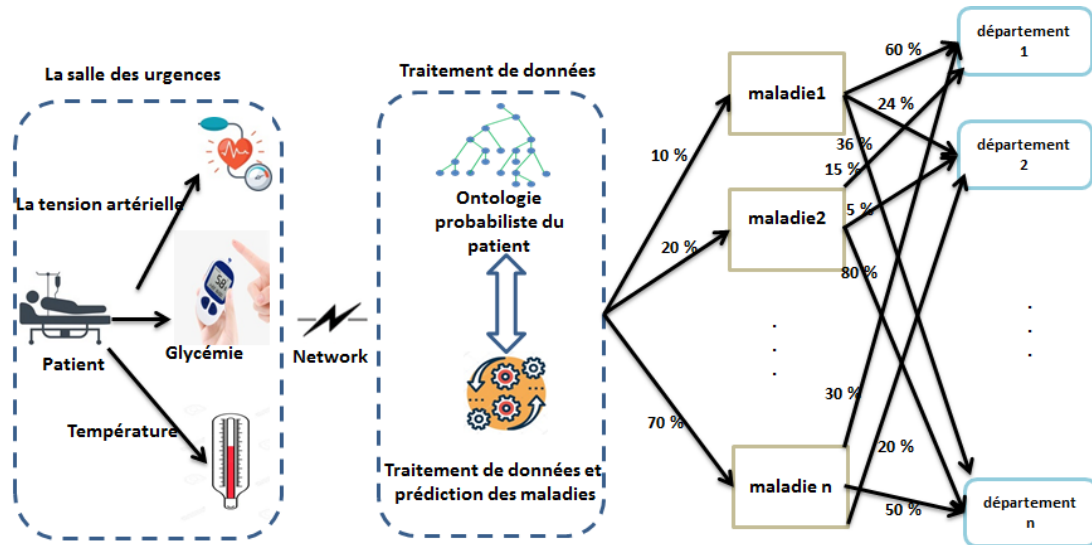


FIGURE 1 – Exemple d'un système d'orientation automatique des patients

Le langage PR-OWL 2 (PR-OWL 2 : **Probabilistic Ontology Web Language**) a été introduit en tant que solution candidate appartenant à l'approche des ontologies probabilistes pour représenter des connaissances incertaines fondées sur la théorie des probabilités. Il s'agit de l'un des travaux les plus développés dans ce domaine, basé sur une logique bayésienne du premier ordre appelée Réseau Bayésien Multi Entités (MEBN : **Multi Entity Bayesian Networks**). PR-OWL 2 a augmenté la sémantique du langage OWL par des annotations supplémentaires pour tenir compte de l'incertitude liée à l'incomplétude et au caractère aléatoire et permettre de la représenter et faire des raisonnements avec elle de manière formelle.

La puissance de PR-OWL 2 réside dans le fait qu'il est basé sur les MEBNs. En effet, les MEBNs permettent de représenter la connaissance probabiliste dans une représentation de base très expressive comme des fragments de réseaux bayésiens. Ces fragments permettent également de représenter les connaissances répétitives et les rendre réutilisables selon les situations observées. Cependant, il reste limité devant des problèmes qui comportent des connaissances floues liées aux termes linguistiques qui sont souvent utilisés par les humains. Par conséquent, le langage PR-OWL 2 hérite cet inconvénient et reste incapable de faire face au flou lié aux concepts, propriétés et instances de l'ontologie.

Notre étude de l'état de l'art dans le domaine de la gestion de l'incertitude et les ontologies a souligné que la connaissance probabiliste et la connaissance floue ont été traitées séparément dans les ontologies. Toutefois, dans les problèmes du monde réel, les deux types de connaissances apparaissent simultanément et très souvent étant liées et complémentaires. Où, la probabilité d'un événement dépend des variables et des valeurs floues [16]. Pour illustrer cela, supposons que nous voulons construire un système intelligent pour aider à l'affectation des patients dans les différents services d'un hôpital. Celle-ci est basée sur les différentes informations captées par les appareils (température, tension artérielle, glycémie, etc.) fixés dans la salle des urgences (voir Fig. 1.).

Les données du patient seront représentées sémantiquement dans un format sophistiqué dans une ontologie probabiliste, puis seront traitées pour prédire la maladie du patient. Sur la base de la maladie prédite, le système prédira le service le plus approprié pour traiter le patient. Dans le module de traitement de données, plusieurs règles de décision impliquent une connaissance probabiliste attachée à une connaissance vague. Par exemple :

- SI la tension artérielle d'un patient est **très élevée**, ALORS il est **probable** qu'il a une maladie cardiovasculaire.
- SI un patient est **très agé**, a des douleurs articulaires, des douleurs abdominales et une jaunisse, ALORS il est **probable** qu'il a un cancer du foie.
- SI un patient est **très fatigué**, maux de tête, a des courbatures, des douleurs, et une **forte** fièvre, ALORS il est **probable** qu'il a COVID-19.

Dans de telles situations, le mot **probable** fait référence à une connaissance probabiliste, tandis que les expressions **très élevée**, **très fatigué**, **très agé**, **forte** et **très proche** font référence à une connaissance floue.

Les applications du Web sémantique doivent modéliser et faire des raisonnements avec des ontologies riches en termes d'incertitude, où la connaissance probabiliste est attachée par une connaissance floue. Pour bien quantifier l'incertitude liée à une telle situation, nous devons développer des formalismes qui tiennent compte des deux types de connaissances simultanément dans les ontologies. Nous avons donc opté pour une démarche visant à enrichir le langage PR-OWL 2 dans le but de tenir compte du flou lié à la connaissance probabiliste. Dans cette optique, nous avons été amenés à formuler la question principale de la recherche dans cette thèse comme suit :

Comment modéliser et faire des raisonnements sur la connaissance floue trouvée dans des ontologies probabilistes formalisées en PR-OWL 2 ?

Telle est la problématique que nous allons tenter de traiter à travers cette thèse. Les sous-questions de recherches qui en découlent sont :

- ***QR1*** : *Comment améliorer les MEBNs afin de représenter les connaissances floues liées aux variables aléatoires de manière efficace ?*
- ***QR2*** : *Comment exploiter cette représentation pour inférer de nouvelles connaissances en tenant compte de la connaissance probabiliste et de la connaissance floue simultanément ?*
- ***QR3*** : *Comment intégrer et adopter cette version de MEBN pour représenter les connaissances floues encapsulées dans une ontologie probabiliste formalisée en PR-OWL 2 ?*

Objectifs et contributions de la thèse

L'objectif de cette thèse est de concevoir une nouvelle approche pour la modélisation et le raisonnement avec la connaissance incertaine dans les ontologies. En effet, l'objectif principal

de cette thèse est de traiter les connaissances probabilistes et les connaissances floues simultanément dans les ontologies.

Les contributions du travail sont les suivantes :

La première contribution est un état de l’art sur les approches de la représentation des connaissances incertaines, notamment dans les ontologies. Une classification de ces approches et des discussions ont été présentées dans notre thèse. Afin de répondre aux questions de recherches que nous avons précédemment développées dans la problématique, nous avons étudié l’existant et nous avons proposé deux autres contributions potentielles qui sont axées sur les deux grandes directives suivantes :

La deuxième contribution développée dans cette thèse consiste en une nouvelle extension des MEBNs afin d’améliorer leurs performances en termes de modélisation et de raisonnement. Notre extension vise à étendre les réseaux bayésiens multi-entités pour modéliser et faire des raisonnements probabilistes en tenant compte du flou au niveau des variables aléatoires. Cette contribution est publiée dans [17] (Riali et al., 2017) visant à adresser les deux premières questions de recherches. Elle a les caractéristiques suivantes :

- La modélisation de la connaissance sémantique floue dans les MEBNs. Ceci est fait en définissant un ensemble de constructeurs (nouveaux types de nœuds) pour distinguer entre la connaissance probabiliste précise et la connaissance probabiliste floue.
- La proposition d’un processus général pour faire l’inférence probabiliste en se basant sur des observations floues dans les MEBNs.

La troisième contribution porte sur une nouvelle approche qui exploite à la fois les modèles probabilistes et la logique floue pour mieux traiter la connaissance dans les ontologies riches en termes d’incertitude. Afin de concrétiser notre approche, nous avons proposé une extension du langage PR-OWL 2. Nous avons exploité notre première contribution pour faire face à l’incertitude liée à la connaissance ontologique floue dans les ontologies probabilistes formalisées en langage PR-OWL 2, en augmentant ses capacités de modélisation et de raisonnement. Cette approche est publiée dans [18] (Riali et al., 2019 a). Elle traite la troisième question de recherche et se caractérise par :

- La mise à jour de l’ontologie de haut-niveau de PR-OWL 2. Ceci en définissant un ensemble de classes et des propriétés pour permettre la représentation de la connaissance probabiliste floue.
- L’exploitation de l’inférence probabiliste en se basant sur des observations floues de Fz-MEBN afin d’augmenter les tâches de raisonnement du langage PR-OWL 2.

Organisation de la thèse et guide de lecture

Notre travail de thèse est structuré en six chapitres répartis en deux principaux volets. Le premier volet est composé de deux chapitres de l’état de l’art ; le deuxième volet contient quatre

chapitres qui présentent nos contributions. La figure 2 présente un guide de lecture qui résume le plan de la thèse en illustrant l'organisation des différents chapitres ainsi que les liens entre eux.

Partie I : État de l'art

Le Chapitre 1 : Incertitude dans le Web sémantique : contexte et challenges

Ce chapitre présente d'abord les ontologies comme un outil de représentation des connaissances en mettant l'accent sur les composants d'ontologies et les modèles de représentation des ontologies. Nous présentons ensuite le Web sémantique en décrivant ses couches. Dans la dernière partie de ce chapitre, une étude sur les classifications et les taxonomies d'incertitude est faite. Elle permet de mettre en évidence les notions de bases sur l'incertitude qui ont un impact direct sur la connaissance ontologique.

Le Chapitre 2 : Approches formelles pour la gestion de l'incertain

Ce chapitre vise à donner un aperçu sur les modèles de la gestion de l'incertitude dans le contexte du Web sémantique. Nous exposons dans la première section de manière brève la logique floue en mettant l'accent sur les fondements théoriques de cette logique. Par la suite, nous décrivons la théorie des probabilités ainsi que la sémantique de cette théorie, dans cette partie nous focalisons notre attention sur les réseaux bayésiens, notamment leurs méthodes de construction à savoir les méthodes d'apprentissage de structures des réseaux bayésiens et des paramètres. Les techniques d'inférences bayésiennes sont ainsi décrites.

Enfin, nous présentons quelques extensions des RBs standards en commençons par les réseaux bayésiens multi entités, une extension qui vise à améliorer l'expressivité et la flexibilité des RBs. Dans la dernière partie, nous présentons les réseaux bayésiens flous. L'accent est mis sur les méthodes d'incorporation d'observations floues dans les probabilités bayésiennes

Partie II : Contributions

Le Chapitre 3 : Modélisation et raisonnement sur les connaissances incertaines

Ce chapitre expose un état de l'art sur les extensions probabilistes et les extensions floues aux ontologies, de la logique des descriptions et du langage RDF. Nous présentons également dans ce chapitre quelques raisonneurs flous. En outre, dans ce chapitre une classification de ces extensions selon plusieurs critères d'évaluations est présentée. Enfin nous exposons une comparaison et une discussion entre les différentes solutions.

Le Chapitre 4 : Extension des Réseaux Bayésiens Multi-Entités basée sur la logique floue

Ce chapitre présente des éléments de réponses aux deux premières questions de recherches

évoquées dans la problématique. Dans ce chapitre, nous allons exposer notre première contribution en présentant une extension floue des réseaux bayésiens multi-entités, en mettant l'accent sur les améliorations apportées par notre extension avec une illustration du processus de la modélisation et du raisonnement probabiliste flou.

Le Chapitre 5 : Une approche pour la gestion de la connaissance floue dans les ontologies probabilistes

Ce chapitre présente une nouvelle approche pour le traitement de la connaissance incertaine au niveau ontologique qui s'oriente vers le traitement de la connaissance probabiliste et la connaissance floue ensemble. Une extension du langage PR-OWL 2 est détaillée, basée principalement sur l'intégration de l'extension floue pour les réseaux bayésiens multi-entités présentée dans le chapitre précédent. Nous présentons également une comparaison entre notre solution proposée et quelques travaux de la littérature.

Le Chapitre 6 : Tests et validation : étude de cas dans le Domaine médical

Ce chapitre est consacré à la présentation d'une étude de cas dans le domaine médical, spécifiquement la maladie du Diabète. Pour montrer l'efficacité de la solution proposée en mettant l'accent sur les capacités de la modélisation de l'incertitude offertes par notre solution. Nous exposons aussi une étude empirique afin de valider et montrer les capacités de raisonnement de notre solution.

Notre thèse se termine par une conclusion générale qui rappelle les principaux objectifs et présente les perspectives et les futures orientations pour améliorer nos contributions.

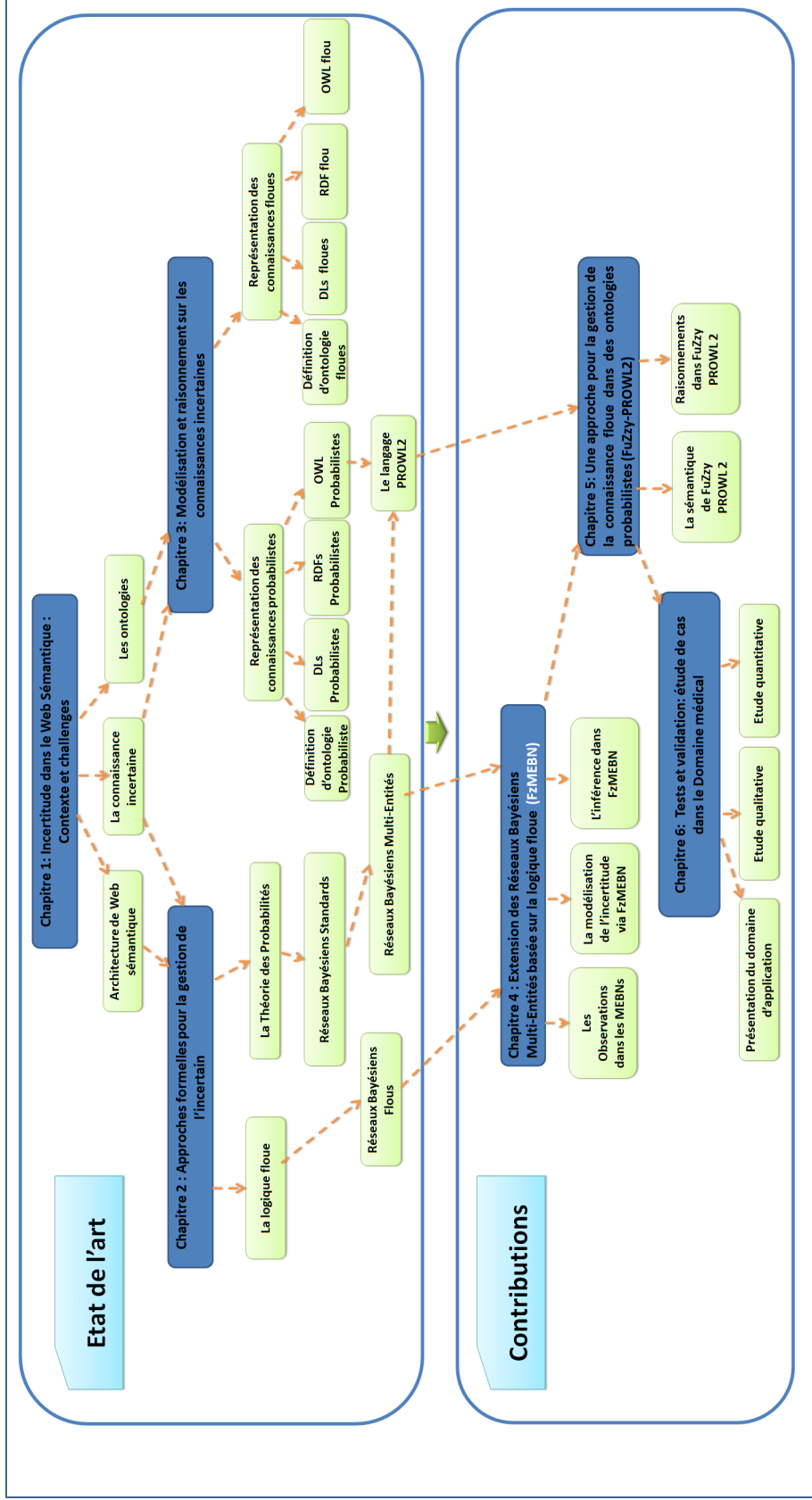


FIGURE 2 – Le synopsis de la thèse

Chapitre 1

Incertitude dans le Web sémantique : contexte et challenges

“Nous souhaitons la vérité, et ne trouvons en nous qu’incertitude.”

— —Blaise Pascal

Sommaire

1.1	Introduction	10
1.2	Notion d’ontologie	10
1.2.1	Définitions dans une perspective informatique	10
1.2.2	Composants d’ontologie	11
1.2.3	Logiques de description (DLs)	12
1.3	Architecture du Web sémantique	14
1.3.1	Couche des URIs (Uniform Resource Identifier)	15
1.3.2	Langage XML (Extensible Markup Language)	15
1.3.3	Langage RDF (Ressources Description Framework)	15
1.3.4	Langage RDF Schéma (RDFS)	17
1.3.5	Langage OWL (Ontology Web Language)	17
1.3.6	Couches règles, logique, preuve et confiance	17
1.4	Connaissance incertaine	18
1.4.1	Classification de l’incertitude comme une information imparfaite	18
1.4.2	Classification de l’incertitude par le W3C	19
1.4.3	Incertaine dans les composants de l’ontologie	22
1.5	Bilan sur le Web sémantique et l’incertitude	24
1.6	Conclusion	25

1.1 Introduction

Le projet du Web sémantique a été fondé en 2001 par TIM BERNERS LEE, qui est le créateur du World Wide Web. Il est maintenant en cours de développement sous le soutien du W3C. Le Web sémantique vise à enrichir le Web actuel par des méta-données afin de décrire et d'exploiter la sémantique de différentes ressources du Web de manière automatique. Son objectif est de rendre le contenu sémantique des ressources du Web interprétable et traitable, non seulement par les êtres humains mais aussi par les machines [19]. Ainsi, l'un des grands défis du Web sémantique est de faire face à la connaissance incertaine, il s'agit de modéliser les imperfections de notre monde pour que les applications du Web sémantique fassent des traitements proches aux raisonnements humains en présence de l'incertitude.

Ce chapitre a pour objectif de présenter en premier temps les ontologies, constituant les entités de base du Web sémantique. Ensuite, nous poursuivons par explorer le Web sémantique en détaillant ses technologies qui sont présentées dans la pile du Web sémantique. Quelques classifications et taxonomies d'incertitude sont exposées pour familiariser avec le concept d'incertitude. Finalement nous présentons un bilan qui résume les points clés qui sont abordés au cours de ce chapitre.

1.2 Notion d'ontologie

Dans cette section, nous nous focalisons sur les différents concepts relatifs aux ontologies qui constituent un élément primordial pour la construction du Web sémantique. En effet, il est difficile de définir le terme "*ontologie*" de manière précise et consensuelle et ceci dépend de son exploitation dans les différents domaines d'applications. Ainsi, nous nous intéressons dans notre étude aux ontologies dans une perspective informatique en présentant plusieurs définitions des ontologies, les composants constituant les ontologies et les modèles de représentation d'ontologies.

1.2.1 Définitions dans une perspective informatique

Nous commençons par la première définition proposée par NECHES [20] : une ontologie définit les termes et les relations de base du vocabulaire d'un domaine ainsi que les règles qui permettent de combiner les termes et les relations afin de pouvoir étendre le vocabulaire.

Dans [21], GRUBER a défini l'ontologie comme suit : « *Une ontologie est une spécification explicite d'une conceptualisation* ».

D'après cette définition, on constate qu'une ontologie est une spécification explicite d'une conceptualisation d'un domaine de connaissance.

Par ailleurs, dans [22], GUARINO et GIARETTA ont tenté de donner des définitions de l'ontologie. Ils ont proposé sept définitions, entre autres : l'ontologie comme une discipline philosophique, l'ontologie comme un système conceptuel informel et comme une spécification

d'une conceptualisation. Puis l'ontologie comme la représentation d'un système conceptuel via une théorie logique et son vocabulaire. En outre, ils ont proposé également d'utiliser le terme « Ontologie » avec un O majuscule pour désigner l'ontologie dans son sens philosophique et « ontologie » avec un o minuscule pour désigner l'ontologie de l'informatique.

Dans [23], BORST a développé la définition de GRUBER en énonçant que : «*Une ontologie est une spécification formelle d'une conceptualisation partagée* »

La définition la plus citée dans la littérature a été proposée par STUDER ET AL, dans [24], les auteurs ont combiné et développé la définition d'ontologie de BORST et GRUBER, en disant : «*une ontologie est une spécification formelle et explicite d'une conceptualisation partagée* »

- **Conceptualisation** : il s'agit d'élaborer un modèle conceptuel visant à identifier les concepts appropriés pour conceptualiser un phénomène particulier du monde réel.
- **Explicite** : les concepts utilisés et leurs propriétés doivent être déclarés et définis explicitement.
- **Formelle** : l'ontologie doit être codée et traduite dans un langage interprétable et traitable par une machine (la machine doit pouvoir traiter la sémantique de l'information représentée).
- **Partagée** : l'ontologie doit capturer la connaissance consensuelle d'un domaine d'application. C'est-à-dire, elle n'est pas réservée à quelques individus mais acceptée et partagée par un groupe ou une communauté.

1.2.2 Composants d'ontologie

Selon GOMEZ-PEREZ [25], la connaissance dans les ontologies est formalisée à l'aide de cinq types de composants « concepts », « relations », « fonctions », « axiomes » et « instances ».

- **Les concepts** : sont les termes ou classes de l'ontologie, ils peuvent être abstraits ou concrets, élémentaires ou composites, réels ou fictifs. Un concept peut être une description d'une tâche, fonction, etc. Les concepts sont caractérisés par une **Extension** qui est l'ensemble des objets (instances) manipulés à travers ce concept, et une **Intension** qui est l'ensemble des propriétés spécifiant la sémantique du concept.
- **Les relations** : pour représenter un type d'interaction entre les concepts du domaine. Les relations sont représentées formellement comme suit :

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

- **Les fonctions** : un cas particulier d'une relation, où le même concept est défini en fonction des n-1 éléments précédents, la représentation formelle d'une fonction est la suivante :

$$F : C_1 \times C_2 \times \dots \times C_{n-1} \implies C_n$$

- **Les axiomes** : représente les assertions qui sont toujours vraies, les axiomes sont utilisés pour modéliser les phrases qui sont toujours vraies. En effet, on les utilise également pour la construction de la base des faits dans l'ontologie (afin d'alimenter l'ontologie).
- **Les instances** : pour représenter les individus (objets) d'un domaine.

Les ontologies ont été proposées comme une solution pour la représentation et le raisonnement avec la connaissance dans divers domaines, tels que l'intelligence artificielle, la science cognitive et notamment dans le Web sémantique. En effet, le Web sémantique s'appuie sur les ontologies dans une couche supérieure afin d'exploiter ses avantages en matière de représentation et de raisonnement sur les connaissances. Dans le cadre de la formalisation des ontologies, plusieurs modèles et structures ont été distingués. Dans la prochaine section, nous décrivons les logiques de description (DLs)

1.2.3 Logiques de description (DLs)

Les DLs sont une famille de formalismes pour la représentation des connaissances basées sur la logique mathématique. Une base de connaissances représentée avec DLs se divise en deux parties comme il est illustré dans la figure 1.1. La première partie correspond à des informations terminologiques TBOX (termes de domaine) et l'autre à des assertions ABOX [26].

- **La partie TBOX « Terminological-BOX »** : cette partie contient les termes du domaine (noms de concepts). Elle inclut les concepts de domaines correspondant à des prédicats unaires et les rôles (relations binaires entre les concepts ou les individus). Elle représente la partie schéma de l'ontologie, permettant la définition des propriétés et des opérations telles que l'union, intersection, négation, les quantificateurs et les contraintes sur les cardinalités d'ensembles.
- **La partie ABOX « Assertionnel-BOX »** : cette partie sert à la représentation des constantes, des instances des concepts et des rôles sur ces constantes. La partie ABOX contient des assertions sur l'appartenance et d'autres sur les rôles relatifs à des individus.

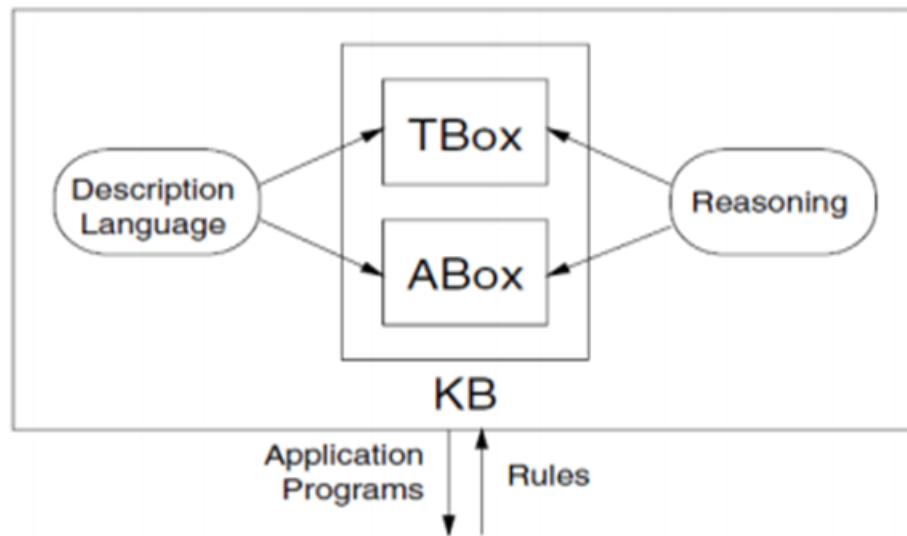


FIGURE 1.1 – Base de connaissances représentée en DLs [26].

Les composants ABOX et TBOX permettent de définir une base de connaissances exploitable et faire des inférences pour la déduction de nouvelles connaissances. En effet, il existe plusieurs types des logiques de description, la différence entre ces types réside dans leurs niveaux d'expressivité et les constructeurs qu'elles supportent. L'une des première DL est le langage FL^- (Frame Logic) qui permet de représenter des quantificateurs existentiels et des quantificateurs universels. La logique AL (Attributif langage) étend le langage FL^- en permettant la représentation de la négation des concepts atomiques. Elle est considérée comme la logique de base des autres DLs.

Les autres types sont des combinaisons entre les différents constructeurs, à titre d'exemples, la logique ALC (Attributif Langage with Complements) étend la logique AL en y ajoutant la négation complète C . Ainsi, le langage du Web sémantique OWL est basé sur les DLs, spécifiquement, $SHIF$, $SHIF(D)$ pour le OWL_LITE , $SHOIN$, $SHOIN(D)$ pour le OWL_DL et enfin $SROIQ(D)$ pour le $OWL2$. En effet, OWL_FULL ne correspond à aucune DL et son raisonnement est indécidable [27].

Les DLs offrent plusieurs services d'inférences qui permettent d'inférer des connaissances implicites à partir des connaissances explicitement représentées dans un DL. Parmi ces services, nous citons :

- **La subsomption des concepts** vise à tester si un concept C_1 est plus général qu'un concept C_2 (cette relation est notée $C_2 \sqsubseteq C_1$). Autrement dit, cette relation est vérifiée si et seulement si l'ensemble d'individus du concept C_1 contient l'ensemble d'individus du concept.
- **La satisfiabilité des concepts**, un concept est satisfiable si et seulement si il possède au moins un individu.
- **Le test d'instanciation** vise à vérifier si un individu I est une instance d'un concept C .

- **Équivalence de classes**, permet de vérifier si la base de connaissances implique que deux classes C_1 et C_2 sont équivalentes et on note $C_1 \equiv C_2$.
- **Récupération d'instances** pour récupérer tous les individus d'une classe donnée.

1.3 Architecture du Web sémantique

Le Web sémantique constitue un espace dans lequel les humains et les machines peuvent travailler en collaboration selon une base sémantique. En effet, la première définition qui se présente lorsque nous parlons du Web sémantique, c'est celle d'un Web intelligent et compréhensible par les machines où elles font des traitements automatisés sur les données.

Selon TIM BERNERS LEE et al « Le Web sémantique est une extension du Web actuel dans lequel le sens des informations est bien défini, ce qui permet aux machines et aux humains de travailler en coopération » [19].

Le Web sémantique (nommé aussi Web 3.0 ou Web de données) est une version intelligente du Web actuel qui vise à donner une sémantique aux informations et représenter les liens sémantiques dans le Web pour que les agents logiciels puissent travailler en collaboration. Il s'agit donc d'une nouvelle extension du Web courant qui s'appuie sur une gamme de technologies et techniques dont l'objectif est de rendre le contenu des ressources du Web (images, pages Web, etc.) accessibles, utilisables et traitables par des agents logiciels et machines. Dans cette perspective, les méta-données formelles ont joué un rôle central pour décrire formellement les connaissances dans le web sémantique.

Les méta-données sont alors des données structurées servant à décrire d'autres données. Elles sont généralement utilisées quand on parle sur l'information descriptive à propos d'une ressource du Web. Parmi les objectifs des méta-données est l'identification des ressources afin de répondre à un besoin d'utilisation de l'information et la traçabilité des caractéristiques d'une ressource. A titre d'exemples :

- Le catalogue d'une bibliothèque est une méta-donnée qui décrit toutes les ressources de la bibliothèque telles que, des livres avec leurs titres, des auteurs, etc.
- Le menu dans un restaurant est une méta-donnée qui donne une vue globale sur tous les services offerts par un restaurant à savoir des plats, des jus, etc.

La proposition du W3C s'appuie sur gamme de technologies présentées en couches pour l'organisation du Web sémantique, la pile du Web sémantique est montrée dans la figure 1.2 dont seulement les couches basses sont à l'heure actuelle stabilisées. Dans ce qui suit, nous donnons un aperçu sur les technologies utilisées dans chaque couche.

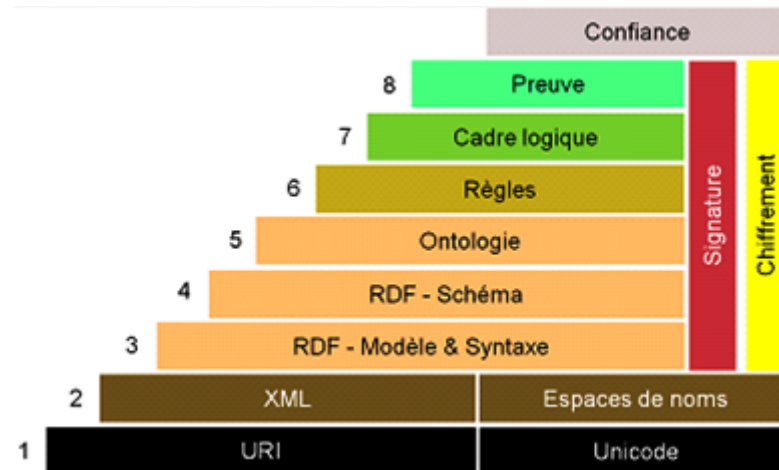


FIGURE 1.2 – Architecture du Web sémantique [28].

1.3.1 Couche des URIs (Uniform Resource Identifier)

C'est la couche la plus basse dans la pile du Web sémantique, son rôle principal est l'identification des ressources dans le Web sémantique afin d'assurer l'interopérabilité syntaxique. En effet, la notion d'URI (Uniform Resource Identifier, identifiant uniforme de ressource) fournit un adressage standard universel permettant d'identifier des ressources. Ainsi, un URI est défini par une courte chaîne qui sert à l'identification de la ressource (physique ou abstraite) sur un réseau, elle est codée en ASCII et standardisée par des IRIs (International Resource Identifier) avec des caractères internationaux Unicode (Unicode : est un encodage textuel universel pour échanger des symboles).

1.3.2 Langage XML (Extensible Markup Language)

XML est un langage à balises spécialisés et conçu pour décrire les données d'une manière simple permettant l'échange et le partage de ces données facilement dans des systèmes hétérogènes, en utilisant un langage de marquage basé sur le texte [29]. Ainsi, chaque fichier XML est associé avec des Définitions des Types de Documents (DTDs) qui décrivent et spécifient la structure des documents XML. Cependant, les DTDs manquent d'expressivité pour décrire des structures de données plus complexes.

Le langage XML Schéma (XMLS) offre des mécanismes de haut niveau pour la représentation de données. La connaissance en XMLS est représentée en XML 1.0 en utilisant les noms d'espaces. Il est construit essentiellement pour étendre considérablement les fonctionnalités trouvées dans les définitions XML de type 1.0 du document (DTD) [30].

1.3.3 Langage RDF (Resources Description Framework)

Le langage RDF est dédié à la représentation des méta-données. Il est défini par le W3C afin de décrire des ressources et les relations entre elles indépendamment des applications. Les

connaissances en RDF sont représentées par un ensemble de triples de la forme sujet, prédicat, objet, comme il est montré dans la figure 1.3.

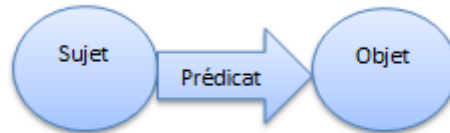


FIGURE 1.3 – Un triple RDF.

où :

- Le sujet représente la ressource à décrire.
- Le prédicat représente la relation entre un sujet et un objet, il permet de mettre en relation un sujet et une propriété.
- L'objet est la valeur de ce prédicat. Il peut être soit une ressource ou bien une donnée (un littéral).

En effet, un ensemble de tels triples constitue un graphe qui représente les connaissances de manière facile à interpréter. Néanmoins, le RDF ne donne aucune signification spécifique pour représenter un vocabulaire expressif comme « sous classe de », ou « le type ».

L'exemple ci-après montre un exemple des triples RDF représentés d'une manière formelle.

*<Mohammed> <est une> <personne>. <Mohammed> <est ami de> <Amine>.
<Mohammed> <est né en> <15 décembre 1991>.
<Mohammed> <étudie à> <Usd Blida 1>. <Usd Blida 1> <est située à> <Blida>.*

La représentation graphique de ce code est illustrée dans la figure 1.4.

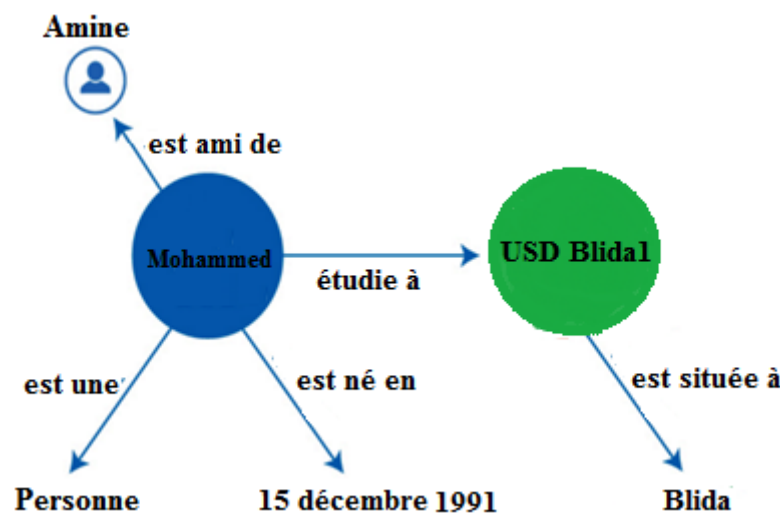


FIGURE 1.4 – Représentation graphique des triples RDF.

1.3.4 Langage RDF Schéma (RDFS)

Afin de remédier aux lacunes du langage RDF, le langage **RDF Schéma (RDFS)** [31, 32] a été proposé comme une extension du RDF. Il est doté d'un vocabulaire très riche de termes et de relations entre ces termes à savoir la notion des classes, les propriétés, les sous propriétés, etc. En effet, le RDFS est reconnu comme un langage d'ontologie caractérisé par :

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

1.3.5 Langage OWL(Ontology Web Language)

Malgré les nombreux services de modélisation offerts par RDFS. Il reste très faible lorsqu'on veut exprimer certaines caractéristiques et propriétés, tels que l'union, l'intersection et aussi la restriction d'une valeur ou d'une cardinalité. Par conséquent, le langage OWL est proposé pour permettre la représentation de ces caractéristiques et propriétés dans les ontologies. Le langage OWL est dédié pour la représentation des ontologies dans le Web sémantique [33]. Il étend le langage RDFS en permettant de représenter plusieurs notions qui n'existent pas dans RDFS, comme les outils de comparaison entre les propriétés et les classes telles que, l'équivalence, le contraire, la symétrie, la disjonction, la transitivité, etc. Par ailleurs, il existe trois sous-langages avec des niveaux de complexité ascendante. Ainsi, les trois sous-langages sont caractérisés par un niveau d'expressivité différent [34].

- *OWL_LITE*, le plus simple sous langage. Il possède un niveau d'expressivité minimale. Il est conçu principalement pour la représentation des thésaurus, des taxonomies ou des hiérarchies de classes avec des contraintes plus ou moins simples. Son utilisation assure la complétude des calculs.
- *OWL_DL* (Ontology Web Language-Description Logic) , le sous-langage *OWL_DL* plus compliqué que *OWL_LITE*. Il est caractérisé par une expressivité maximale, mais avec une complétude de calcul et en garantissant la calculabilité. C'est-à-dire, un temps fini de calcul et les toutes conclusions sont garantis d'être calculables.
- *OWL_FULL*, le sous-langage *OWL_FULL* est le sous-langage le plus compliqué, destiné aux utilisateurs qui ont besoin d'une expressivité maximale du vocabulaire. Ce sous-langage ne garantit pas la calculabilité et la décidabilité. C'est-à-dire, les conclusions ne sont pas toutes garanties d'être calculables.

1.3.6 Couches règles, logique, preuve et confiance

Les couches règles, logique, preuve et confiance ne correspondent pas à des technologies standardisées. La couche logique repose sur des langages permettant d'exprimer des règles de

raisonnement pour construire un système d'inférence à base de règles. Quant à la couche preuve, son rôle est la génération de contrôles pour la vérification des déclarations effectuées dans le Web sémantique. Ainsi, la couche confiance vise à garantir et donner confiance sur les résultats obtenus à travers l'explication et les justifications des résultats de preuve. Elle garantit également la fiabilité et la source des informations utilisées à savoir la signature électronique en s'appuyant sur les techniques de la cryptographie.

1.4 Connaissance incertaine

Il est difficile d'attribuer à la notion d'incertitude une définition unique et consensuelle puisque celle-ci a été définie dans des contextes différents. Ainsi, parmi les définitions proposées dans la littérature, nous citons :

- Selon ZADEH [35], "*l'incertitude est une propriété inhérente qui caractérise l'information*". En effet, *l'incertitude est considérée comme un attribut de l'information. Elle est inévitable dans la plupart des raisonnements et dans de nombreux domaines du monde réel et surtout dans les domaines de jugement, tels que la médecine, les affaires, le droit, l'éducation, etc.*
- *L'incertitude est l'écart entre l'information requise pour exécuter une tâche et l'information disponible* [36],
- *L'incertitude est un sentiment de doute, dans le contexte de l'action, qui bloque ou retarde l'action* [37],
- *L'incertitude apparaît quand un individu est incapable à prédire quelque chose* [38],
- *L'incertitude est de ne pas pouvoir répondre à une question dans un contexte donné* [39].

Le terme "*Incertain*" correspond à un concept très large et sa signification dépend souvent du contexte scientifique dans lequel elle a été employée.

Dans les sections suivantes, nous présentons plusieurs typologies d'incertitude selon différents contextes. nous commençons par l'incertitude comme un type d'imperfection d'informations (contexte de l'IA). Nous exposons ensuite l'incertitude comme un challenge défini par le W3C, et finalement nous mettons en évidence l'incertitude qui affecte la connaissance ontologique.

1.4.1 Classification de l'incertitude comme une information imparfaite

La classification présentée dans cette section est la plus répandue en IA. L'incertitude est considérée comme type particulier d'imperfection qui affecte les informations, elle est placée dans un niveau bas dans la hiérarchie de cette classification.

Comme il est montré dans la figure 1.5, la notion d'information imparfaite recouvre plusieurs types à savoir [40] l'inconsistance, l'incomplétude, l'imprécision et l'incertitude.

En effet, l'incertitude est le type d'imperfection le plus traité et discuté dans la littérature. Elle peut être considérée comme une propriété subjective ou objective [41]. L'incertitude liée au caractère aléatoire est une propriété objective, quant à l'incertitude subjective, elle est liée aux opinions des agents sur la véracité de données.

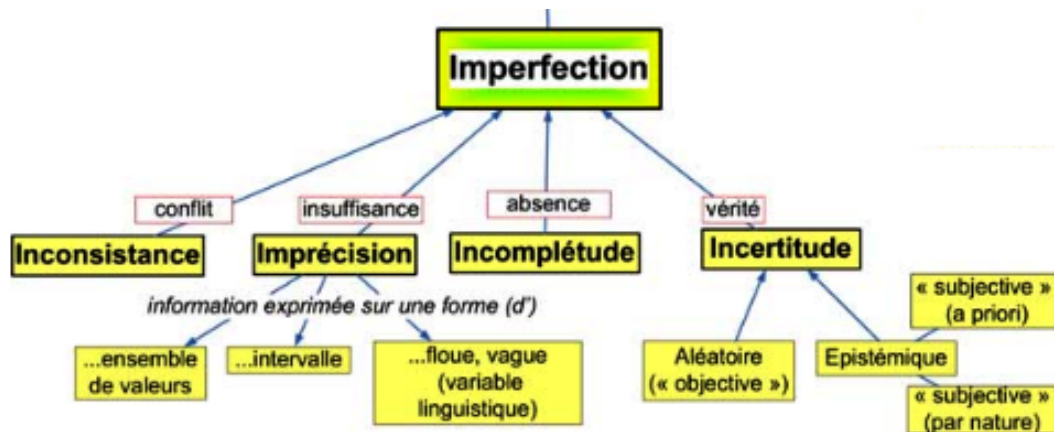


FIGURE 1.5 – Les types d'imperfection [40].

1.4.2 Classification de l'incertitude par le W3C

Le Web sémantique en tant qu'un Web de données traitables automatiquement par des machines, mondial et ouvert nécessite des formalismes capables de représenter et de raisonner en présence d'incertitude pour être à la hauteur de ses promesses. Pour cette raison, les membres du groupe URW3-XG [12] ont fait une étape très importante dans la spécification de l'incertitude qui affecte le raisonnement dans le World Wide Web ainsi que les questions que nous devons prendre en considération lors de la conception d'une représentation standard qui tienne compte de l'incertitude. En effet, ils ont proposés une ontologie de haut niveau, illustrée dans la figure 1.6, pour mieux définir les défis de la représentation et du raisonnement avec des connaissances incertaines sur le World Wide Web.

Dans la classification présentée dans cette section, le terme «*Incertain*» désigne toutes les formes de la connaissance imparfaite [12], entre autres, nous pouvons citer : l'incomplétude, l'imprécision, l'ambiguïté. Dans ce qui suit nous donnons un aperçu sur cette ontologie.

1. **Les phrases (sentences)**, ce sont des expressions dans un langage logique formel qui peuvent être associées par des valeurs de vérités (axiomes, formules, etc).
2. **Le Monde (world)** représente le monde auquel appartient la phrase déclarée.
3. **les Agents** représentent ceux qui ont fait les déclarations des phrases, un agent peut être une machine ou un être humain.
4. **L'incertitude (Uncertainty)**, une déclaration sur le sujet de l'incertitude associée à la phrase.

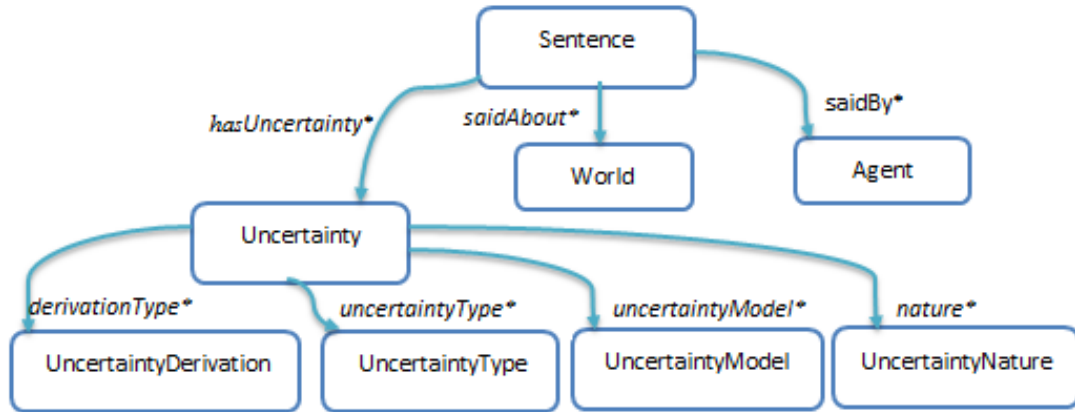


FIGURE 1.6 – L'ontologie de haut niveau de l'incertitude définie par URW3-XG [12]

Dans ce qui suit, nous présentons plusieurs variantes à savoir la nature de l'incertitude, la dérivation de l'incertitude, les types d'incertitude et les modèles utilisés pour traiter l'incertitude

1. **La nature de l'incertitude (Uncertainty Nature)**, cette classe capture les informations sur la nature d'incertitude à savoir si l'incertitude est inhérente à la connaissance exprimée par la phrase ou c'est le résultat d'un manque de connaissance de l'agent.
 - * **Stochastique**, l'incertitude stochastique caractérise l'aspect aléatoire de l'information. L'incertitude stochastique, nommée aussi variabilité qui est une propriété inhérente du monde (le hasard). En effet, elle concerne les phénomènes et les situations dont l'occurrence est considérée comme aléatoire ou stochastique. C'est-à-dire, des phénomènes qui ne donnent pas souvent les mêmes résultats même sous des conditions similaires. Par ailleurs, on parle de l'incertitude qui est due à l'aléatoire lorsqu'on sait que l'événement E ne se passe pas toujours de la même manière [42].
 - * **Épistémique**, l'incertitude épistémique est due à l'incomplétude de l'information et le manque de connaissance de l'agent dans son environnement. Cette incertitude est nommée aussi ignorance et elle est présente à cause d'insuffisance de connaissances. On parle de l'incertitude épistémique lorsqu'on ne sait pas exactement comment l'événement E se passe [30]. Contrairement à l'incertitude stochastique, ce type d'incertitude est réductible par l'acquisition de nouvelles informations [43, 44], et on peut réduire le degré d'incertitude par des observations supplémentaires sur la phrase incertaine.
2. **La dérivation de l'incertitude (Uncertainty Derivation)**, la dérivation de l'incertitude peut être objective ou subjective (jugement subjectif, supposition).
 - * **Objective**, la dérivation de l'incertitude est considérée comme objectif lorsque les phrases incertaines sont évaluées via un outil reproductible.

- * **Subjective**, on considère la dérivation d'incertitude subjective, lorsque les phrases incertaines sont évaluées par un jugement subjectif. Par exemple, l'estimation d'un agent (une supposition, un deviné).

3. **Les types d'incertitude (Uncertainty Type)**, l'incertitude peut se présenter sous plusieurs types comme il est montré dans la figure 1.7.

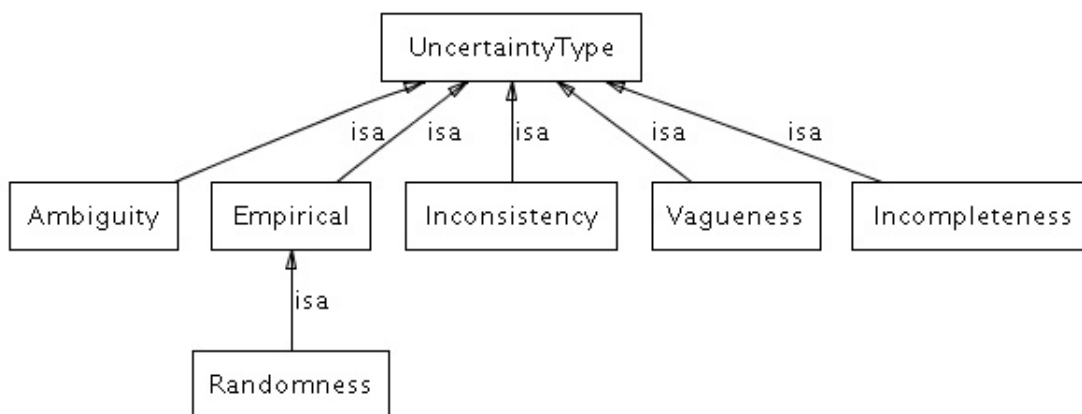


FIGURE 1.7 – Les types d'incertitude [12].

- * **L'ambiguïté**, on parle de l'ambiguïté dans une phrase incertaine lorsqu'on a des termes qui sont interprétables de plusieurs manières. En effet, la sémantique des termes dans une phrase n'est pas clairement spécifiée et donc il est difficile à déterminer si la phrase est satisfaite ou pas ce qui va engendrer une situation d'incertitude. Par exemple :
 - « **Amina sent la rose** ». Deux différentes interprétations sont possibles pour cette phrase. Soit Amina est en train de respirer l'odeur d'une rose, soit le parfum que Amina porte sent la rose.
- * **L'aléatoire**, on parle de l'aléatoire ou le hasard lorsqu'on ne connaît pas une loi générale gouvernante qui explique la réalisation des phénomènes.
- * **L'imprécision**, l'imprécision est relative au contenu des phrases à caractère vague et approximatif. En effet, les déclarations contiennent des concepts pour lesquels il n'existe pas de définition exacte, tels que : *grand, petit, proche, loin, etc.* Par exemples :
 - ◇ **La distance entre deux régions r1 et r2 est entre 55 km et 70 km.**
 - ◇ **L'hôtel Al-Ansar est très proche de la gare.**
- * **L'incomplétude**, on parle de l'incomplétude si une partie de l'information est absente. En effet, l'incomplétude correspond à l'absence (globale ou partielle) des données dans les bases de données ou par un jugement incomplet et incertain donné par un agent du Web sémantique.

* **L'inconsistance**, on parle de l'inconsistance si une phrase contient des connaissances contradictoires ou elle est contradictoire avec d'autres phrases.

4. **Les modèles de la gestion d'incertitude (Uncertainty Models)**. Cette classe contient des informations sur les théories mathématiques qui peuvent être employées pour faire face à l'incertitude à savoir :

- **La théorie des possibilités** est une approche qui permet de mesurer l'incertitude subjective sur des événements [45]. Elle permet de représenter des connaissances incomplètes et ambiguës. Ainsi, la théorie des possibilités est un cadre pour quantifier l'incertitude subjective sur des événements. Elle permet de formaliser deux connaissances, à quelle mesure un événement est possible et dans quelle mesure on est certain de la réalisation d'un événement [15].
- **La théorie des ensembles approximatifs** est un paradigme mathématique proposé par Zdzisław Pawlak [46, 47]. Cette théorie étend la théorie des ensembles classique, l'idée est de définir l'ensemble approximatif (rough set) par deux par deux autres ensembles, une approximation inférieure (approximation basse) et une approximation supérieure (approximation haute).
- La théorie de **Dempster–Shafer**, également appelée théorie des fonctions de croyance, est un formalisme général pour le raisonnement avec l'incertitude. Cette théorie a été conçue pour la première fois par P. Dempster [48] dans le contexte de l'inférence statistique, la théorie a ensuite été développée par Glenn Shafer dans une perspective plus générale pour la modélisation de l'incertitude [49]. Cette théorie permet de combiner des observations de différentes sources en retournant un seul degré de croyance. Ceci en se basant sur un paradigme théorique mathématique appelé fonction de croyance qui vise à fusionner les informations.
- Nous présentons en détails dans le prochain chapitre **la théorie des probabilités la logique floue et la combinaison de plusieurs modèles (hybride)**.

1.4.3 Incertitude dans les composants de l'ontologie

L'ontologie du haut-niveau définie par le W3C donne une vision globale sur l'incertitude liée au Web Sémantique. Cependant, elle n'est pas assez détaillée pour décrire comment l'incertitude est liée à la connaissance représentée dans l'ontologie d'un domaine d'application. Ainsi, elle ne montre pas les liens entre les formalismes qu'on doit utiliser et chaque type d'incertitude.

Selon [13, 14, 15], deux types de catégories de composants se distinguent dans une ontologie, la classe des composants **bien-définis** et la classe des composants **mal-définis**. En effet, l'auteur *Cyril de Runz* dans [15] a mis l'accent dans sa thèse sur les connaissances incertaines du domaine archéologique. Il a proposé une nouvelle classification d'incertitude, elle est illustrée dans la figure 1.8. Dans cette classification, l'incertitude est considérée comme un type d'imperfection.

Elle montre de manière explicite la taxonomie d'imperfection à savoir les types d'imperfection (incertitude, imprécision ou ambiguïté), les sous-types d'imperfection (vague, approximation, conflit ou non-spécificité) et comment ces sous-types d'imperfection sont liés à la connaissance représentée dans l'ontologie. En plus, elle permet de spécifier les formalismes convenables pour traiter chaque sous-type.

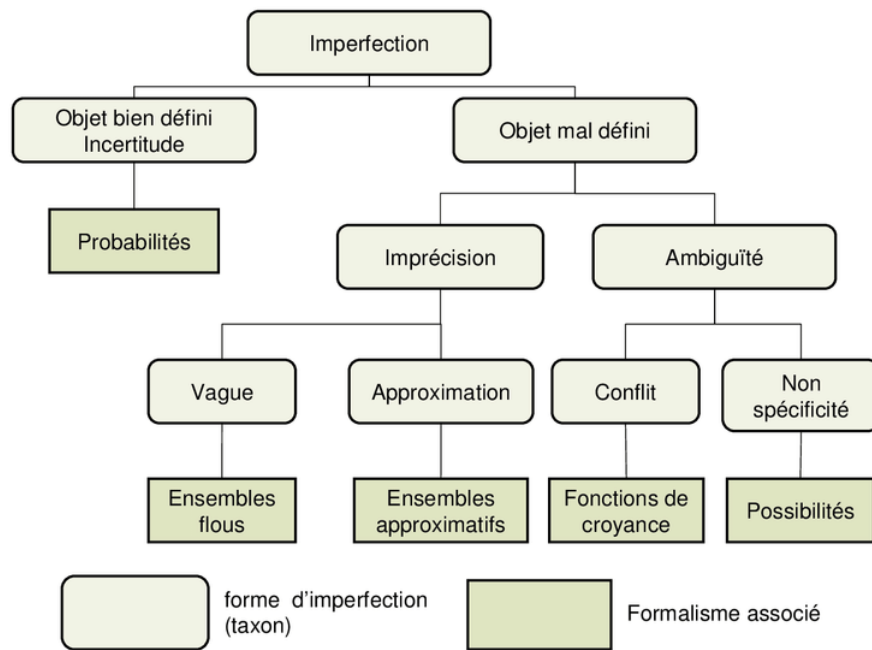


FIGURE 1.8 – Taxonomie de l'incertitude liée à la connaissance ontologique [15].

Dans la première catégorie, les composants de l'ontologie sont bien définis. Donc ce sont plus faciles à traiter et avec les approches probabilistes et l'incertitude liée à cette catégorie est peu profonde [14].

Contrairement à la première catégorie, la deuxième concerne les composants mal-définis et l'incertitude est en relation avec des problèmes de précision ou d'ambiguïté. Donc les approches qui doivent modéliser ce type d'incertitude moyenne ou profonde vont au-delà de la théorie des probabilités [14].

Différents modèles ont été exploités pour faire face à l'incertitude dans les ontologies. Ces modèles sont employés selon le type d'incertitude peu-profonde ou profonde et dépend parfois du domaine d'application.

- On parlera sur les "*connaissances probabilistes*" dans le cas où les composants sont **bien-définis**. L'incertitude qui affecte ces concepts provient du caractère aléatoire dû à une variabilité naturelle des phénomènes stochastiques ou par des erreurs qui se produisent à partir des données incomplètes. Ces concepts sont traitable par les approches probabilistes. Pour faire face à ce type de connaissances probabilistes dans les ontologies, un nouveau paradigme théorique a été développé, nommé « les ontologies probabilistes ». Il traite

les composants biens-définis de l'ontologie en se basant sur la théorie des probabilités, notamment, les Réseaux Bayésiens. Ainsi, les ontologies probabilistes constituent maintenant un axe de recherche prometteur et intéressant pour la communauté d'intelligence artificielle, d'ingénierie de connaissance et le Web Sémantique. Dans les années récentes, un bon nombre de travaux ont émergés pour définir les ontologies probabilistes, leurs fondements et leurs méthodes de construction. Dans le reste de cette thèse, nous allons passer en revue ces travaux (voir section 3.2).

- Si l'incertitude est due à la mauvaise définition de la classe d'objet ou d'un objet individuel. Dans ce cas, elle est liée à une classe ou un ensemble au sein de l'univers est une question du flou ou d'approximation. Elle peut être traitée respectivement par la théorie des ensembles flous ou les ensembles approximatifs.
 - **Connaissances floues**, l'information est floue si elle est définie à l'aide de connaissances subjectives ou flexibles (manque de détails), en effet, ces connaissances contiennent des termes dont leurs définitions ne sont pas précises [15].
 - **Connaissances approximatives**, la nature approximative de l'information provient de catégories aux limites mal définies. C'est-à-dire des objets dont la représentation ne peut être clairement définie [15].

Elle peut aussi survenir en raison de l'ambiguïté, c'est à dire, la confusion sur la définition des ensembles au sein de l'univers.

- **Conflit** (le désaccord), si au minimum deux classifications contradictoires pour un unique objet sont possibles [15].
- **Non-spécificité**, c'est lorsque la définition d'une relation ou d'un objet peut conduire à plusieurs sens, ou lorsque l'échelle de l'analyse peut amener à de multiples interprétations [15].

Les ontologies floues dans cette perspective, ont été fondées afin de surmonter les insuffisances des ontologies classiques. Elles permettent de modéliser et faire des raisonnements avec des composants mal-définis, notamment, les composants attachés avec la connaissance floue. Au cours de dernières années, plusieurs définitions et extensions aux langages du Web sémantique ont été proposées dans la littérature pour mettre en œuvre des ontologies floues. Ainsi, de nombreux raisonneurs sont apparus pour faire des déductions et des inférences floues sur des représentations floues telles que, la satisfiabilité des concepts flous, la subsomption de concepts flous, etc. Nous allons passer en revue ces extensions et ces raisonneurs dans le reste de cette thèse (voir section 3.3).

1.5 Bilan sur le Web sémantique et l'incertitude

Le Web sémantique vise à structurer les informations sur le Web, permettant aux agents logiciels de les gérer et les utiliser de manière plus intelligente. Cela peut être réalisé en représentant

le contenu du Web comme une connaissance structurée sémantiquement, d'une manière compréhensible et traitable par les machines. Dans ce contexte, les ontologies ont été considérées comme une clé pour la représentation et le traitement automatique de l'information au niveau sémantique. Les ontologies ont été appliquées avec succès dans plusieurs domaines afin de représenter les connaissances, de les partager et d'assurer l'interopérabilité entre les différents systèmes hétérogènes.

Cependant, avec le développement d'Internet et l'émergence du Big Data, la modélisation et le raisonnement avec l'information qui provienne de diverses sources de données réparties, incomplètes et imprécises, dans des environnements hétérogènes, inconsistants et incertains, deviennent un besoin crucial pour les applications du Web Sémantique. Il est souvent nécessaire pour ces applications d'acter et faire des décisions critiques dans de telles circonstances. L'omniprésence de tous ces problèmes dans la majorité des situations du monde réel représente un vrai défi pour les chercheurs de la communauté de l'intelligence artificielle, l'ingénierie de connaissances et notamment le Web sémantique.

Le Web sémantique repose essentiellement sur les capacités de la modélisation et de raisonnement des ontologies. Néanmoins, une limite majeure des ontologies classiques c'est qu'elles ne fournissent pas un support approprié pour gérer des connaissances incertaines, qui caractérisent la plupart des applications du monde réel. En effet, étudier l'incertitude qui accompagne la connaissance ontologique, permet de faire connaître et quantifier la fiabilité et la pertinence qui lui est accordée. Récemment, la gestion de l'incertitude est devenue un sujet de grand intérêt au sein de la communauté du Web sémantique ; les applications du Web sémantique doivent prendre en compte l'incertitude lors de la manipulation de la connaissance ontologique afin d'être à la hauteur des promesses du Web sémantique.

Comme nous l'avons déjà évoqué, différentes approches ont été distinguées pour faire face à l'incertitude dans le Web sémantique au niveau des connaissances ontologiques, telles que les approches possibilites [50, 51], les approches basées sur la théorie Dempster–Shafer [52, 53]. Dans le cadre de notre problématique, nous nous focalisons sur les approches basées sur la logique floue et les approches basées sur la théorie des probabilités.

1.6 Conclusion

Dans ce chapitre, nous avons présenté le Web sémantique comme une extension du Web actuel en discutant ses objectifs et les technologies utilisées pour sa construction. A travers ce chapitre nous avons constaté que l'ontologie est l'une des piliers sur lesquels repose le Web sémantique. Elle constitue un élément fondamental pour faire véhiculer les connaissances dans le Web. Nous avons exploré aussi l'incertitude dans le Web sémantique, notamment l'incertitude dans la modélisation ontologique et les modèles utilisés pour traiter l'incertitude dans les ontologies. Ainsi, le prochain chapitre sera consacré à détailler ces modèles, en particulier, la théorie des probabilités et la logique floue.

Chapitre 2

Approches formelles pour la gestion de l'incertain

“La connaissance progresse en intégrant en elle l'incertitude, non en l'exorcisant.”

— Edgar Morin

Sommaire

2.1	Introduction	27
2.2	Logique floue	27
2.2.1	Variables linguistiques	28
2.2.2	Ensembles flous	28
2.2.3	Sémantique de la logique floue	31
2.3	Théorie des probabilités	31
2.3.1	Probabilités conditionnelles	31
2.3.2	Sémantique de la probabilité	32
2.4	Réseaux Bayésiens Standards	33
2.4.1	Construction d'un réseau bayésien	34
2.4.2	Inférence dans les réseaux bayésiens	36
2.5	Réseaux Bayésiens Multi-Entités (MEBNs)	38
2.5.1	Modélisation des connaissances avec MEBN	38
2.5.2	Inférence dans MEBN	41
2.6	Réseaux Bayésiens Flous	44
2.6.1	Méthode de pondération	47
2.6.2	Distribution probabiliste floue	49
2.6.3	Méthode de l'évidence virtuelle	49
2.6.4	Bilan sur les méthodes d'inférence probabiliste floue	51
2.7	Discussion	51
2.8	Conclusion	53

2.1 Introduction

Les systèmes de gestion des connaissances ont pour but de modéliser le monde réel de manière formelle pour traiter ses connaissances automatiquement par des machines. Cependant, les systèmes de gestion des connaissances de la génération actuelle ont des capacités très limitées pour atteindre cet objectif. Pour ces systèmes, la gestion de l'incertitude qui est présente souvent dans la plus part des scénarios du monde réel est devenue un problème sérieux. Pour faire face à la connaissance incertaine représentée dans des ontologies, diverses solutions ont été développées dans la littérature qui s'appuie sur des modèles mathématiques.

L'objet de ce chapitre est de présenter les modèles de gestion de l'incertain. La première section est consacrée à l'exploration de la logique floue. La section qui suit présente la théorie de probabilités. La section suivante expose les réseaux bayésiens comme étant un modèle graphique probabiliste pour la gestion de la connaissance probabiliste. Les réseaux bayésiens multi entités, une extension des RBs classiques pour augmenter leur expressivité et flexibilité sont exposés dans la section qui suit. Finalement, la dernière section étale les réseaux bayésiens flous qui sont une combinaison des RBs standards et la logique floue pour bénéficier des avantages des deux formalismes en matière de modélisation et de raisonnement.

2.2 Logique floue

Dans le monde réel, nous rencontrons souvent des situations dans lesquelles nous ne pouvons pas déterminer si leurs états sont vrais ou faux. Autrement dit, ces états sont flous, le terme «flou» fait référence à des affirmations qui sont vagues et ne sont pas claires. En effet, la logique floue est conçue pour faire face à une telle situation. Elle vise à traduire un ensemble de règles de langage humain fournies par les experts en leurs équivalentes règles mathématiques pour simplifier le travail du concepteur de système. Ceci va offrir une flexibilité très précieuse pour le raisonnement et donne des représentations beaucoup plus naturelles du comportement des systèmes du monde réel.

Dans la logique booléenne, chaque proposition est vraie ou fausse, c'est-à-dire qu'elle peut prendre la valeur de vérité 1 ou 0. Les ensembles booléens exigent des conditions d'adhésion strictes. En revanche, les ensembles flous ont des exigences d'adhésion plus flexibles permettant une adhésion partielle à un ensemble.

En effet, la théorie des ensembles classiques est basée sur la logique booléenne où un élément appartient ou non à un sous-ensemble. Cependant, la logique floue tolère et représente l'imprécision relative à l'appartenance des éléments aux sous-ensembles. Elle autorise à un élément d'appartenir totalement ou partiellement à un sous-ensemble. Le terme d'ensemble flou et la logique floue apparaissent pour la première fois en 1965, lorsque le professeur Lotfi A. Zadeh, a publié un article intitulé « Ensembles flous » [35]. Il porte sur la notion de sous-ensembles flous pour fournir une alternative de gestion des affirmations imparfaitement décrites et vagues .

Dans ce qui suit, nous présentons les variables linguistiques, les ensembles flous et la sémantique de la logique floue.

2.2.1 Variables linguistiques

Une variable linguistique (concept vague) représente un état dans le système à modéliser. Sa valeur est définie par des termes utilisés lors de la description de la proposition étudié. Ils peuvent être des mots ou des phrases d'un langage naturel tels que : Très faible, Faible, Moyen, élevé, Très élevé. En effet, chaque variable linguistique est caractérisée par le triple $\langle x, T(x), U \rangle$ [54].

- x est le nom de la variable.
- $T(x)$ est l'ensemble des valeurs linguistique que peut prendre x .
- U est l'univers du discours associé avec la valeur de base.

Par exemple, T (Température) = (**froide, médium, chaude**). Chaque terme dans T (Température) est caractérisé par un ensemble flou dans l'univers de discours $U = [0, 50]$.

2.2.2 Ensembles flous

Un sous-ensemble flou F dans un univers de discours X peut être représenté par un ensemble de couples ordonnés :

$$F = \{(x, \mu_F(x)) / x \in X\} \quad (2.1)$$

Formellement, chaque sous ensemble floue F est caractérisé par sa propre fonction d'appartenance donnée comme suit [55] :

$$\mu_F(x) : X \rightarrow [0, 1]. \quad (2.2)$$

Où,

- $\mu_F(x) = 1$, si et seulement si x est totalement dans F .
- $\mu_F(x) = 0$, si x n'appartient pas à F .
- $0 < \mu_F(x) < 1$, si x appartient partiellement à F .

La figure 2.1 donne une illustration graphique d'un ensemble classique et d'un ensemble flou.

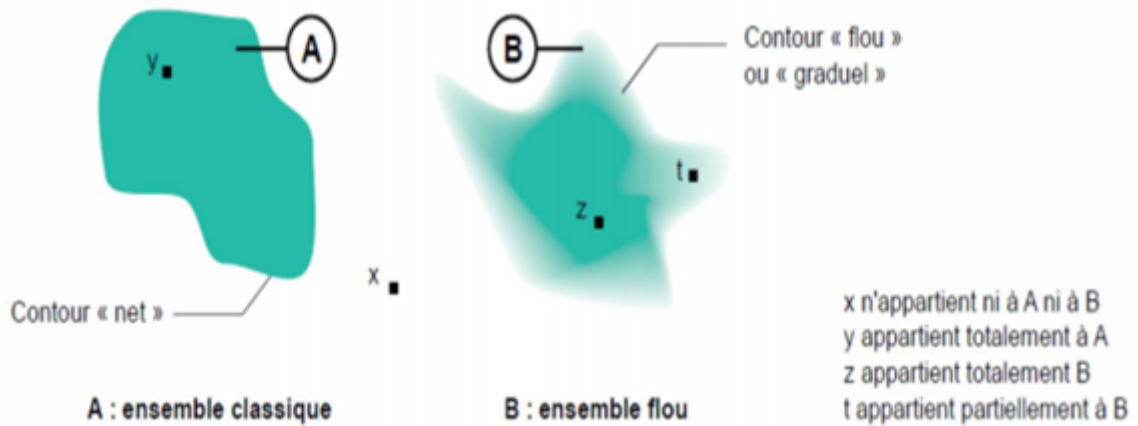


FIGURE 2.1 – Illustration graphique d'un ensemble classique et d'un ensemble flou [56].

Il n'existe pas une règle générale dans la littérature pour définir la fonction d'appartenance. La loi de la fonction et ses paramètres dépendent du domaine d'application et de la contribution de l'expert. Chaque fonction d'appartenance est caractérisée par les propriétés suivantes :

- **Le Type** : La forme de la fonction qui peut être Triangulaire, Trapèze, Right-Shoulder et Left-Shoulder.
- **La Hauteur** : La hauteur de la fonction d'appartenance A, noté : $h(A)$

$$h(A) = \sup\{\mu_A(x) \mid x \in X\}. \quad (2.3)$$

En effet, un sous-ensemble flou est dit normalisé s'il a une hauteur égale à 1.

- **Le Noyau** : Le noyau $N(A)$ est l'ensemble des éléments qui appartiennent totalement à A.

$$N(A) = \{x \in X \mid \mu_A(x) = 1\} \quad (2.4)$$

- **Le Support** : Cet ensemble décrit l'ensemble des éléments qui sont partiellement dans A. il est défini comme :

$$\text{supp}(A) = \{x \in X \mid \mu_A(x) > 0\} \quad (2.5)$$

Chaque fonction est définie par une loi permettant de calculer les degrés d'appartenance de n'importe quel élément dans l'univers de discours à un sous-ensemble.

- **Classique (Crisp)**. Une fonction d'appartenance Classique est définie par deux arguments a, b . La loi d'une fonction d'appartenance Classique est donnée comme suite :

$$\mu(x) = \begin{cases} 1 & \text{si } a \leq x < b \\ 0 & \text{sinon.} \end{cases} \quad (2.6)$$

- **Triangulaire**. Une fonction d'appartenance Triangulaire est définie par trois arguments a, b et c. La loi d'une fonction d'appartenance triangulaire est donnée comme suite :

$$\mu(x) = \begin{cases} \frac{x-a}{b-a} & a \leq x < b \\ \frac{x-c}{b-c} & b \leq x \leq c \\ 0 & \text{sinon.} \end{cases} \quad (2.7)$$

— **Trapèze.** Une fonction d'appartenance Trapèze est définie par quatre arguments a , b , c et d . La loi d'une fonction d'appartenance Trapèze est notée par (a, b, c, d) , donnée comme suite :

$$\mu(x) = \begin{cases} \frac{x-a}{b-a} & a \leq x < b \\ 1 & b \leq x < c \\ \frac{d-x}{d-c} & c \leq x < d \\ 0 & \text{sinon.} \end{cases} \quad (2.8)$$

— **Right-Shoulder.** Une fonction d'appartenance Right-Shoulder est définie par trois arguments a , b et c . La loi d'une fonction d'appartenance Right-Shoulder est donnée comme suite :

$$\mu(x) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{x-b} & a < x < b \\ 1 & x \geq b \end{cases} \quad (2.9)$$

— **Left-Shoulder.** Une fonction d'appartenance Left-Shoulder est définie par trois arguments a , b et c . La loi d'une fonction d'appartenance Left-Shoulder est donnée comme suite :

$$\mu(x) = \begin{cases} 1 & x \leq a \\ \frac{-x+a}{b-a} & a < x < b \\ 0 & x \geq b \end{cases} \quad (2.10)$$

La représentation graphique de ces fonctions d'appartenances est illustrée dans la figure 2.2.

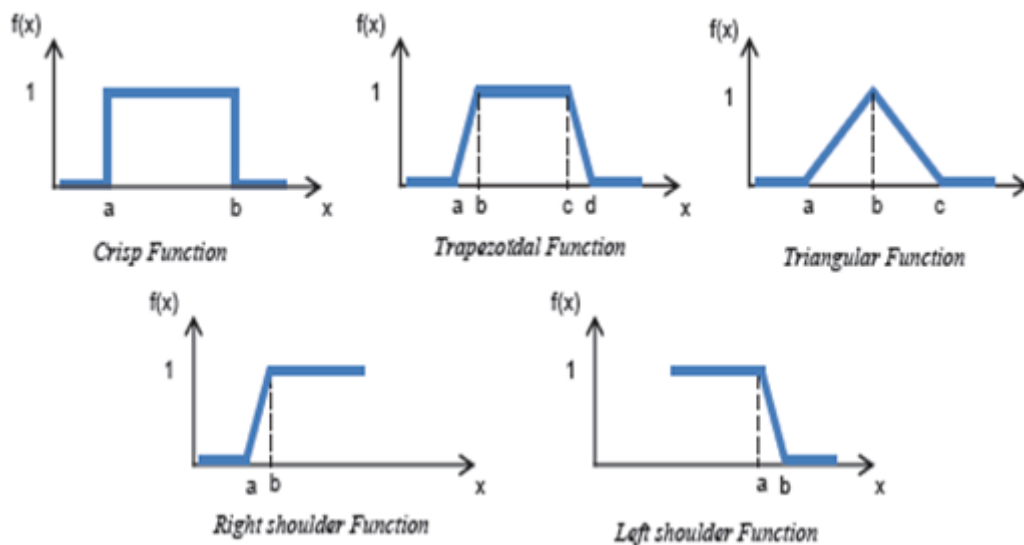


FIGURE 2.2 – Fonctions d'appartenances usuelles [35].

2.2.3 Sémantique de la logique floue

La logique floue est une approche basée sur les sous-ensembles flous permettant de manipuler la connaissance floue et fournit un raisonnement approximatif en se basant sur des vérités partielles. La logique floue permet de définir des valeurs d'appartenances comprises entre 0 et 1. Ceci permet de modéliser l'adhésion partielle à un ensemble en utilisant des fonctions d'appartenances ainsi que la représentation des variables linguistiques des concepts flous de manière naturelle tels que "Assez" et "Très", etc.

La fonction d'appartenance est alors une mesure :

- Du degré auquel un élément est membre d'un ensemble.
- Du degré d'appartenance.

2.3 Théorie des probabilités

Avant d'entamer les notions de bases sur les réseaux bayésiens, nous donnons dans cette section un bref rappel sur la théorie des probabilités. En effet, la théorie des probabilités est l'étude mathématique des phénomènes caractérisés par le caractère aléatoire ou incertain. Ainsi, la probabilité est utilisée pour modéliser des événements lorsque les résultats d'une expérience réalisée souvent dans les mêmes conditions et circonstances ne sont pas toujours les mêmes, par exemple : lancer un dé. Nous allons décrire dans les sections suivantes les probabilités conditionnelles et la sémantique de la probabilité.

2.3.1 Probabilités conditionnelles

Définition 2.3.1 (La Probabilité Conditionnelle). Soit $(\Omega, P(\Omega), P)$ un espace probabiliste lié à une expérience aléatoire. Soient A et B deux événements de Ω , tel que $P(B) \neq 0$.

La probabilité que l'évènement A soit réalisé sachant que B est réalisé est définie de la manière suivante :

$$P(A | B) = \frac{P(A \cap B)}{P(B)} \quad (2.11)$$

Soient A et B deux événements dans l'univers Ω . Étant donné qu'on connaît la probabilité de B et la probabilité conditionnelle de A sachant B , on peut trouver la probabilité $P(A \cap B)$ de l'intersection de A et B par la formule suivante (Théorème des probabilités composées) :

$$P(A \cap B) = P(A)P(B | A) = P(B)P(A | B). \quad (2.12)$$

Pour tous les événements A et B tel que $P(A) \neq 0$, les propriétés suivantes sont vérifiées :

- $P(\Omega) = 1$.
- $P(A | A) = 1$.
- $P(A \cup B) = P(A) + P(B) - P(A \cap B)$.

- Dans le cas où A et B sont incompatibles (c.à.d. $A \cap B = \Omega$), alors :
 $P(A \cup B) = P(A) + P(B)$, $P(A \cap B) = 0$ et $P(A | B) = 0$

Dans la théorie des probabilités, le théorème de Bayes permet de décrire la probabilité pour que une variable X soit à l'état x sur la base d'une connaissance préalable sur d'autres variables liées avec X.

Exemple 2.3.1 Si le cancer est lié à l'âge, alors en utilisant le théorème de Bayes, on peut évaluer la probabilité de développer le cancer chez un patient en utilisant son âge avec une précision, par rapport à l'évaluation de la probabilité de cancer sans la connaissance de son âge.

Soient A et B sont deux évènements, la loi de Bayes est donnée par la formule suivante :

$$P(A | B) = \frac{P(B | A)P(A)}{P(B)}. \quad (2.13)$$

Où :

- $P(B) \neq 0$. C'est-à-dire, l'évènement B est réalisable (**c'est une constante de normalisation**).
- $P(B | A)$: la probabilité que l'évènement B se produit étant donné que l'évènement A est réalisé (**la vraisemblance**).
- $P(A | B)$: la probabilité que l'évènement A se produit étant donné que l'évènement B est réalisé. On appelle cette mesure **la probabilité a posteriori**.
- $P(A)$ et $P(B)$ sont des probabilités marginales, elles représentent les probabilités d'observer $P(A)$ et $P(B)$ indépendamment l'une de l'autre (c'est des **probabilités a priori**).

2.3.2 Sémantique de la probabilité

La valeur de probabilité accordée à un évènement peut être interprétée de plusieurs manières. En effet, selon (DUDA et al.,) [57]. La seule description satisfaisante de l'incertitude est sous forme probabiliste qui possède deux interprétations à savoir l'interprétation subjective et l'interprétation objective.

- Les *Probabilités objectives*, nommées aussi des probabilités bayésiennes sont calculées par l'étude expérimentale des phénomènes aléatoires en répétant le même évènement plusieurs fois. La fréquence d'apparition d'une configuration précise sera utilisée pour déterminer sa probabilité.
- Les *Probabilités subjectives* sont définies comme des degrés de croyances, elles ne peuvent pas être confirmées par l'expérience car l'évènement est unique et irréversible. Généralement, les probabilités subjectives sont fournies par l'expertise d'un ensemble d'experts du domaine en question.

Ainsi, les probabilités objectives sont très faciles à valider à travers des expérimentations. Par contre les probabilités subjectives ne sont pas faciles à valider et leur exactitude dépend de l'expertise des experts.

2.4 Réseaux Bayésiens Standards

Les réseaux bayésiens [58, 59, 60, 61, 62, 63] constituent l'une des approches les plus efficaces pour représenter, analyser et traiter des connaissances probabilistes. En effet, un RB est un modèle hybride combinant la théorie des graphes et la théorie des probabilités. Récemment, ils ont été largement utilisés dans de nombreux domaines tels que le diagnostic médical, l'analyse des risques, etc.

Les RBs ont été proposés ainsi afin d'introduire la notion d'incertitude dans les systèmes experts en se basant sur le théorème de Bayes. Ils sont capables de représenter des connaissances qualitatives à savoir les liens de causalités et les indépendances entre les variables et une connaissance quantitatives sous forme des probabilités. Les RBs ont été adoptés récemment dans de nombreux domaines tels que le domaine médical, le domaine industriel, etc.

Définition 2.4.2 (Réseau Bayésien) *Un réseau bayésien sur un ensemble de variables aléatoire V discrètes ou continues, nommées aussi nœuds, est un couple (\mathbf{G}, \mathbf{P}) , tel que :*

- $\mathbf{G} = (X, E)$ est un graphe orienté acyclique où X représente l'ensembles des sommets et E l'ensemble d'arcs. Chaque $(x, y) \in E$ indique une dépendance causale entre x et y , c'est-à-dire, x influe directement sur y .
- Chaque nœud Y est associé avec une table de probabilités conditionnelle (CPT : Conditional Probability Table), $P(y | P_a(y))$, où : $P_a(y)$ est l'ensemble des parents du nœud y dans G .
- P est une distribution de probabilités sur X .

Exemple 2.4.2 *Pour bien illustrer la notion des RBs, on considère un exemple de réseau bayésien qui modélise un processus aléatoire où des voisins appellent au bureau d'une personne s'ils entendent une alarme sonner dans l'appartement de cette personne. Cette alarme peut être déclenchée, par un tremblement de terre ou un cambriolage. Le graphe qui modélise cette situation est illustré dans la figure 2.3.*

Un réseau de bayésien fournit une description complète d'un domaine particulier en permettant d'exprimer les relations de causalités entre les variables du modèle. Chaque variable est représentée par un sommet et il existe un arc allant de X_1 à X_2 si et seulement si X_1 influe directement sur X_2 . Étant donné un réseau bayésien R , il est possible de factoriser la loi de probabilité jointe sur l'ensemble de variables $X = \{X_1, \dots, X_n\}$ comme suit :

$$P(X_1, \dots, X_n) = \prod_{i=1}^n P(X_i | P_a(X_i)). \quad (2.14)$$

Tel que, $P_a(X_i)$ représente l'ensemble de parents du noeud X_i .

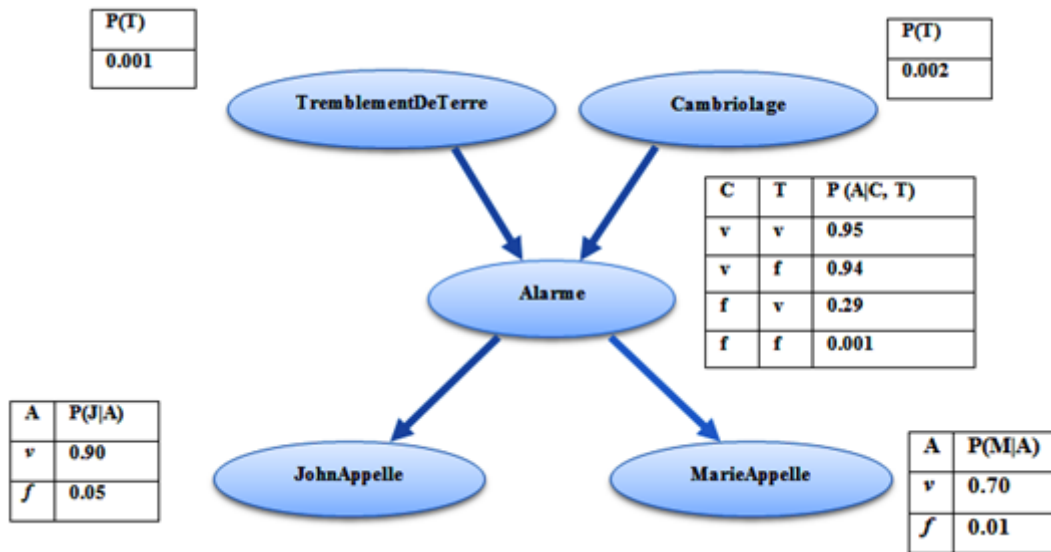


FIGURE 2.3 – Exemple d'un réseau bayésien [59] .

Dans les sections suivantes, nous présentons les différentes méthodes de construction des RBs et les algorithmes d'inférences dans les RBs.

2.4.1 Construction d'un réseau bayésien

Les algorithmes d'apprentissage dans les réseaux bayésiens reviennent à définir un modèle RB qui s'adapte mieux aux connaissances du problème à représenter et à traiter. En effet, le problème d'apprentissage consiste à faire l'apprentissage de la structure et l'apprentissage des paramètres. L'apprentissage de la structure consiste à définir une meilleure liaison entre les différentes variables aléatoires afin d'avoir un graphe ou une structure optimale. Quant à l'apprentissage des paramètres, il consiste à estimer les paramètres du RB qui correspond à la distribution des probabilités conditionnelles et à priori de chaque variable.

2.4.1.1 Apprentissage de la structure

L'apprentissage de la structure d'un réseau bayésien consiste à identifier les nœuds et les connexions possibles entre ces nœuds à partir des données d'apprentissage. La recherche de la bonne structure de réseaux bayésiens est un problème NP-complet [64]. Ceci est due au fait que l'espace de recherche est exponentiel en fonction du nombre de variables représentant le problème.

Deux grandes catégories de méthodes ont été développées pour l'apprentissage de la structure. **Les méthodes de la première catégorie** sont basées sur la recherche des indépendances conditionnelles entre les différents nœuds [65]. **Les méthodes de la deuxième catégorie** sont basées sur *le score du graphe*. Un algorithme simple pour trouver la meilleure structure d'un

réseau bayésien est donc de parcourir de façon exhaustive l'ensemble de tous les graphes possibles. Par la suite, il faut associer un score à chaque graphe, puis choisir le graphe ayant la plus grande valeur de score comme une structure optimale. Ainsi d'autres algorithmes vise à optimiser la recherches ont été développés tel que l'algorithme efficace **K2** [66].

En effet, l'apprentissage de la structure ne suffit pas pour la construction d'un réseau bayésien, il s'agit encore de définir les tables de probabilités conditionnelles de chaque variable, on appelle cette tâche l'apprentissage des paramètres.

2.4.1.2 Apprentissage des paramètres

L'apprentissage des paramètres consiste à estimer les distributions de probabilités. Ce qu'on appelle aussi paramètres des lois correspondantes du réseau bayésien à partir des données disponibles. Nous présentons dans cette section les techniques d'apprentissage de paramètres selon le type de données complètes ou incomplètes.

- **Les données complètes** : dans le cas où la base d'apprentissage est complète et il n'y a pas des données manquantes. Il s'agit d'estimer la probabilité d'un événement en se basant sur la fréquence d'apparition de l'événement dans la base de données. On appelle cette approche maximum de vraisemblance (MV).

Soit $N_{i,j,k}$ le nombre d'occurrences simultanées dans la base où de la variable X_i est dans l'état x_k et ses parents sont dans la configuration x_j . Les paramètres du réseau $\Theta_{i,j,k}^{(MV)}$ sont donnés comme suit :

$$\Theta_{i,j,k}^{(MV)} = N_{i,j,k}^* = \sum_{i=1}^n P(X_i = x_k \mid P_a(X_i) = x_j) \quad (2.15)$$

- **Les données incomplètes** : les données incomplètes peuvent souvent apparaître dans les bases de données où certaines valeurs sont manquantes. Dans cette optique, l'algorithme itératif Expectation, Maximisation (EM) a été proposé par Dempster [67] afin d'estimer les paramètres du réseau à partir des données incomplètes. L'idée de l'algorithme EM est de rechercher les paramètres en répétant les deux étapes Espérance et Maximisation jusqu'à la convergence.

Algorithm 1: L'algorithme EM

1 Initialiser $\Theta^{(0)}$ aléatoirement à l'instant $t=0$

2 **repeat**

— **Espérance** : Utilisation des paramètres courants $\Theta_{i,j,k}^{(t)}$ pour estimer l'espérance d'apparition des différentes configurations.

$$N_{i,j,k}^* = \sum_{i=1}^n P(X_i = x_k | Pa(X_i) = x_j)$$

— **Maximisation** : Estimation des nouveaux paramètres par maximum de vraisemblance en utilisant l'espérance des statistiques obtenue à l'étape précédente.

$$\Theta_{i,j,k}^{(t+1)} = \frac{N_{i,j,k}^*}{\sum_k N_{i,j,k}^*}$$

3 **until** $|\Theta_{i,j,k}^{(t+1)} - \Theta_{i,j,k}^{(t)}| < \xi$;

2.4.2 Inférence dans les réseaux bayésiens

Étant donné le modèle réseaux bayésiens composé d'un ensemble de variables aléatoires $X = \{X_1, X_2, \dots, X_n\}$, on cherche à calculer la probabilité qu'une variable $X_i \in X$ est dans l'état x_i , sachant que l'on connaît la valeur de certaines d'autres variables (variables d'observations) $Y = \{Y_1 = y_1, Y_2 = y_2, \dots, Y_m = y_m\} \subset X$.

En effet, l'inférence bayésienne est une sorte de calcul de probabilités conditionnelles. Il s'agit de mettre à jour des probabilités des variables non observées après l'observation des valeurs d'un certain nombre d'autres variables afin de calculer les probabilités a posteriori sous forme $P(X_i = x_i | Y)$ du modèle. Par ailleurs, le problème d'inférence se formule comme la tâche de trouver la probabilité suivante :

$$P(X_i = x_i | Y_1 = y_1, Y_2 = y_2, \dots, Y_m = y_m). \quad (2.16)$$

Ainsi, une application directe du théorème de Bayes lors de l'inférence bayésienne peut conduire à des calculs très coûteux en temps d'exécution (une explosion combinatoire). Pour cette raison, plusieurs travaux ont été développés dans le domaine de l'inférence dans les réseaux bayésiens. En effet, deux grandes familles d'algorithmes d'inférence ont été distinguées : les méthodes d'inférence exactes et les méthodes d'inférence approchées. Nous présentons dans ce qui suit un aperçu de ces méthodes.

2.4.2.1 Inférence exacte

Les algorithmes d'inférence exacte utilisent les indépendances conditionnelles contenues dans le réseau associé à chaque nœud pour calculer les probabilités à posteriori exactes.

L'inférence dans les réseaux bayésiens par la méthode d'envois de messages locaux a été introduite par Pearl dans [68]. En effet, pour chaque nœud du réseau, il s'agit de mettre à jour régulièrement la distribution des probabilités marginales de proche en proche avec ses voisins pour prendre en compte les observations sur un ou plusieurs nœuds, par transmission de messages entre variables voisines dans le graphe. Néanmoins, le problème avec cette méthode réside dans la structure où elle est utilisable que lorsque le réseau bayésien possède une forme d'arbre.

L'auteur dans [69] a introduit l'algorithme de « Ensemble de coupe » dont l'objectif est de changer la connectivité du réseau afin de trouver un ensemble de variables appelées « ensemble de coupe » ; dans le graphe obtenu, l'inférence sera effectuée en utilisant l'algorithme des messages locaux.

Par ailleurs, la méthode de l'arbre de jonction a été développée dans [70, 71], cette nouvelle démarche consiste à construire une structure plus raffinée à partir du réseau initial nommée l'arbre de jonction. En effet, la construction de l'arbre de jonction nécessite un ensemble de sous-étapes permettant de transformer le graphe initial en un arbre de jonction, dont les nœuds sont des clusters (regroupement) de nœuds du graphe initial afin d'obtenir un graphe plus efficace. Par la suite, il s'agit d'une phase de propagation afin de calculer les probabilités à posteriori où les nouvelles observations concernant une ou plusieurs variables sont propagées dans l'ensemble du réseau, de manière à mettre à jour l'ensemble des distributions de probabilités du réseau. Ceci se fait en utilisant la méthode d'inférence par les messages locaux et l'inférence se fait sur les potentiels des cliques construits (nœuds de l'arbre de jonction).

En effet, il est prouvé que l'inférence exacte dans les RBs est un problème NP-Difficile¹ [72]. Dans les cas pratiques, la complexité des algorithmes d'inférence dépend de plusieurs facteurs, notamment : le nombre de nœuds du réseau bayésien, i.e., la taille du réseau, la structure du réseau bayésien, i.e., nombre de chemins multiples entre les nœuds. Pour cette raison, de nombreuses méthodes d'inférence approchée ont été proposées afin de surmonter cette limite. Dans la prochaine section nous donnons un aperçu sur ces méthodes.

2.4.2.2 Inférence approchée

Les méthodes d'inférences exactes ne sont pas efficaces pour de grands réseaux. C'est pourquoi les approches approximatives ont été proposées. En effet, les méthodes d'inférence approchées donnent des estimations approximatives des probabilités a posteriori. Par ailleurs, il est possible de distinguer deux familles principales des méthodes d'inférences approchées.

1. L'auteur a démontré que l'inférence bayésienne est NP-Difficile en faisant une réduction de problème d'inférence bayésienne au problème connu 3SAT.

- La première sous-classe d’algorithmes vise à simplifier le réseau. Il s’agit par exemple de modifier la topologie du réseau en supprimant les arcs ayant des poids faibles ou bien d’attribuer la valeur zéro aux probabilités faibles pour les ignorer lors du processus d’inférence ce qui va diminuer le temps du calcul. Parmi ces solutions, l’algorithme d’inférence proposé dans [73] qui consiste à supprimer les relations de dépendance les plus faibles du graphe afin de réduire la taille des tables de probabilités et par conséquent réduire le temps de calcul.
- D’autres méthodes d’inférences approchées, appelées méthodes stochastiques, d’échantillonnage, ou Sampling. Elles visent à tirer N échantillons d’une distribution d’échantillonnage et calculer par la suite une probabilité a posteriori approximative.

2.5 Réseaux Bayésiens Multi-Entités (MEBNs)

Au cours des dernières années, plusieurs approches ont été proposées qui visent à intégrer les représentations de premier ordre avec les modèles graphiques sémantiques probabilistes. Dans ce contexte, MEBN est proposé comme une extension de réseau bayésien en utilisant la logique du premier ordre (FOL : First Order Logic). Ils fournissent la puissance expressive de la logique du premier ordre et les capacités du raisonnement probabiliste des réseaux bayésiens [74, 75]. Comme les réseaux bayésiens classiques, les MEBNs reposent sur la représentation graphique sous forme de graphe dirigé acyclique de variables aléatoires.

La nouveauté du MEBN réside dans sa capacité de représenter les attributs internes des variables aléatoires et les relations entre ces variables aléatoires. Il étend également les capacités des réseaux bayésiens classiques pour permettre la représentation des modèles graphiques avec des sous-structures répétées. La relation incertaines entre des groupes de variables aléatoires liées ayant une sémantique commune est exprimée en termes de distribution de probabilité représentée par des fragments MEBN (MFragments).

Dans MEBN, la connaissance est représentée comme une collection de fragments de réseaux bayésiens qui peuvent être instanciés et combinés plusieurs fois. Cette instantiation vise à générer des réseaux bayésiens standards pour former des réseaux bayésiens spécifiques à une situation précise (SSBN : Specific Situation Bayesian Network) qui seront utilisés par la suite pour la prise de décision.

2.5.1 Modélisation des connaissances avec MEBN

Un MEBN est un ensemble de théories composées de MFragments bayésiens. Une théorie MEBN (MTheory) représente implicitement une distribution de probabilité conjointe sur les variables aléatoires qu’elles comportent. Un MFrag représente la connaissance incertaine d’une collection de variables aléatoires liées. Les variables aléatoires représentent les attributs et les propriétés d’un ensemble d’entités. En effet, un MFrag est un graphe orienté acyclique qui représente les dépendances entre ses variables aléatoires.

Les variables aléatoires d'un MFrag sont des expressions de la logique du premier ordre. Elles contiennent généralement comme des arguments une ou plusieurs variables ordinaires. Par ailleurs, les variables ordinaires sont des variables qui sont substituées par des instances des entités au cours du processus d'instanciation (à l'étape d'inférence) selon les observations disponibles dans la base de connaissances. Formellement, un MFrag est défini comme un 5-tuple $\mathbf{F} = (\mathbf{C}, \mathbf{I}, \mathbf{R}, \mathbf{G}, \mathbf{D})$:

- **Nœuds de Contextes (C : Context Nodes)** : ce sont des variables aléatoires représentent des conditions qui doivent être satisfaites pour que la distribution dans un MFrag soit valide.
- **Nœuds Résidants (R : Resident Nodes)** : représentent les variables aléatoires qui forment le cœur du sujet d'un MFrag. Les variables pertinentes ayant une certaine sémantique commune seront regroupées pour construire un fragment MFrag. La distribution probabiliste locale (LPD : Local Probability Distribution) de chaque variable doit être unique et défini explicitement dans son MFrag original.
- **Nœuds d'entrées (I : Input Nodes)** : les nœuds d'entrées fournissent un moyen permettant la réutilisation des nœuds résidents déjà existants et faire les liens sémantiques entre les MFrag. Ils représentent des pointeurs afin de référencer des nœuds résidents d'un autre MFrag. En effet, chaque nœud d'entrée doit pointer au moins vers un nœud résident dans un MFrag. Les nœuds d'entrées peuvent influencer sur les distributions des probabilités des nœuds résidents. Néanmoins, leurs propres distributions sont définies ailleurs (c'est à dire dans leurs propres MFrag d'origine).
- **G** : représente un graphe MEBN orienté acyclique. Il permet de décrire les relations de dépendances entre les nœuds.
- **D** : est un ensemble de distributions locales liées à chaque nœud. Ces distributions seront utilisées afin de construire les tables de probabilités conditionnelles lors de l'inférence.

Exemple 2.5.3 *Pour illustrer les concepts de bases des MEBNs, prenant l'exemple montré dans la figure 2.4, modélisant une MTheory pour soutenir le diagnostic d'un patient entrant dans une clinique avec une fièvre. S'il a récemment visité une région touchée par une épidémie de grippe, cela pourrait en être la cause. Sinon, le patient a un autre type de virus [76].*

Cette MTheory contient trois MFrag qui sont : Region-MF, RegionVisit-MF et Patient-MF.

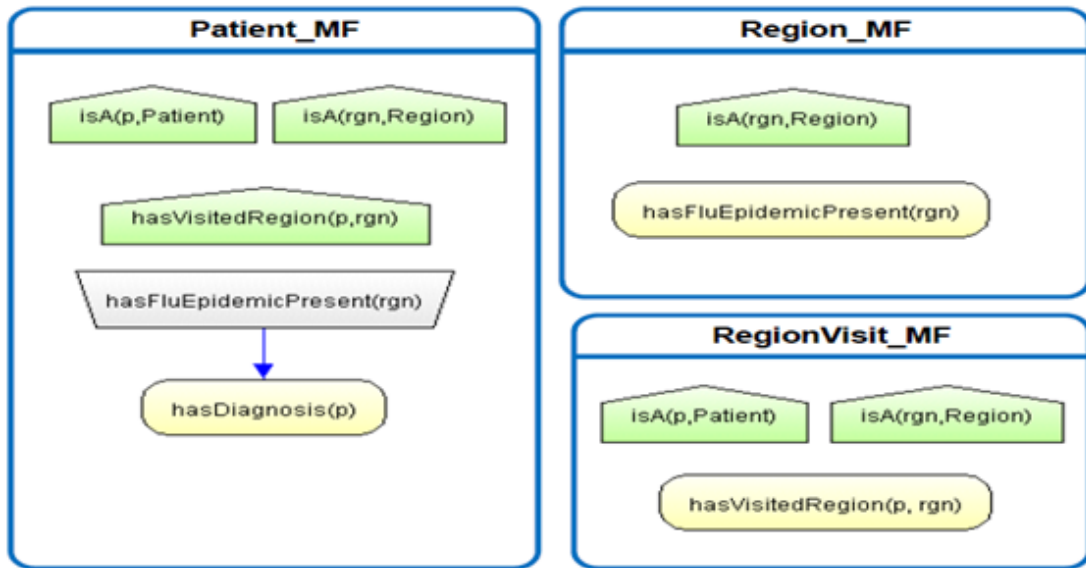


FIGURE 2.4 – Exemple d’une MThéorie [76] .

- **RegionVisit-MF** est composé de trois nœuds, deux nœuds de contextes : `isA (rgn, Region)` pour définir une variable de type region et `isA (p, Patient)` pour définir une variable ordinaire de type patient. Elle contient aussi un nœud résident `HasVisitRegion(p, rgn)`, ce nœud représente la connaissance à propos des visites des patients.
- **Region-MF** contient deux nœuds, un nœud de contexte `isA (rgn, Region)` pour définir une variable ordinaire de type region, et un nœud résident booléen nommé `hasFluEpidemicPresent(rgn)` pour savoir si la Region présente une épidémie de grippe ou non.
- **Patient-MF** représente un le Mfrag sur lequel nous allons lancer des tests de diagnostics sur des patients, il contient cinq nœuds, deux nœuds de contextes « `isA (p, Patient)` et `isA (rgn, Region)` » pour définir des variables ordinaires `p` et `rgn` et le troisième nœud de contexte
- **HasVisitRegion(p, rgn)** pour définir une condition, ce nœud est importé à partir du Mfrag `Region-MF`. Il contient un nœud d’entrée `hasFluEpidemicPresent(rgn)` qui est importé du Mfrag `RegionVisit-MF` et finalement le nœud résident `hasDiagnosis(p)` pour diagnostiquer le patient. Par ailleurs, la définition de la LPD de `hasDiagnosis(p)` est illustrée dans la figure 2.5.

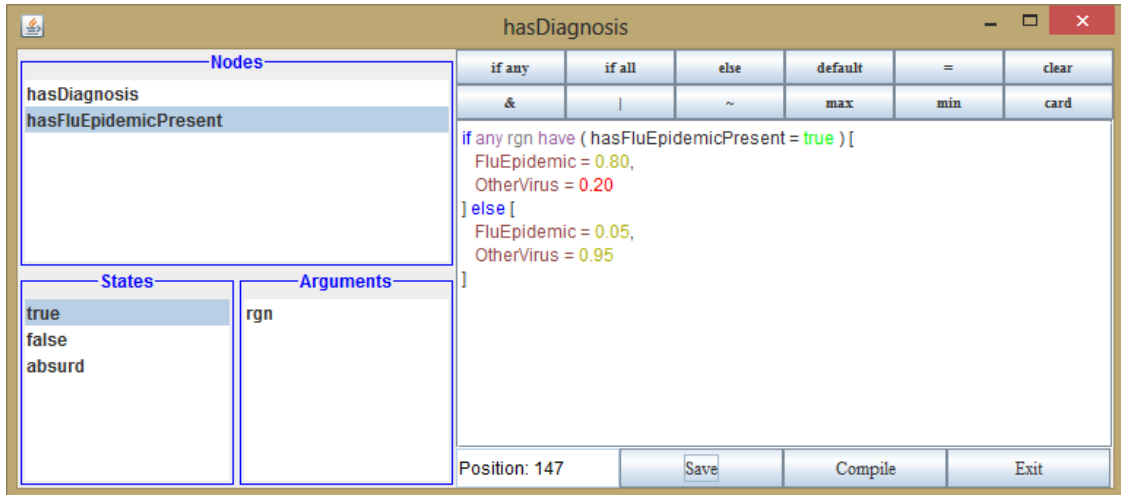


FIGURE 2.5 – Une Distribution Locale du noeud hasDiagnosis [76]

2.5.2 Inférence dans MEBN

Les MEBNs sont fondés sur le concept graphique de représentation de la distribution de probabilité conjointe pour un ensemble de variables aléatoires dans un domaine d'intérêt particulier. Ils visent à améliorer les réseaux bayésiens standards en combinant la sémantiques de la logique du premier ordre et les capacités de modélisation et du raisonnement sous l'incertitude offertes par la théorie des probabilités. En effet, MEBN est un langage du premier ordre pour la modélisation des bases de connaissances probabilistes en se basant sur les fragments de réseaux bayésiens. Il est très expressif en matière de représentation probabiliste des bases de connaissances car il permet de modéliser des problèmes très complexes où l'incertitude règne.

L'idée de l'inférence dans MEBN est de commencer par une liste de requêtes et une liste de noeuds d'observations, puis, pour chaque noeud de l'ensemble, évaluer son MFrag, en instanciant les noeuds résidents et les noeuds d'entrées sur la base des valeurs des variables ordinaires des noeuds de contexte dans la même MFrag. Cela va permettre de construire un RB standards spécifique à la situation observée. L'algorithme d'inférence de laskey est donné par les étapes déclinées dans les étapes ci-dessous [74] :

Algorithm 2: Génération d'un SSBN -L'algorithmme d'inférence de laskey-

Input: Liste des instances de nœud requêtes (Q)

Liste des instances des nœuds d'observation (E)

Output: Un SSBN **B**

1 Étape 1 :(Initialisation)

- Créer un ensemble (R_0) contenant l'union des instances des nœuds requêtes et des instances des nœuds d'observation .
- Initialiser i à 1.
- Initialiser **MaxIter** par le nombre d'itérations maximal.
- Les nœuds de R_1 sont marqués comme non-terminé.

Étape 2 :(Construction de la structure)

if $i = \text{MaxIter}$ **ou** R_i **est vide** **then**

└ Passer à l'étape 3

else

 créer un ensemble vide R_{i+1} .

 Pour chaque nœud x (non-terminé) de cet ensemble R_i , effectuez les étapes ci-dessous.

- Récupérer le MFrag **Mx**, dans lequel x est un nœud résident ;
- Instancier les variables ordinaires de **Mx** en utilisant les valeurs d'argument du nœud x . Puis, évaluer les nœuds de contexte de **Mx**.
- Instancier les parents du nœud x dans **Mx**. Les nœuds parents doivent être des nœuds d'entrée ou des nœuds résidents. Les valeurs des variables ordinaires des nouveaux nœuds sont définies à partir des valeurs des variables ordinaires du nœud enfant x ou des valeurs récupérées par les nœuds de contexte pour les variables ordinaires non présentes dans x . Les nouveaux nœuds sont marqués comme **non-terminé** et ajoutés à l'ensemble R_{i+1} .
- Marquer le nœud comme **terminé** ;

Étape3 : (Élagage de la structure)

- Dans le SSBN, supprimer les nœuds qui n'influencent pas les distributions probabilistes des nœuds de requête étant donné les nœuds d'observation.

Étape4 : (Génération de la distribution probabiliste)

- Construire les CPTs pour les nœuds du SSBN. La CPT de chaque nœud x est générée à partir du LPD en fonction des parents et des états du nœud x .

Étape 5 : (Compilation et initialisation du SSBN généré)

- Appliquer un algorithmme d'inférence de réseau bayésien standard pour calculer la distribution conditionnelle des instances de nœud requêtes en se basant sur des instances de nœud d'observation.
-

Exemple 2.5.4 On suppose qu'on a trois régions stockées dans la base de connaissances, « Assia », « Austoralia » et « Africa » et un patient « Patient 1 ». Les observations pour chaque variable sont illustrées dans le tableau 2.1.

Le SSBN généré en utilisant l'algorithme de laskey pour l'entité « Patient1 » par l'instanciation de la MThéorie présentée dans la figure 2.4 est illustré dans la figure 2.6 :

Nœuds résidents	Variables	Observations
hasVisitedRegion()	Patient1, Asia	True
	Patient1, Australia	True
	Patient1, Africa	True
hasFluEpidemicPresent()	Asia	True
hasFluEpidemicPresent()	Africa	False

TABLE 2.1 – Les observations stockées dans la base de connaissances.

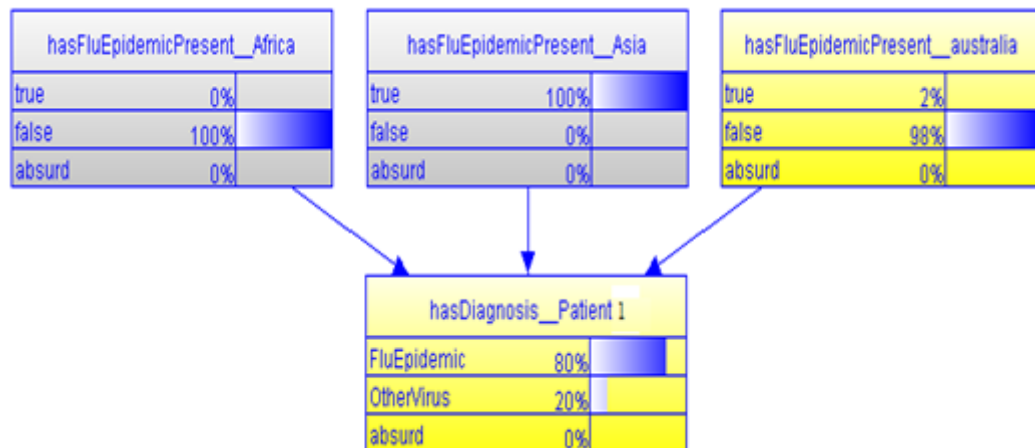


FIGURE 2.6 – Le SSBN généré pour Patient1

D'après le SSBN illustré dans la figure au-dessus, nous constatons que le patient 1 est touché par une épidémie de grippe avec une probabilité de 80%.

Afin d'étendre les MEBNs en utilisant la logique floue, les auteurs dans [77, 78] ont tenté d'améliorer le MEBN classique pour prendre en considération la connaissance floue. En effet, ils ont remplacé la logique du premier ordre par la logique du premier ordre floue (FFOL : Fuzzy First Order Logic) lors de la définition des contraintes contextuelles et sémantiques. Ils ont élargi ainsi la définition de MFrag classique en ajoutant des règles floues « If-Then-Else ». Dans cette nouvelle extension les MFrag flous ont été légèrement modifiés et définis comme **6-Tuple F** = (C, I, R, G, D, S). Où :

- C représente ensemble de nœuds contextuels représentent les structures sémantiques de la connaissance en utilisant des phrases de la logique floue du premier ordre,

- **R** représente ensemble de nœuds résidents,
- **I** représente ensemble de nœuds d'entrées liés à d'autres MFrag flou,
- **G** représente un graphe MFrag flou,
- **D** représente un ensemble de distributions locales pour chaque nœud résident,
- **S** représente un ensemble de règles « If-Then-Else » utilisées par un système d'inférence floue (FIS : Fuzzy Inference System).

Les théories (Fuzzy-MTheory) ont été définies comme un ensemble de MFrag flous pour lesquels les contraintes de cohérence sont satisfaites. Le processus d'inférence dans le MEBN flou est également abordé par les auteurs qui consiste à générer un SSFBN (SSFBN : Situation Specific Fuzzy Bayesian Networks) dans la première étape. En outre, les auteurs ont présenté une nouvelle version de l'algorithme de l'arbre de jonction pour traiter l'inférence probabiliste floue dans les SSFBN.

Ainsi, l'algorithme de l'arbre de jonction modifié pour la propagation des observations floues est basé sur la méthode de pondération. Les auteurs ont présenté des formules pondérées utilisant à la fois les degrés d'appartenance et les distributions de probabilité. Cependant, le fait que l'inférence est basée sur des formules pondérées et l'algorithme de l'arbre de jonction a été modifié peut présenter des complications si la complexité du problème traité augmente, car l'algorithme de l'arbre de jonction appartient à la classe des algorithmes d'inférence exacte qui est un problème NP-Difficile [72].

2.6 Réseaux Bayésiens Flous

Les Réseaux Bayésiens Flous (RBFs) combinent les capacités des réseaux bayésiens et de la logique floue pour bénéficier des avantages des deux modèles à la fois. En effet, les RBFs sont des modèles hybrides combinant à la fois la théorie des graphes, la théorie des probabilités et la logique floue pour traiter des connaissances floues liées à des expériences statistiques et à la fréquence d'occurrence d'un événement.

D'une part, les réseaux bayésiens sont généralement utilisés pour déterminer la probabilité d'un événement basé sur les observations de diverses variables. Néanmoins, ils ne peuvent pas faire des inférences probabilistes lorsque les observations sont floues.

D'autre part, la logique floue présente de nombreux avantages. Elle est considérée comme le modèle le plus approprié pour faire face à des connaissances floues liées au contenu d'une information. Elle permet de représenter l'appartenance progressive à un ensemble flou ainsi que le raisonnement humain dont les règles de décisions peuvent être représentées et déclarées en langage naturel.

Par conséquent, un RBF est une extension de réseau bayésien permettant de traiter les probabilités des événements sous des observations floues. Il se compose de deux types de nœuds ; les nœuds précis sont les nœuds dont la signification est précise et les nœuds flous sont les

nœuds dont la signification est vague. Comme nous l'avons mentionné précédemment, les RBFs ont montré récemment leur efficacité dans de nombreuses tâches difficiles et des problèmes du monde réel. Parmi ces travaux, nous citons :

- La classification bayésienne [79];
- L'analyse des risques [80, 81, 82, 83];
- Systèmes de recommandations [84].

La seule similarité entre les deux approches c'est que les deux sont dédiées pour faire face à des imperfections sur des données et des connaissances. Ainsi, leurs valeurs (connaissances quantitatives) sont représentées par des nombres réels dans l'intervalle $[0,1]$. Cependant, la combinaison de la théorie des probabilités et la logique floue est difficile et elle n'est pas si claire. Ceci est dû au fait que la sémantique de la valeur d'appartenance et la mesure de probabilité sont très différentes.

En effet, la logique floue est interprétée par un degré d'appartenance de l'état à un ensemble. Par contre, la mesure de probabilité est interprétée comme une croyance d'une personne ou une connaissance extraite à partir de l'occurrence d'un évènement. Néanmoins, de nombreuses méthodes différentes ont été proposées dans la littérature afin d'incorporer les degrés d'appartenances avec les probabilités lors de l'inférence probabiliste. Le tableau 2.2 synthétise les similarités et les différences entre la théorie des probabilités et la logique floue [85].

	Probabilités	Logique floue
Similarités	$0 \leq P(A) \leq 1$	$0 \leq \mu_A(x) \leq 1$
Différences	-Modélise l'incertitude subjective et l'incertitude objective. * $A \cup \neg A = \Omega$ i.e. $P(A \cup \neg A) = 1$. * $A \cap \neg A = \emptyset$ i.e. $P(A \cap \neg A) = 0$.	-Modélise l'imprécision de données. * $A \cup \neg A = \Omega$ N'implique pas $\mu_{A \cup \neg A}(x) \neq 0$ * $A \cap \neg A = \emptyset$ N'implique pas $\mu_{A \cap \neg A}(x) \neq 1$.

TABLE 2.2 – Comparaison entre la théorie des probabilités et la logique floue.

L'inférence dans les réseaux bayésiens classiques est basée sur des observations certaines. Pour chaque nœud observé, un seul de ses états doit être observé et avec certitude de 100%. Contrairement aux réseaux bayésiens classiques, les réseaux bayésiens flous permettent d'inférer des nouvelles connaissances en se basant sur un raisonnement probabiliste flou. Lorsqu'un nœud est flou, plusieurs états peuvent être observés à la fois avec des degrés d'appartenances graduels. Toutefois, il n'existe pas un modèle unique définissant les réseaux bayésiens flous. Les trois méthodes principales pour faire une inférence probabiliste floue sont les suivantes : **la méthode de pondération, la méthode de l'évidence virtuelle et la méthode distribution probabiliste floue.**

Exemple 2.6.5 . Pour mettre en évidence les trois approches, nous présentons dans ce qui suit un exemple qui sera utilisé modélisant la relation de causalité entre l'hémoglobine (Hb) et l'anémie (A). L'anémie est une condition dans laquelle il y a un manque de globules rouges saines

dans le corps pour transporter l'oxygène adéquat aux tissus du corps. Avoir de l'anémie peut rendre le patient fatigué et faible. En outre, l'hémoglobine est une métalloprotéine contenant du fer. Son rôle est de transporter l'oxygène (O₂) depuis l'appareil respiratoire (poumons, branchies) vers le reste de l'organisme.

Le graphe RBF qui correspond à cette situation est illustré dans la figure 2.7.

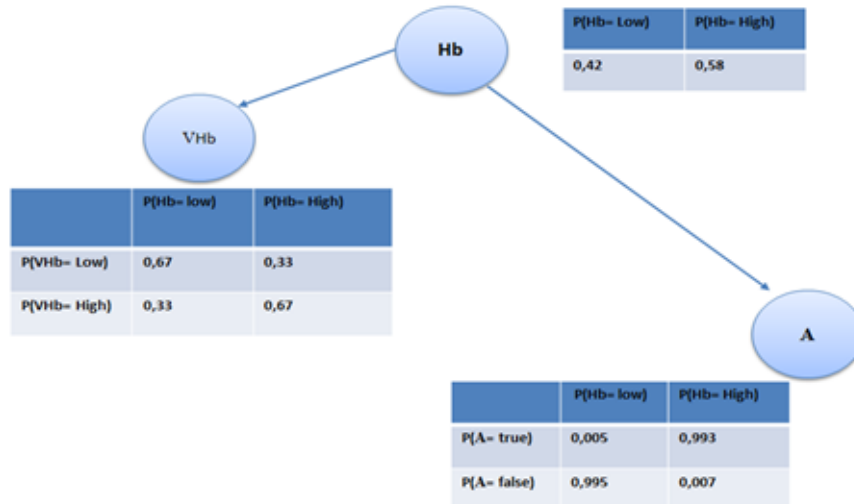


FIGURE 2.7 – Exemple d'un RBF.

Dans notre exemple, nous allons considérer la variable Hb comme floue, c'est pour cette raison, nous allons fuzzifier ses états, i.e., attribuer une fonction d'appartenance à chacun de ces états. Le tableau suivant illustre cette étape.

Etat	Interval	Type et paramètres de la fonction
lowHb	[0 – 13] gms	Trapezoid("low",0,0,10,16)
highHb	[13 – 20] gms	Trapezoid("high",10,16,20,20)

TABLE 2.3 – La fuzzification de la variable Hb.

On suppose que la valeur d'hémoglobine d'un patient P1 soit mesurée à 14 gms, les valeurs d'appartenances à chaque sous-ensemble flou comme il est montré dans la figure 2.8, sont définies comme suit :

$$\mu_{low}(14) = 0.67$$

$$\mu_{high}(14) = 0.33.$$

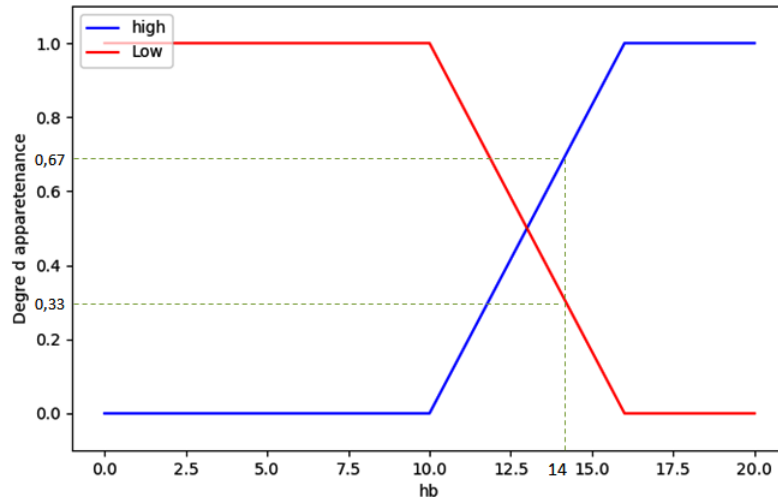


FIGURE 2.8 – Représentation graphique des fonctions d'appartenance.

Supposons que l'on cherche à calculer la requête suivante (Requête1) :

$$P(A = false | Hb)$$

Cette requête représente la probabilité que le patient P1 ne souffre pas de l'anémie sachant l'hémoglobine est low avec un degré de 0.67 et high avec un degré de 0.33.

2.6.1 Méthode de pondération

La méthode pondérée a été proposée dans [85, 86], l'idée principale de cette méthode est d'étendre les différentes règles utilisées dans les réseaux bayésiens en associant une valeur de degré d'appartenance à chaque loi ; cette méthode consiste à étendre la loi de Bayes afin d'incorporer les degrés d'appartenances dans les probabilités comme des poids lors de l'inférence. En effet, les règles bayésiennes floues peuvent ensuite être définies pour prendre en charge l'inférence bayésienne floue dans un modèle RBF.

Ainsi, dans [86] les auteurs ont distingué trois cas possibles qui sont :

- Dans le premier cas, le nœud de requête est flou,
- Dans le deuxième cas, le nœud d'observation est flou,
- Dans le troisième cas, le nœud d'observation et le nœud de requête sont flous.

Enfin, chaque règle bayésienne floue utilise la loi de la marginalisation floue.

Soient A et B deux variables aléatoires. Dans le cas où la variable A est floue la loi de Bayes pondérée est définie comme suite :

$$P(A | B_j) = \frac{\sum_{i \in I} \mu_A(A_i) P(B_j | A_i) P(A_i)}{P(B_j)} \quad (2.17)$$

Où, $\mu_A(A_i)$ représente le degré d'appartenance de l'élément à l'état i et B_j représente la variable B à l'état j.

Dans le cas où la variable B est floue ; la loi de Bayes pondérée est définie comme suite :

$$P(A_j | B) = \frac{\sum_{i \in I} \mu_B(B_i) P(B_i | A_j) P(A_j)}{P(B)} \quad (2.18)$$

Où, $\mu_B(B_i)$ représente le degré d'appartenance de l'élément à l'état i et A_j représente la variable A à l'état j.

En outre, la loi de probabilité marginale de la variable floue A est donnée comme suite :

$$P(A) = \sum_{i \in I} \mu_A(A_i) P(A_i) \quad (2.19)$$

Où, $\mu_A(A_i)$ représente le degré d'appartenance de l'élément à l'état A_i .

Dans le cas où les deux variables sont floues :

$$P(A | B) = \frac{\sum_{i \in I} \sum_{j \in J} \mu_A(A_i) \mu_B(B_j) P(B_j | A_i) P(A_i)}{P(B)} \quad (2.20)$$

Où, $\mu_A(A_i)$ représente le degré d'appartenance de l'élément à l'état i et $\mu_B(B_j)$ représente le degré d'appartenance de l'élément à l'état j.

La Requête 1 sera traitée utilisant la méthode de pondération proposée dans [86] comme suit :

$$\begin{aligned} P(A = false | Hb) &= \frac{\sum_{Hb_i \in I} \mu_{Hb_i}(14) P(Hb = Hb_i | A = false) P(A = false)}{P(Hb)} \\ &= \frac{\mu_{low}(14) P(Hb = low | A = false) P(A = false)}{P(Hb)} \\ &\quad + \frac{\mu_{high}(14) P(Hb = high | A = false) P(A = false)}{P(Hb)} \\ &= \frac{0.67 * 0.995 * 0.42 + 0.33 * 0.007 * 0.42}{0.7} \\ &= 0.60 \end{aligned} \quad (2.21)$$

Où,

$$\begin{aligned} P(A = false) &= P(A = false | Hb = low) P(Hb = low) \\ &\quad + P(A = false | Hb = high) P(Hb = high). \\ &= 0.995 * 0.42 + 0.007 * 0.58 \\ &= 0.42. \end{aligned} \quad (2.22)$$

$$\begin{aligned}
P(Hb) &= \mu_{low}(14) * P(Hb = low) + \mu_{high}(14) * P(Hb = high) \\
&= 0.67 * 0.42 + 0.33 * 0.58 \\
&= 0.47.
\end{aligned} \tag{2.23}$$

2.6.2 Distribution probabiliste floue

Cette méthode consiste à construire des distributions probabilistes floues [87, 88]. Les degrés d'appartenances seront directement intégrés dans les distributions des probabilités pour construire des distributions probabilistes floues. Ainsi, cette méthode est fondée sur deux étapes, la première étape sert à calculer les degrés d'appartenances et les représenter en notation particulière (par des vecteurs), lors de la deuxième étape, les degrés d'appartenances calculés en première étape seront incorporés directement dans les probabilités conditionnelles pour répondre à des requêtes.

Pour répondre à la *Requête 1* en utilisant la méthode des distributions probabilistes floues, nous devons tout d'abord représenter les degrés d'appartenances relatifs au nœud Hb comme un vecteur.

$$\begin{aligned}
Hb &= [low_{\mu_{low}(14)}, high_{\mu_{high}(14)}] \\
&= [low_{0.67}, high_{0.33}]
\end{aligned} \tag{2.24}$$

Par la suite, il s'agit d'incorporer les degrés d'appartenances dans la CPT du nœud A.

$$\begin{aligned}
A &= [\{P(A = true | Hb = high), P(A = false | Hb = high)\} \mu_{high}(14), \\
&\quad \{P(A = true | Hb = low), P(A = false | Hb = low)\} \mu_{low}(14)] \\
&= [\{true_{0.93}, false_{0.007}\}_{0.33}, \{true_{0.005}, false_{0.995}\}_{0.67}] \\
&= [true_{0.93*0.33+0.005*0.67}, false_{0.995*0.67+0.007*0.33}] \\
&= [true_{0.31}, false_{0.69}].
\end{aligned} \tag{2.25}$$

Nous avons obtenu à la fin une distribution probabiliste floue pour la variable A en se basant sur les observations de la variable Hb. A partir de cette distribution, nous pouvons constater que $P(A = false | Hb) = 0.69$ et $P(A = true | Hb) = 0.31$.

2.6.3 Méthode de l'évidence virtuelle

La méthode de l'évidence virtuelle a été proposée dans [89], afin d'incorporer des connaissances externes telle que l'incertitude associée aux observations dans des réseaux bayésiens comme il est mentionné dans [90, 91]. Par ailleurs, cette technique a été utilisée dans [92] pour incorporer des degrés d'appartenances dans un réseau bayésien. Elle consiste à ajouter, pour

chaque nœud flou dans le RB, un nœud fils appelé « nœud virtuel » qui sert à stocker les observations incertaines ou floues. En effet, les degrés d'appartenance seront mis dans la CPT du nœud virtuel et seront incorporées avec les probabilités conditionnelles comme des poids lors de l'inférence bayésienne en appliquant un algorithme d'apprentissage standard.

Pour répondre à la Requête 1 en utilisant la méthode d'évidence virtuelle, nous devons ajouter un nœud virtuel nommé VHb dans le réseau comme un nœud fils du nœud Hb, il contient les degrés d'appartenance relatifs à l'observation de la Requête 1. La nouvelle structure du RB est illustrée dans la figure 2.9.

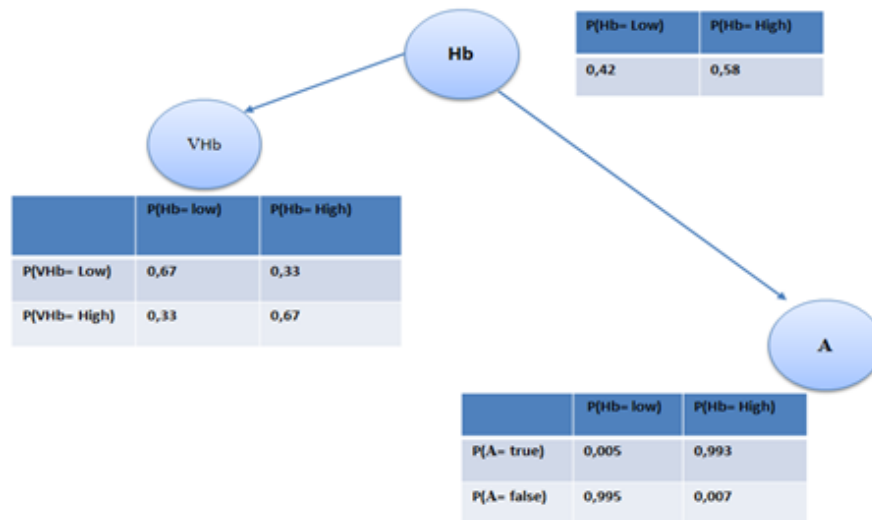


FIGURE 2.9 – La nouvelle structure du RBF avec le nœud virtuel VHb.

Pour calculer la requête $P(A=false|Hb)$, nous devons remplacer le nœud d'observation Hb par son nœud virtuel VHb. Dans ce cas la requête devient :

$$\begin{aligned}
 P(A = false|VHb = low) &= \frac{P(VHb = low | A = false)P(A = false)}{P(VHb = low)} \\
 &= \frac{0,67 * 0,42}{0,47} \\
 &= 0,60.
 \end{aligned}
 \tag{2.26}$$

$$\begin{aligned}
 P(VHb = low) &= P(VHb = low | Hb = low)P(Hb = low) \\
 &\quad + P(VHb = low | Hb = high)P(Hb = high) \\
 &= 0,67 * 0,42 + 0,33 * 0,58 \\
 &= 0,47.
 \end{aligned}
 \tag{2.27}$$

Il est à noter que la méthode de l'évidence virtuelle et la méthode de pondération ont données le même résultat.

2.6.4 Bilan sur les méthodes d'inférence probabiliste floue

Dans les sections précédentes nous avons présenté les trois méthodes principales pour incorporer les connaissances floues dans l'inférence bayésienne. En effet, la première méthode consiste à incorporer les observations floues dans les probabilités comme des poids. Cette méthode semble simple dans la façon de combiner les probabilités et les valeurs d'appartenance floues. Néanmoins, elle présente plusieurs inconvénients, notamment : le fait que les différentes formules ont été modifiées, en particulier, la formule de Bayes, les algorithmes d'inférences dans les réseaux bayésiens doivent également être modifiés et mis à jour. Car ces algorithmes sont basés sur les formules bayésiennes standard. Par conséquent, l'inférence ne peut être effectuée qu'avec un algorithme d'inférence personnalisé. Cette limite peut poser des complications lorsque la complexité du problème augmente car on ne peut pas changer le type d'algorithme facilement si nous sommes confrontés à un problème qui nécessite un algorithme d'inférence approché.

La méthode des distributions probabilistes floues intègre les degrés d'appartenances directement dans les distributions probabilistes. Cependant, cette méthode présente un inconvénient majeur c'est qu'elle peut donner une explosion de complexité ceci est dû à l'énumération de tous les états possibles pour toutes les variables impliquées dans la requête [87]. Par ailleurs, la méthode de l'évidence virtuelle vise à incorporer les degrés d'appartenances lors de l'inférence bayésienne comme des observations floues.

Comme nous l'avons remarqué dans l'exemple illustratif, ses résultats sont similaires à la méthode pondérée présentée en haut et cela sans modifier les différentes formules bayésiennes. Le fait que la formule de Bayes n'ait pas été modifiée dans cette méthode cela présente plusieurs avantages, notamment, il n'y a pas une nécessité de modifier les algorithmes d'inférences classiques. En effet, nous pouvons les utiliser tels qu'ils sont définis. Le choix d'un algorithme d'inférence par la suite dépend de la complexité du problème et la contribution du développeur. Dans le cas où nous sommes confrontés à un problème complexe il est préférable d'utiliser un algorithme d'inférence approché.

2.7 Discussion

Ce chapitre vise à explorer les formalismes de traitement de l'incertitude, notamment, la logique floue et la théorie des probabilités en particulier les RBs.

En effet, la logique floue et la théorie des probabilités permettent de quantifier l'incertitude comme des poids positifs à des événements.

La logique floue est proposée pour traiter la connaissance floue due à l'imprécision de données en se basant sur la notion d'appartenance graduel des éléments.

En effet, les RBs ont un rôle central pour la représentation des connaissances incertaines. Les RBs permettent de représenter les connaissances relatives à un domaine précis à l'aide d'un graphe dirigé acyclique, la représentation graphique de la connaissance fournie par les réseaux

bayésiens est aussi un avantage pour les systèmes de raisonnement à base incertaine dans divers domaines notamment le Web sémantique. Ainsi, les RBs présentent plusieurs avantages, parmi lesquels, nous citons :

- La possibilité de consolider des connaissances de diverses natures et sources dans un même modèle à savoir : des connaissances extraites à partir de données historiques ou empiriques (des probabilités objectives), l'expertise d'un expert de domaine qui est exprimée sous forme de règles stochastiques ou de probabilités subjectives et enfin l'incorporation de nouvelles observations fournit par un observateur dans le système afin de mettre à jour l'état de croyances des variables du modèle.
- La représentation graphique d'un réseau bayésien est compréhensible par des non spécialistes, elle est également définie de manière explicite ce qui facilite son exploitation pour l'aide à la décision, sa vérification et son évolution, ceci est dû au fait que le décideur peut bien visualiser le modèle et comprendre son fonctionnement.
- Le même modèle peut être déployé pour plusieurs objectifs, entre autres : pour l'évaluation, la prévision.

En revanche, la logique floue est considérée maintenant comme un modèle pour le traitement des problèmes avec des affirmations floues, elle a été appliquée dans plusieurs domaines, son succès est dû à plusieurs raisons telles que :

- La simplicité et la flexibilité.
- La gestion de problèmes avec des connaissances imprécises et incomplètes.
- La facilité de modéliser le raisonnement humain où elle permet de modéliser des règles proches du langage naturel.

Le besoin de représenter certains aspects dans les réseaux bayésiens standards a donné naissance à plusieurs extensions pour améliorer leur niveau de modélisation et leurs performances de raisonnement. Parmi ces modèles, nous citons :

- **Réseaux Bayésiens Multi-Entités** : cette extension vise à augmenter l'expressivité des RBs standards en utilisant la logique du premier ordre pour les rendre flexibles et afin de faire face aux problèmes avec des sous structures répétitives.
- **Réseaux bayésiens flous** : cette extension permet de traiter les variables floues dans le réseau bayésien, elle permet également de faire une inférence probabiliste basée sur des observations floues. Cette extension est efficace surtout dans le cas où le problème à modéliser présente des connaissances floues liées à des connaissances probabilistes.

Il est à noter que la liste des extensions présentée dans ce chapitre n'est pas exhaustive et qu'il existe d'autres extensions développées dans la littérature.

2.8 Conclusion

Dans ce chapitre, nous avons donné un aperçu sur les modèles de gestion de la connaissance incertaine, notamment, la logique floue la théorie de probabilité en mettant l'accent sur les modèles graphiques probabilistes.

Nous avons commencé par explorer la logique floue et les fondements théoriques de cette logique ainsi que sa sémantique.

Nous avons donné un aperçu sur la théorie des probabilités à savoir les probabilités conditionnelles et bayésiennes, nous avons aussi présenté les interprétations de la mesure de probabilité. Ensuite, nous avons exposé les RBs avec leurs caractéristiques, leurs méthodes d'apprentissages à savoir l'apprentissage de la structure et l'apprentissage de paramètres et les algorithmes dédiés pour faire l'inférence probabiliste. En outre, une section très importante de ce chapitre a été consacrée pour la présentation de quelques extensions des RBs, notamment, les MEBNs.

Dans la dernière partie, nous avons présenté les notions de base liées aux réseaux bayésiens flous comme un modèle combinant les RBs et la logique floue pour traiter les connaissances probabilistes et floues simultanément. En effet, les trois principales approches ont été présentées dans ce chapitre.

Chapitre 3

Modélisation et raisonnement sur les connaissances incertaines

“ Le doute est le commencement de la sagesse . ”

— — Aristote

Sommaire

3.1	Introduction	54
3.2	Représentation des connaissances probabilistes	55
3.2.1	Logiques des descriptions probabilistes	56
3.2.2	Extensions probabilistes du langage RDF	56
3.2.3	Extensions probabilistes du langage OWL	57
3.3	Représentation des connaissances floues	61
3.3.1	Définitions des ontologies floues	61
3.3.2	Logiques de description floues	62
3.3.3	Extensions floues de OWL	62
3.3.4	Extension floue de RDF	63
3.3.5	Raisonnement sur des connaissances floues	63
3.4	Synthèse et comparaison	64
3.4.1	Discussion sur les ontologies probabilistes	66
3.4.2	Discussion sur les ontologies floues	69
3.5	Conclusion	72

3.1 Introduction

Les informations véhiculées dans le Web sémantique sont susceptibles d’être incertaines. En effet, le Web sémantique repose principalement sur les capacités offertes par les ontologies pour modéliser et faire des raisonnements avec les connaissances. Néanmoins, les ontologies

classiques sont considérées inefficaces quand le domaine à traiter présente de l'incertitude. Elles ne peuvent pas modéliser et représenter formellement la connaissance incertaine qui peut se présenter sous plusieurs formes à savoir l'incomplétude, l'aléatoire et l'imprécision, etc. Cela a conduit la communauté du Web sémantique à trouver des alternatives quant à la représentation et le raisonnement avec l'incertitude

Différentes approches ont été développées dans la littérature et elles ont été identifiées comme étant des extensions des ontologies classiques. Ces nouvelles approches permettant la modélisation et le raisonnement avec l'incertitude. Elles peuvent être classées dans deux principales classes : les approches probabilistes et celles basées sur la logique floue. Dans ce troisième chapitre, nous présentons ces deux classes d'approches.

3.2 Représentation des connaissances probabilistes

Il existe trois classes d'approches probabilistes. La classification se fait selon le type de formalisme utilisé, à savoir, les logiques de description, le langage RDF et les ontologies en OWL.

Les ontologies probabilistes sont utilisées pour décrire explicitement les connaissances incertaines de manière formelle et sophistiquée. En outre, elles visent à élargir les capacités d'ontologies standards en fournissant des mécanismes permettant de représenter des connaissances probabilistes. Ceci en se basant sur les modèles probabilistes notamment les RBs. Ainsi, elles peuvent également bénéficier des moyens que fournissent les RBs telle que l'inférence probabiliste.

Nous présentons dans les sections suivantes les approches probabilistes par catégorie à savoir *les logiques de description probabilistes, les extensions probabilistes du langage RDF et les extensions probabilistes du langage OWL.*

Définition 3.2.3 (Ontologie probabiliste). *Selon la définition donnée dans [93], une ontologie probabiliste est une représentation de la connaissance formelle explicite qui exprime des connaissances sur un domaine d'application. Elle implique :*

- *Les types d'entités qui existent dans le domaine,*
- *Les propriétés de ces entités,*
- *Les relations entre ces entités.*
- *Les processus et les événements qui se passent avec ces entités,*
- *Les régularités statistiques qui caractérisent le domaine,*
- *Les connaissances peu concluantes, ambiguës, incomplètes, peu fiables, et dissonantes liées aux entités du domaine,*
- *L'incertitude sur toutes les formes de connaissance citées ci-dessus,*

3.2.1 Logiques des descriptions probabilistes

Dans [94], une extension du *DL – classic* nommée *P – Classic* est proposée permettant de représenter des connaissances terminologiques et les composants probabilistes du domaine (des propriétés, des individus, etc). Ceci est fait à l’aide de nouveaux constructeurs à savoir les classes probabilistes *P – classe*.

Une DL probabiliste nommée *Creedal* a été proposée dans [95]. Elle étend la logique ALC afin de traiter l’incertitude. En effet, de nouveaux constructeurs ont été ajoutés pour permettre à l’*ALC* de traiter les connaissances probabilistes en représentant un ensemble de contraintes telles que les contraintes conditionnelles $P(A | B) = a$ et $P(r) = b$,. Où, A et B représentent des classes, a et b représentent des probabilités et r représente un rôle.

P – SHOQ [96] étend le *SHOQ* classique en ajoutant les constructeurs nécessaires pour représenter un ensemble de contraintes telles que $P(A | B) = a$, $a \in [c, d]$ et $[c, d] \subset [0, 1]$, . Où, A et B représentent deux concepts.

Une extension probabiliste de la logique de description *EL* a été proposée dans [97] et nommée *BEL* basée sur les RBs. En effet, dans [98, 99], les auteurs ont mis en œuvre un raisonneur bayésien sur les ontologies DL basé sur la sémantique de *BEL*, nommé BORN. Une caractéristique clé de *BORN* est qu’il fournit un moyen pour représenter une subsumption probabiliste de *BEL*.

3.2.2 Extensions probabilistes du langage RDF

Le langage RDF est un formalisme pour la représentation des connaissances qui s’est rapidement imposé comme un modèle de données. Il permet de représenter et annoter des informations sur les ressources de Web. Malgré sa popularité, RDF reste limité et ne fournit pas des alternatives pour exprimer des connaissances non déterministes, notamment les connaissances probabilistes. Dans cette partie, nous exposons, certaines extensions du langage RDF permettant la modélisation de la connaissance probabiliste.

Dans [100], une extension probabiliste du langage RDF a été proposée qui fournit un vocabulaire pour représenter les relations probabilistes dans RDF. Cette extension fournit également un protocole pour l’interrogation et l’inférence dans le RB. En outre, un algorithme pour transformer un ensemble de relations probabilistes d’un graphe RDF en un RB a été développé. La démarche adoptée dans ce système se résume dans les étapes suivantes

- Représenter les relations probabilistes dans un graphe RDF.
- Transformer le graphe RDF en réseau bayésien et le stocker dans un raisonneur bayésien.
- Transformer le graphe d’observation et l’envoyer au raisonneur.
- Formuler la requête et la transmettre au raisonneur, et
- Récupérer le résultat du raisonnement.

Une autre contribution intéressante a été développée dans [101], nommée pRDF (Probabilistic RDF). Les auteurs ont proposé une syntaxe ainsi qu'un modèle théorique et sémantique pour la représentation de la connaissance probabiliste dans RDF. Ils ont proposé également des algorithmes pour répondre efficacement aux requêtes posées aux ontologies pRDF.

3.2.3 Extensions probabilistes du langage OWL

Après l'étude détaillée des travaux existants dans la littérature, nous avons constaté que ces travaux se focalisent sur la combinaison des ontologies avec les réseaux bayésiens. Dans ce qui suit nous donnons un aperçu de ces travaux.

3.2.3.1 Travaux de recherche de Ding et al., 2006

Dans [1], les auteurs ont proposé une extension probabiliste à OWL, appelé BayesOWL. Elle est basée sur les réseaux bayésiens standards. En effet, BayesOWL étend les capacités de « OWL » pour la modélisation et le raisonnement avec l'incertitude. Il applique un ensemble de règles pour transformer la hiérarchie de classes (taxonomie) définie dans une ontologie OWL en un réseau bayésien. Le réseau bayésien généré est constitué de deux types de nœuds à savoir : les *nœuds de concepts* et les *LNœuds*. Les *nœuds de concepts* représentent les concepts de l'ontologie qui sont connectés entre eux par des arcs dirigés qui relient les classes mères avec les classes filles. Quant aux *LNœuds* sont un type particulier de nœuds utilisés pour modéliser les relations définies par owl : « owl :intersectionOf, owl :unionOf, owl :complementOf, owl :equivalentClass ; owl :disjointWith ».

BayesOWL présente plusieurs limites, notamment le fait qu'il tient compte seulement la connaissance probabiliste liée aux classes et les relations d'héritage entre elles. Cependant, dans une ontologie on peut distinguer d'autres types de relations non hiérarchiques.

3.2.3.2 Travaux de recherche de Yang et Calmet, 2005

Une autre approche présentée dans [2], nommée OntoBayes. Elle est basée sur les annotations pour la modélisation des probabilités à priori et conditionnelles. Cela est fait en représentant des distributions des probabilités via les classes *FullProbDist*, *PriorProb* et *CondProb* (voir la figure 3.1). Ainsi, les dépendances entre les nœuds sont décrites par les propriétés rdfs : *DependsOn*. En effet, OntoBayes pallie à la limite de BayesOWL et tient compte des propriétés OWL pour la modélisation. Chaque variable aléatoire dans le RB peut être également une propriété de données ou une propriété d'objets.

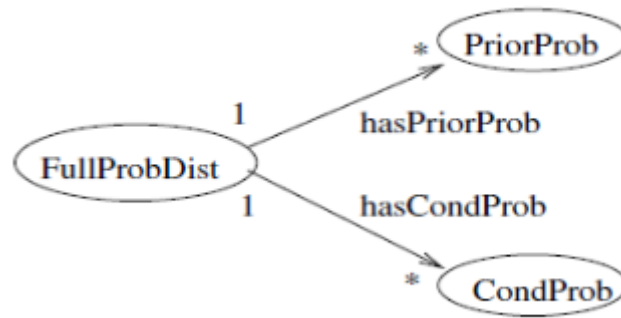


FIGURE 3.1 – L’ontologie de haut-niveau de OntoBayes [2].

3.2.3.3 Travaux de recherche de Costa et al., 2005 ; Carvalho et al., 2017

Les auteurs dans [3] ont proposé un langage probabiliste nommé PR-OWL comme une extension. Il permet de représenter la connaissance probabiliste impliquée dans une ontologie formalisée en OWL. Outre que les caractéristiques trouvées dans le langage d’ontologies OWL, PR-OWL tient compte des régularités statistiques des domaines en se basant sur le MEBNs. Il permet de déduire de nouvelles formes de connaissances par inférence avec une prise en compte de l’incertitude liées à ces connaissances.

PR-OWL est très flexible et expressif du fait qu’il est basé sur la logique de premier ordre bayésienne. PR-OWL permet une représentation formelle et explicite des connaissances déterministes et des connaissances non déterministes, notamment les connaissances probabilistes de domaine étudié. En effet, PR-OWL représente la sémantique de MEBN à l’aide d’une ontologie de haut-niveau. Cette ontologie contient des classes et des propriétés pour représenter formellement les éléments de MEBN, à savoir : les MTheories, les MFrag, les noeuds, etc.

Récemment, une version améliorée de PR-OWL [4] appelée PR-OWL2.0 a été développée. Elle est bien compatible avec le langage OWL pour la représentation des correspondances entre les propriétés en OWL et les nœuds en PR-OWL et les états de nœuds avec les types déjà définis en OWL.

3.2.3.4 Travaux de recherche de Carvalho et al., 2016

Les auteurs dans [102] ont proposé une méthodologie pour la construction des ontologies probabilistes. Elle est basée sur un processus itératif et incrémental, inespéré du processus unifié¹ (UP : **U**nified **P**rocess) pour la conception d’une ontologie probabiliste. Cette méthodologie nommée UMP-ST (UMP-ST : **U**ncertainty **M**odeling **P**rocess for **S**emantic **T**echnology) et ses étapes sont déclinées en actions ci-dessous présentées :

1. Le processus unifié fournit un cadre au développement logiciel itératif et incrémental pour la construction de systèmes selon le paradigme orienté objet.

- **La spécification des besoins** : dans cette étape le développeur doit spécifier explicitement les besoins derrière le développement d'une ontologie probabiliste. Il s'agit de définir les questions auxquelles l'ontologie probabiliste devrait répondre, c'est-à-dire, les requêtes à poser au système en cours de conception.
- **L'analyse et la conception** : l'objectif de cette étape est de décrire les classes d'entités, leurs attributs, les relations entre elles et les règles qui peuvent être déterministes ou stochastiques.
- **L'implémentation** : cette étape vise à implémenter l'ontologie dans un langage spécifique et formel.
- **Les tests** : au cours de cette étape, il est question de vérifier la validité de l'ontologie probabiliste.

3.2.3.5 Travaux de recherche de Zheng et al., 2016

Les auteurs dans [5], ont développé une extension à PR-OWL afin d'intégrer la logique floue dans PR-OWL. Les auteurs ont modifié l'ontologie de méta-modèle de PR-OWL afin de représenter la connaissance floue comme un ensemble de règles floues « If-Then-Else ». Pour ce faire, un ensemble de classes et de propriétés a été ajouté dans l'ontologie de haut-niveau de PR-OWL pour représenter la sémantique de la logique floue.

3.2.3.6 Travaux de recherche de Ishak et al., 2011

Dans [103], une nouvelle approche est proposée qui vise à construire des réseaux bayésiens orientés objets (OOBN :OOBN : Oriented Object Bayesian Network) à partir des ontologies existantes. Dans cette approche, les auteurs ont défini un ensemble de règles de mapping pour construire un OOBN basé sur une ontologie classique. En outre, plusieurs algorithmes ont été développés pour faire la génération d'un OOBN basée sur ces règles. En effet, un OOBN [104] est une application directe du paradigme de l'orienté objet dans les réseaux bayésiens standards. L'élément de base est la classe qui représente un fragment de réseau bayésien qui peut être instancié plusieurs fois pour créer des RBs standards. Les OOBNs permettent une représentation encore plus compacte des connaissances et surtout lorsque le RB contient des structures répétitives. Ils permettent également de modéliser des systèmes complexes pour lesquels le même mode de raisonnement apparaît dans différents sous-systèmes.

3.2.3.7 Travaux de recherche de Fenz ., 2012

Une autre approche présentée dans [105] se concentre sur la construction d'un réseaux bayésien en se basant sur une ontologie de domaine existante. Cette méthode comporte les étapes suivantes :

- **Sélection des classes, individus et propriétés** pertinentes par les experts du domaine.

- **Création de la structure** du réseau bayésien par les experts du domaine en se basant sur les composants sélectionnés dans l'étape précédente.
- **L'intégration des connaissances factuelles** existantes dans l'ontologie dans le RB et la construction des tables de probabilités conditionnelles préservant les contraintes sémantiques de l'ontologie.

Les caractéristiques clés de cette méthode est qu'elle intègre les connaissances factuelles existantes lors de la construction des CPTs, en particulier les connaissances probabilistes. En plus de ça , un plug-in protégé a été développé pour pouvoir exécuter les étapes de construction.

3.2.3.8 Travaux de recherche de Mouenis et al., 2014

Dans [106], une démarche a été proposée dont le but est de bénéficier des avantages de réseaux bayésiens standards et les ontologies à la fois. En effet, elle vise à intégrer un réseau bayésien qui est supposé déjà construit dans une ontologie OWL pour permettre à la connaissance incertaine d'être codée formellement. Pour ce faire, une méthodologie a été proposée pour transformer et encoder un réseau bayésien en ontologie. En effet, les auteurs ont proposé un ensemble de règles de mappage dans lesquelles les nœuds sont transformés en un ensemble de concepts dans le langage ontologique OWL. Les instances de chaque classe sont générées à partir des états de nœud correspondant. En outre, plusieurs propriétés ont été définies pour représenter les probabilités à priori telles que $P(A)$ et les probabilités conditionnelles telle que : $P(A | B)$.

3.2.3.9 Travaux de recherche de Emna et al., 2016a ; Emna et al., 2016b

Dans Emna et al., 2016a [6], les auteurs ont présenté une extension du méta-modèle OWL2 (ODM : Ontology Definition Meta-Model for OWL2) [107], appelée (PODM : Probabilistic Ontology Definition Meta-Model for OWL2) afin de fournir les moyens nécessaires à la construction d'ontologies probabilistes. Ceci est fait en ajoutant de nouveaux composants tels que classes probabilistes, individus probabilistes, propriétés de données probabilistes, etc. Ces composants permettent à l'ancien méta-modèle OWL2 de représenter la connaissance probabiliste. En outre, dans Emna et al., 2016b[7], les auteurs ont proposé une méthodologie afin de guider l'ontologiste à construire des ontologies probabilistes basées sur PODM.

3.2.3.10 Travaux de recherche de Mohammed et al., 2016

Un travail récent de [8], dans lequel les auteurs ont proposé une solution nommée ontologie probabiliste hybride. Elle est basée sur les réseaux bayésiens hybrides afin de gérer simultanément des composants discrets et continus dans l'ontologie. En effet, les auteurs ont défini un ensemble de classes et propriétés afin de construire un RB hybride et le représenter sémantiquement dans l'ontologie pour qu'elle soit probabiliste.

3.2.3.11 Travaux de recherche de Emna et al., 2018

Dans [108], les auteurs ont proposé une méthode récente pour la construction d'ontologies probabilistes. Ils ont distingué entre les composants précis et probabilistes de l'ontologie. Par conséquent, ils ont transformé uniquement les composants de l'ontologie qui peuvent être attachés par l'incertitude en un RB, notamment l'incertitude liée aux instances de l'ontologie.

3.3 Représentation des connaissances floues

Bien que les ontologies classiques ont été standardisées pour la modélisation et le raisonnement avec les connaissances, elles ne sont pas capables de représenter les connaissances floues. Pour combler cette lacune, de nombreuses extensions floues des ontologies classiques ont été développées. Elles combinent la logique floue avec les ontologies classiques. Dans cette section, nous allons donner un aperçu sur les approches proposées dans la littérature.

3.3.1 Définitions des ontologies floues

Comme les ontologies classiques, il n'existe pas une définition unique des ontologies floues. De nombreuses définitions pour les ontologies floues ont été distinguées. Certaines sont plus générales que d'autres. En effet, l'auteur dans (Bobillo, 2008) a défini l'ontologie floue comme une ontologie qui utilise la logique floue pour fournir une représentation naturelle de la connaissance vague et /ou imprécise, et facilite les raisonnements avec elle.

Dans [9], les auteurs ont défini une ontologie floue comme une 6- Tuple :
 $OF = \langle X, A, C, R_{XC}, R_{AC}, R_{CC} \rangle$, tel que :

- X est un ensemble d'objets.
- A est l'ensemble des attributs décrivant les objets.
- C est un ensemble de concepts (classes).
- **La relation floue** $R_{XC} : X * C \implies [0, 1]$, permet d'associer un degré d'appartenance à la paire (x_i, c_i) pour tout $x_i \in X$ et $c_i \in C$.
- **La relation floue** $R_{AC} : A * C \implies [0, 1]$, définit la correspondance entre l'ensemble des attributs A et l'ensemble des concepts ; et,
- Finalement, **la relation floue** $R_{CC} : C * C \implies [0, 1]$, définit l'imprécision entre des relations classe-filles / classes-mères entre l'ensemble des concepts C .

Dans [10], la définition d'une ontologie floue est donnée comme un 5-tuple
 $OF = \langle I, C, R, F, A \rangle$, tel que :

- I représente un ensemble d'instances de concepts,
- C représente un ensemble de concepts,
- R représente un ensemble de relation,

- **F** représente un ensemble de relations floues, et
- **A**, représente un ensemble d'axiomes.

Dans le reste de cette section, nous allons présenter les logiques de description floue, les extensions floues de OWL, les extensions floues de RDF et quelques raisonneurs sur des connaissances floues.

3.3.2 Logiques de description floues

Dans le cadre de la modélisation de l'imprécision dans les ontologies, plusieurs extensions de DLs ont été proposées dans la littérature, nous citons :

FuzzySROIQ(D) est une extension de *SROIQ(D)* (équivalant à OWL2) dans laquelle les concepts sont considérés comme des ensembles flous d'individus et les rôles sont représentés par des relations binaires floues. Les axiomes s'étendent également et certains d'entre eux sont vrais avec des degrés. *FuzzySROIQ(D)* [109] utilise des alphabets qui représentent les concepts, les individus et les rôles, tel que :

- Les concepts complexes sont notés par le symbole **C** et **A** dans le cas de concepts atomiques.
- Plusieurs types de rôles ont été pris en compte, entre autres : Les rôles abstraits qui sont notés par **R** et les rôles concrets par **T**. Ainsi, R_A représente l'ensemble des rôles atomiques abstraits, **R-** représente des rôles inverses, **S** représente un simple rôle. **U** représente le rôle universel (un rôle est universel si et seulement s'il est vraie pour chaque pair d'individus).
- Les auteurs ont fait une distinction entre les individus. En effet, les individus sont notés par **a**, **b** si l'individu est abstrait et **v** s'il est concret.

3.3.3 Extensions floues de OWL

Plusieurs langages flous du Web Sémantique ont été émergés pour donner vie aux ontologies floues. Nous allons présenter dans cette section un aperçu sur les différentes extensions des langages proposées dans la littérature pour faire face au flou dans les ontologies.

En effet, le langage OWL est un langage standardisé par le W3C pour la formalisation des ontologies, plusieurs chercheurs ont essayé d'améliorer les capacités de ce langage pour modéliser la connaissance floue dans le web sémantique. Parmi ces travaux nous citons :

Dans [11], une méthode pour la représentation des ontologies floue basée sur OWL2 est proposée, nommée *FuzzyOWL2*. Cette approche est basée sur les annotations pour la modélisation de la connaissance floue qu'on ne peut pas la représentée directement avec OWL2. *FuzzyOWL2* utilise des annotations supplémentaires « fuzzyLabel » pour représenter la sémantique de Fuzzy *SROIQ(D)*. Il est composé d'un ensemble de concepts flous, des individus flous, des rôles flous et des axiomes flous. En outre, cinq types de données peuvent être représentés en utilisant les

éléments de FuzzyOWL2 à savoir Trapèze, Triangular, Left-shoulder, Right-shoulder et la cinquième fonction est le résultat d'application d'un modificateur flou entre la fonction linéaire et triangulaire.

3.3.4 Extension floue de RDF

Dans [110], une extension de RDF est présentée, basée sur le triple RDF (sujet, Object, Prédicat) flou. Elle permet de modéliser des relations (triplets) floues. Son interprétation est : Le sujet « s » et l'objet « o » sont liés par la relation « R » avec un certain degré.

3.3.5 Raisonnement sur des connaissances floues

Le succès des ontologies floues dépend de la présence et de la disponibilité d'un raisonneur capable d'effectuer des raisonnements et des inférences sur des connaissances floues représentées dans les ontologies. Pour ce faire, plusieurs raisonneurs ont été proposés dans la littérature. Dans cette section nous donnons un aperçu sur les plus cités dans la littérature.

DeLorean [111, 112] est un raisonneur et / ou un traducteur qui fait la traduction d'une ontologie floue en une ontologie classique. Il manipule des connaissances représentées dans SROIQ(D) floue et SHOIN(D) floue. L'ontologie non-floue résultante à partir de cette traduction préserve la sémantique de la représentation floue initiale. Elle sera par la suite traitée par un DL raisonneur classique. La version courante de DeLorean supporte plusieurs autres tâches de raisonnements comme la vérification de consistance d'une base de connaissance, etc.

Fuzzy DL [113] est un raisonneur expressif pour la logique de *Description-SHIF*. Il permet aux utilisateurs de définir des composants ontologiques flous. La caractéristique clé de ce raisonneur est qu'il comporte plusieurs interfaces graphiques pour faciliter aux utilisateurs leurs tâches, à savoir : avec un interpréteur de commandes, Protégé plug-in, et Java API. Avec l'interpréteur des commandes, l'utilisateur peut manipuler le FuzzyDL à partir des commandes en ligne. Avec Protégé plug-in on peut développer facilement des ontologies floues OWL 2 et l'interroger avec des requêtes. Une API java est également disponible lorsque l'utilisateur veut utiliser les résultats de raisonnement dans ses applications.

FuzzyDL fournit plusieurs tâches de raisonnements telles que la vérification de consistance, la subsomption. Parmi lesquelles, nous citons :

- **La consistance de la Base de connaissances** : une base de connaissance **K** est consistante si et seulement s'il ya un modèle qui satisfait chaque axiome de cette base de connaissance.
- **La satisfiabilité des concepts** : un concept flou **C** est **D**-satisfiable dans une base de connaissance **K** si et seulement s'il existe un modèle de **K** dans lequel le concept **C** a possède au moins une instance avec un degré supérieure ou égal à **D**.
- **Meilleur degré de satisfiabilité** : cette tâche consiste à la recherche du degré maximal sachant que **C** est **D**-satisfiable dans **K**.

- **Le degré de satisfiabilité minimale** : cette tâche consiste à la recherche du degré minimal sachant que C est D -satisfiable dans K .
- **Subsomption de concepts** : C_2 D -subsume C_1 dans une base de connaissance K , si et seulement si dans chaque modèle de k , $C_2 \sqsubseteq C_1$ avec un degré supérieure ou égal à D .
- **Récupération des individus** : étant donné une base de connaissance K et un concept C . Cette tâche consiste à calculer le nombre d'instances appartenant à C avec un degré différent de zéro et avec un degré minimal dans chaque modèle de K .

3.4 Synthèse et comparaison

Dans cette section nous présentons une comparaison entre les travaux existants de la modélisation des connaissances probabilistes et floues dans les ontologies. Pour ce faire, nous utilisons les critères d'évaluation suivants :

- **Les modèles utilisés** : cette caractéristique permet de déterminer les modèles utilisés pour faire face à l'incertitude dans les ontologies, à savoir, les RBs standards, les RBs hybrides, les MEBNs, les OOBNs, etc.
- **Les inférences fournies** : cela représente les inférences fournies par chaque approche, à savoir, (a) l'inférence ontologique tel que la satisfiabilité de concepts, la subsomption de concepts, classification des individus, etc., (b) *l'inférence probabiliste* où il s'agit de calculer la probabilité d'un évènement à partir des observations sur quelques variables aléatoires, (c) *l'inférence ontologique floue* telles que la satisfiabilité floue d'un concept, classification floue des individus, etc., et (d) *l'inférence probabiliste floue* où il s'agit de calculer la probabilité d'un évènement à partir des observations floues sur d'autres variables.
- **Connaissances modélisées** : cette caractéristique permet de spécifier les types d'incertitude traitée par l'approche, à savoir, la connaissance floue et la connaissance probabiliste.
- **Formalisme utilisé** : cela représente la technologie utilisée par chaque approche, à savoir, *les LDs, le RDF et le langage OWL*.

En effet, le tableau illustré dans la figure 3.2 récapitule la comparaison entre les travaux les plus connexes à nos contributions.

Approches	Modèles utilisés										Technologie utilisée.			Connaissances modélisées			Inférences fournies		
	Théorie de probabilités	RB standard	RB Hybride	Logique Floue	MEEN	OOEN	LDs	OWL	RDF	Connaissances Probabilistes	Connaissances Floues	Inférence Probabiliste	Inférence ontologique	Inférence ontologique floue					
Ceylan et al., 2015							✓					✓							
Giugno et al., 2002		✓																	
Polastro et al., 2012																			
Koller, 1997																			
Fukushige et Yoshio, 2005		✓							✓										
Yang et Calmet, 2005		✓							✓										
Ding et al., 2006		✓							✓										
Costa et al., 2005		✓							✓										
Carvalho et al., 2017		✓			✓				✓										
Erma et al., 2017		✓							✓										
Mohammed et al., 2016									✓										
Erma et al., 2016a	✓								✓										
Erma et al., 2016b									✓										
Costa et al., 2003		✓							✓										
Zheng et al., 2016									✓										
Fenz , 2012		✓							✓										
Ishak et al., 2011		✓							✓										
Monemis et al., 2014		✓							✓										
(Bobillo et al., 2009)		✓							✓										
Bobillo et al., 2011									✓										
Mazzeri et al., 2005)									✓										
Bobillo et al., 2008 ;																			
Bobillo et al., 2012																			
Bobillo et Straccia., 2016																			
Djellal et Boufaïda., 2017				✓															

✓ Le critère est traité par l'approche.

FIGURE 3.2 – Tableau récapitulatif de la comparaison.

3.4.1 Discussion sur les ontologies probabilistes

Nous avons classé les approches probabilistes en trois catégories. La première catégorie représente un échantillon de solutions proposées afin de permettre aux DLs de représenter des connaissances probabilistes (koller, 1997; polastro et al., 2012; giugno et al., 2002; ceylan et al., 2015) [94, 95, 96, 98], telles que des contraintes probabilistes et faire quelques tâches de raisonnements probabilistes telles que la subsumption probabiliste, etc.

La deuxième catégorie représente un ensemble d'extensions du langage RDF afin de lui permettre de représenter la sémantique d'un RB. Ceci dont l'objectif de bénéficier des avantages des deux formalismes à la fois (fukushige.2005; udrea et al., 2006) [100, 101].

La troisième catégorie regroupe un ensemble de solutions basées sur la théorie des probabilités, notamment les réseaux bayésiens et leurs extensions pour traiter l'incertitude liée aux composants de l'ontologie codée dans le langage OWL. Trois démarches sont distingués dans la littérature pour la combinaison des ontologies avec les RBs, en particulier :

1. La première démarche se porte sur la construction des RBs à partir des ontologies existantes. Dans cette démarche les solutions proposées dans [103, 105] visent à construire des RBs à partir des ontologies existantes. En effet, l'approche proposée dans Ishak et al., 2011 [103], modélise les connaissances probabilistes de l'ontologie sous forme d'un OOBN. Cette approche est conçue pour faire face à des problèmes complexes avec des structures répétitives. Quant à l'approche proposée dans Fenz ., 2012 [105], s'appuie sur des ontologies existantes pour construire des RBs standards. Les deux solutions permettent de traiter les connaissances probabilistes, cependant. Elles ne fournissent pas des alternatives afin d'incorporer cette connaissance dans l'ontologie pour qu'elle soit formalisée et traitable automatiquement par des machines. A notre avis, ces approches doivent ajouter une étape d'intégration de connaissances probabilistes dans leurs processus de modélisation. Cela va permettre d'incorporer les connaissances probabilistes modélisées dans le RB dans l'ontologie de manière formelle.
2. La deuxième démarche a pour objectif d'intégrer des RBs dans des ontologies classiques Mouenis et al., 2014 [106]. Elle vise à représenter des connaissances modélisées dans un graphe RB telle que les liens de dépendances et les mesures des probabilités dans une ontologie d'une manière formelle. Cette solution tente de représenter formellement la sémantique des RBs dans une ontologie. Cependant, les liens entre les variables aléatoires représentés dans le RB et les composants de l'ontologie ne sont pas pris en considération par la solution.
3. La troisième démarche se focalise sur le développement des extensions des langages du Web sémantique. En effet, ces extensions font la construction d'un RB à partir d'une ontologie et l'intégration sémantique du RB dans l'ontologie. L'avantage de cette démarche réside dans le fait que la sémantique de RB sera formellement représentée dans l'ontologie qu'elle préserve la sémantique de l'ontologie initiale. Les auteurs dans [1, 2, 3, 4, 5, 6,

7, 8, 108] ont augmenté les capacités du langage OWL pour modéliser des connaissances à caractère incertain en utilisant de nouvelles classes et propriétés. Ces nouvelles classes et propriétés permettent la modélisation des graphes RBs qui capturent les connaissances incertaines de l'ontologie et les représenter dans l'ontologie de manière formelle.

BayesOWL et OntoBayes [1, 2] visent à étendre le langage OWL en utilisant les RBs standards. Il s'agit dans la première étape de construire un Rb à partir de la sémantique de l'ontologie classique. Par la suite la sémantique du réseau bayésien généré sera représentée formellement à l'aide des annotations supplémentaire en OWL. En effet, les deux solutions sont basées sur des annotations pour la représentation du réseau bayésien construit dans l'ontologie.

En ce qui concerne PR-OWL [3, 4], est une ontologie de haut niveau pour le développement des ontologies probabiliste basée sur un autre type de réseaux bayésiens appelé MEBN. Comme déjà mentionné, un MEBN est une extension de RB standard dont le but est d'augmenter son expressivité par l'intégration de la logique de premier ordre. Pour créer des ontologies probabilistes basées sur PR-OWL, un Framework (UnBBayes) puisant est développé pour ça, il est constitué de plusieurs interfaces et des plug-ins (MEBN, PR-OWL1.0 et PR-OWL2.0). Cette approche présente plusieurs avantages, notamment, elle est basée sur les MEBNs et ces derniers présentent plusieurs avantages par rapport aux RBs standards. En outre, un grand nombre d'applications réelles ont été développées en se basant sur ce langage.

Par ailleurs, l'extension présentée dans Zheng et al., 2016 [5] nous semble intéressante car elle tente de traiter des connaissances floues dans les ontologies probabilistes. En effet, dans cette solution les auteurs ont augmenté l'ontologie de haut niveau de PR-WOL pour la modélisation des règles floues sans proposer des mécanismes pour faire l'inférence probabiliste floue en utilisant les connaissances floues modélisées dans l'ontologie.

La solution proposée dans Mohammed et al., 2016 [8], présente une forte contribution dans le domaine. Elle est basée sur les réseaux bayésiens hybrides afin de traiter les connaissances probabilistes attachées à des composants ontologiques continus et discrets. L'objectif de la contribution présentée dans Emna et al., 2016a; Emna et al., 2016b [6, 7], est de définir un méta-modèle pour les ontologies probabilistes. En effet, les auteurs ont proposé une extension du méta-modèle (ODM-OWL2) en ajoutant un ensemble de classes et propriétés pour distinguer entre les composants probabilistes et les composants déterministes de l'ontologie. Comme avantage de cette approche c'est qu'elle permet de modéliser des problèmes en utilisant la théorie des probabilités seulement et qui n'est pas lié aux RBs. Par conséquent, elle est efficace surtout dans le cas de l'insuffisance et l'indisponibilité de données qui peut être considéré comme un obstacle dans l'étape d'estimation de probabilités dans les réseaux bayésiens [6].

La solution présentée dans Emna et al., 2018 [108], vise à construire des ontologies probabilistes en se basent sur la construction d'un RB à partir des instances de l'ontologie.

Néanmoins, ce travail présente des inconvénients, notamment : Le fait que dans le réseau bayésien généré, les états des nœuds sont uniquement booléens ; outre, il s'occupe uniquement sur des connaissances probabilistes liées aux instances de l'ontologie et ne tient pas compte des connaissances probabilistes liées à la partie schéma de l'ontologie.

Par ailleurs, diverses applications basées sur les ontologies probabilistes ont été développées dans la littérature, parmi lesquelles nous citons :

- **Le cloud computing** [114]. Dans cette étude, un modèle de confiance et ontologie probabiliste pilotés par les réseaux bayésiens pour la gestion des accords de services dans l'environnement de services infonuagiques a été proposé. Le système comporte plusieurs modules entre autres, un module de confiance et un module intelligence.

Le *module de confiance* a pour objectif de calculer le niveau de confiance d'un fournisseur donné en analysant les données historiques de sa réputation via une inférence bayésienne [115].

Le *module intelligence* a pour but de collecter une base de connaissances et représenter sémantiquement des violations et des prédictions précédentes dans une ontologie probabiliste formalisée dans le langage PR-OWL2. Il exploite ainsi l'inférence bayésienne de l'ontologie probabiliste qui est basée sur les MEBNs pour prédire la probabilité d'une violation future en exploitant les données historiques.

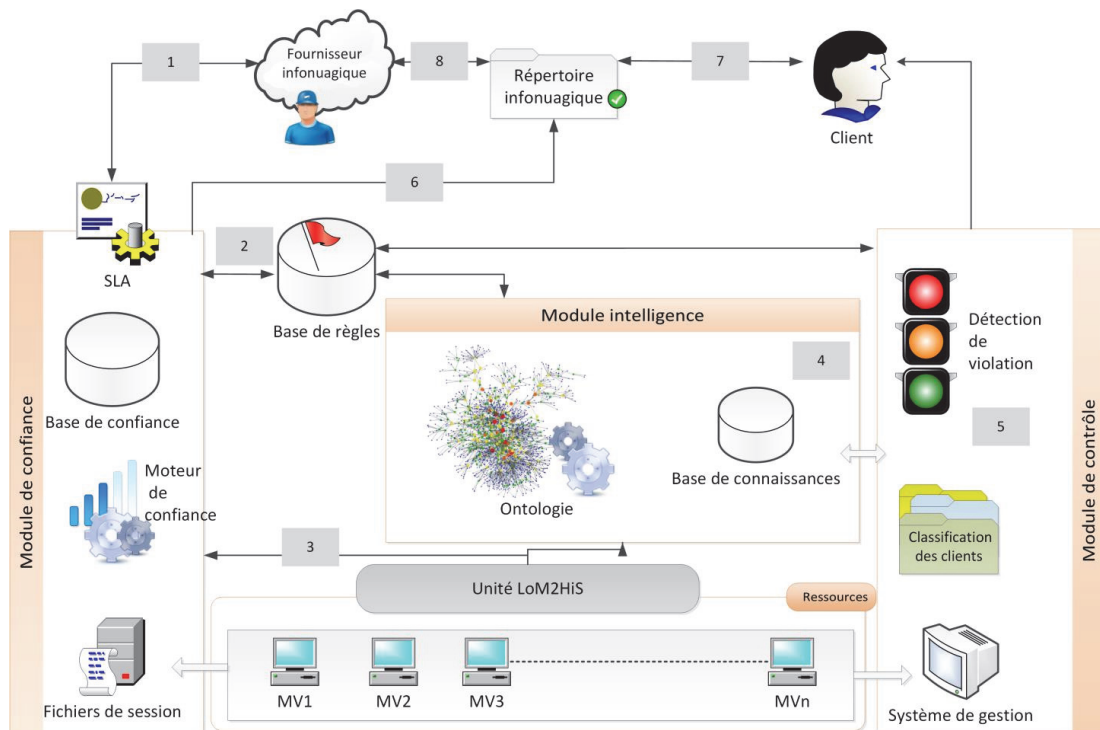


FIGURE 3.3 – Architecture globale du système développé [114].

- **Systèmes d'aide à la décision pour l'identification des terroristes** [116]. Dans ce travail, un système d'aide à la décision basé sur une ontologie probabiliste codée dans le langage PR-OWL2 pour l'identification des terroristes a été développé. Le système proposé est capable à identifier les personnes qui doivent faire l'objet d'un examen plus approfondi et qui présentent un risque pour les pays cibles.

En effet, une ontologie probabiliste d'identification du terrorisme a été conçue et implémentée. Elle peut aider l'analyste à déterminer la probabilité qu'une personne soit impliquée dans le terrorisme en utilisant des informations sur ses relations et ses communications.

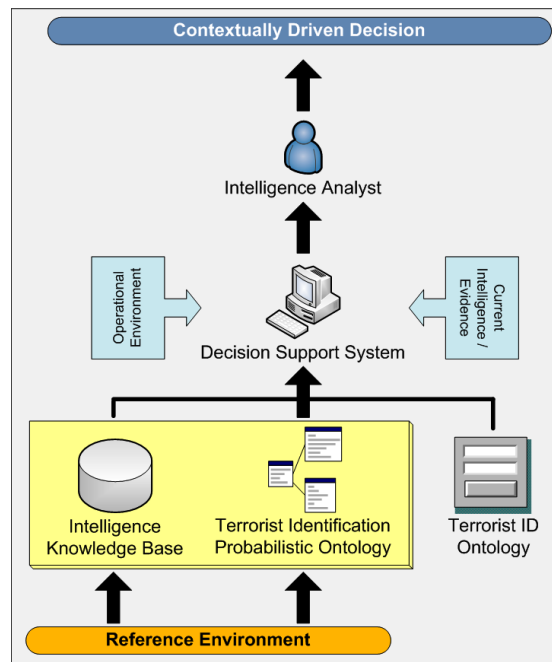


FIGURE 3.4 – Architecture du système d'aide à la décision d'identification des terroristes [116].

3.4.2 Discussion sur les ontologies floues

Les approches floues traitent la connaissance floue dans les ontologies classiques afin de traiter les composants mal-définis de l'ontologie. Comme nous l'avons déjà évoqué, il n'existe pas une définition unique pour les ontologies floues, chaque auteur a sa propre vision sur ce que sont les ontologies floues. Cependant, tous les auteurs sont d'accord qu'une ontologie floue est une ontologie classique qui utilise la logique floue pour traiter ses composants floue. En effet, on distingue trois classes d'approches floues, les approches basées sur les DLs, permettant de représenter le flou dans leurs constructeurs. Les approches basées sur le RDF représente des triples flous, et finalement, les approches basées sur le langage OWL pour faire face à la connaissance floue dans ses composants. En outre, plusieurs raisonneurs ont été implémentés pour faire des

taches de raisonnement sur les ontologies floues en tenant compte de la connaissance floue représentée formellement dedans. Par ailleurs, les ontologies floues ont prouvés leurs succès dans un bon nombre d'applications réelles, entre autres :

- **L'aide à la décision** [117]. Un système d'aide à la décision mobile basé sur une ontologie floue a été implémenté. En effet, une plateforme Web et une application Android ont été développées pour aider les utilisateurs à choisir le vin le plus adapté au contexte du dîner. Dans cette étude, l'ontologie floue du vin (Fuzzy Wine Ontology) a été exploitée, cette ontologie été conçue pour être utilisée dans des applications développées à des fins industrielles [118].

Comme le montre la figure 3.5, le système est capable d'effectuer des recommandations du vin selon les informations fournies par l'utilisateur à travers un questionnaire. Les questions sont présentées une par une aux invités pour une meilleure lisibilité. Le système est capable de recommander aux invités les vins selon leur pertinence avec la recherche.

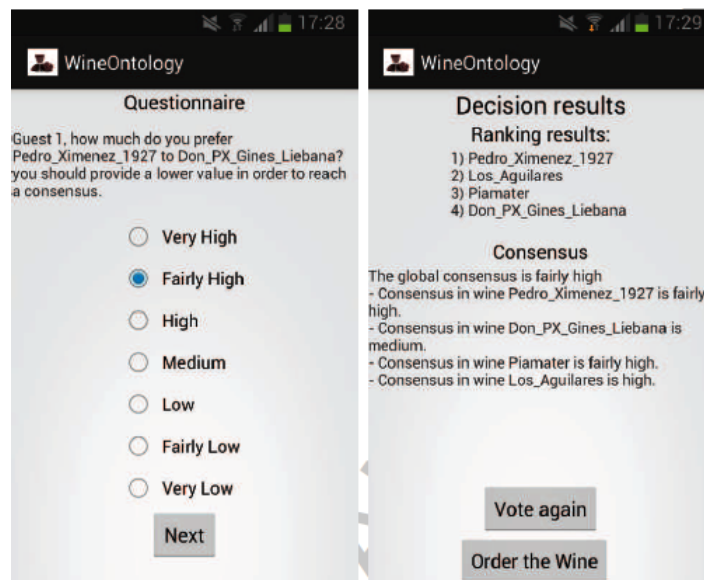


FIGURE 3.5 – Capture d'écran du questionnaire et les résultats temporaires de la décision [117].

- **Applications médicales** [119]. Dans ce travail, un système de raisonnement à base de cas pour le domaine du diabète a été implémenté. Le système repose principalement sur une ontologie floue pour modéliser les connaissances du domaine et pour répondre à des requêtes sémantiques en tenant compte du flou au niveau des composants de l'ontologie. L'ontologie développée contient 63 classes (floues), 54 propriétés d'objets (floues), 138 propriétés de données (floues), 105 types de données flous. Elle a été peuplée de 60 cas issus des données réelles des diabétiques.

Le système prend en charge de nombreux types de raisonnement, tels que le raisonnement sémantique, précis et flou. Ainsi, le moteur d'exécution de requêtes sémantiques floues est illustré dans la figure 3.6

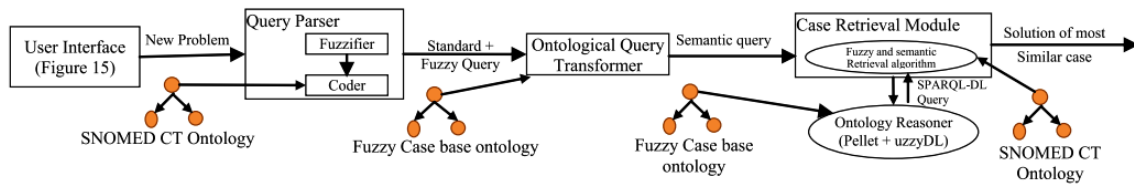


FIGURE 3.6 – Le moteur d'exécution de requêtes sémantiques floues [119].

La saisie des requêtes d'interrogations a été facilitée grâce à l'interface illustrée dans la figure 3.7. Cette interface permet d'introduire toutes les informations requises pour prédire un diagnostic du diabète.

FIGURE 3.7 – Un formulaire de saisie de requête sémantique floue [119].

En résumé, malgré les efforts fournis par la communauté de chercheurs dans le Web sémantique pour la modélisation et le raisonnement avec l'incertitude dans les ontologies, et malgré

le nombre considérables des applications réussies en utilisant les ontologies probabilistes et les ontologies floues. A ce jour, il n'y a aucune solution considérée comme un standard par le W3C; ce qui montre que ce domaine est encore dans son jeune-âge et les solutions proposées dans la littérature n'ont pas encore atteint un niveau de maturation et d'expressivité suffisantes pour bien modéliser des problèmes de monde réel avec l'incertitude. Cependant, les deux principales classes d'approches sont les approches probabilistes et les approches floues. Les approches probabilistes permettent de traiter les connaissances probabilistes attachées aux composants **biens-définis** de l'ontologie. Les approches floues sont fondées sur la logique floue traitent la connaissance floue qui est liée aux composants d'ontologie qui sont **mal-définis**.

Chaque solution tente de traiter un type particulier de connaissance incertaine. Il est difficile de conclure qu'une solution est meilleure qu'une autre. Ce jugement dépendrait fortement des besoins derrière la modélisation et du raisonnement avec la connaissance ontologique probabiliste ou floue et parfois de la complexité de problème à traiter. En outre, la connaissance probabiliste et la connaissance floue ont été traités séparément dans les ontologies et peu d'efforts ont été menés dans cette perspective. Toutefois, dans la plupart des cas, les problèmes du monde réel impliquent plusieurs types d'incertitude en même temps, où les composants mal-définis et les composants bien-définis apparaissent simultanément. Par conséquent, il y a une nécessité de les traiter simultanément dans les ontologies afin d'offrir une meilleure flexibilité aux agents du Web sémantique pour acter et faire des raisonnements automatiques en présence de l'incertitude. Dans cette perspective, nous proposons dans le reste de cette thèse une solution prometteuse.

3.5 Conclusion

A travers ce chapitre, nous avons présenté un état de l'art sur les ontologies probabilistes et les ontologies floues. Nous avons exposé plusieurs approches probabilistes et d'autres floues qui visent à améliorer la représentation et le raisonnement ontologique classique en tenant compte de l'incertitude. Pour chacune des deux approches, nous avons présenté quelques définitions pour mettre en évidence les notions de bases. Finalement, nous avons discuté et comparé ces approches en utilisant plusieurs critères d'évaluation en élaborant un tableau récapitulatif de la comparaison.

Comme nous l'avons précisé précédemment, les connaissances floues et les connaissances probabilistes ont été traitées séparément. Dans le reste de la thèse, nous proposons une solution qui tient compte de ces limites. Dans notre première contribution, nous proposons une solution qui traite le flou et les probabilités à la fois. Cette solution sera exploitée par la suite dans le contexte du Web sémantique afin de traiter simultanément les connaissances floues et les connaissances probabilistes dans les ontologies.

Chapitre 4

Extension des Réseaux Bayésiens

Multi-Entités basée sur la logique floue

“Les vérités qu’on aime le moins à apprendre sont celles que l’on a le plus d’intérêt à savoir”
— Kôdô Sawaki

Sommaire

4.1	Introduction	73
4.2	Motivation de la problématique	74
4.3	Réseaux Bayésiens Multi Entités Flous	76
4.3.1	Modélisation de l’incertitude via FzMEBN	76
4.3.2	Observations dans FzMEBN	78
4.3.3	Inférence dans FzMEBN	84
4.3.4	Exemple Illustratif	89
4.4	Synthèse	92
4.5	Conclusion	94

4.1 Introduction

Les MEBNs sont des modèles graphiques probabilistes permettant de représenter les différents problèmes du monde réel comme des fragments des RBs. Ces fragments regroupent un ensemble de variables représentées sous forme d’expressions de la logique du premier ordre liées entre elles et ayant une sémantique commune. Ainsi, les MEBNs permettent de faire l’inférence probabiliste pour déduire de nouvelles connaissances par la génération d’un SSBN qui reflète bien la situation observée. Cela en mettant à jour de l’état de croyances d’une entité cible en se basent sur les observations disponibles sur d’autres entités du modèle.

Un réseau MEBN peut représenter plusieurs formes d’incertitude. Il peut représenter des connaissances probabilistes épistémiques et ou stochastiques dans une MThéorie. Ainsi, la croyance

relative à une entité à un instant donné est exprimée sous la forme de distributions de probabilités à posteriori, obtenues par la propagation des observations sur certaines entités en appliquant des algorithmes d'inférence classiques. Cependant, dans MEBN, les différentes variables sont supposées précises et les observations sont souvent considérées comme certaines. C'est-à-dire, on connaît avec certitude et précision l'appartenance de la valeur observée pour chaque entité aux différents états.

Dans ce contexte, plusieurs questions se posent et s'imposent :

- *Comment modéliser la connaissance floue liée aux variables floues (c'est-à-dire, non précises) dans un réseau bayésien multi-entités ?*
- *Comment tenir compte des observations floues (non précises) dans un réseau bayésien multi-entités ?*
- *Comment ces observations sont-elles représentées ?*
- *Comment ces observations sont-elles propagées lors de l'inférence ?*

Nous allons donner des éléments de réponse à ces questions et nous proposons une nouvelle approche qui traite les observations floues dans les MEBNs.

Le reste de ce chapitre est organisé comme suit : nous commençons par décrire le contexte de notre travail. Nous poursuivons par donner des éléments de réponses aux questions citées en haut et nous exposons notre approche pour le traitement du flou dans les MEBNs en détaillant ses concepts, ses éléments de base. Puis nous présentons la notion des observations floues dans notre extension et nous allons présenter le processus d'inférence de notre solution. Nous terminons par une synthèse et une conclusion.

4.2 Motivation de la problématique

Les RBs sont des modèles de représentation et de raisonnement avec des connaissances incertaines. Un RB peut capturer deux types de connaissances : une connaissance qualitative et autre quantitative. La connaissance qualitative est capturée à l'aide d'un graphe dirigé acyclique constitué d'un ensemble de variables liées entre elles par des arcs. En effet, les arcs représentent des liens de dépendances et de causalité entre des variables. Quant à la connaissance quantitative, est quantifiée sous forme de distributions de probabilités, permettant de mesurer numériquement l'incertitude liée aux variables connectées du graphe.

Les RBs visent à représenter les connaissances du système sous forme d'un graphe facile à comprendre et interpréter par les spécialistes et les non-spécialistes du domaine. En outre, les RBs exploitent ces connaissances en matière d'aide à la décision en se basant sur l'inférence bayésienne, afin de déduire de nouvelles connaissances à partir des observations supplémentaires sur certaines variables. Ainsi, l'inférence bayésienne consiste à calculer les probabilités à posteriori possibles dans le RB étant donné un ensemble d'observations. Il consiste à mettre à jour l'état de la croyance de certaines variables en se basant sur les informations observées

sur d'autres variables. Il tient compte des connaissances représentées dans le modèle et les observations. Ceci est fait par la propagation des observations relatives aux variables observées et mettre à jour les distributions de probabilité pour les autres variables au fur et à mesure.

Cependant, dans les cas pratiques, les observations accordées par un observateur peuvent être suivies par incertitude et du flou. En effet, la tâche de collecte des observations peut être soumise aux imprécisions et aux incertitudes qui peuvent être engendrées par l'observateur lui-même ou les instruments de mesure [120]. Par exemple, l'information fournie par l'observateur d'un appareil de mesure de tension artérielle peut être non fiable ou floue par nature.

Comme nous l'avons déjà mentionné dans le chapitre 3, plusieurs méthodes ont été développées pour traiter le flou dans les RBs. Nous nous sommes intéressés dans notre travail au traitement des observations floues dans les MEBNs.

Les MEBNs offrent une approche prometteuse qui traite le problème de l'expressivité et la flexibilité dans les RBs standards par la combinaison des RBs avec la logique du premier ordre et l'utilisation de la notion des fragments des RBs. Cependant, le problème est que les MEBNs peuvent traiter que la connaissance probabiliste et ils ignorent la connaissance floue qui est souvent liée à leurs variables aléatoires. En effet, pour modéliser et quantifier l'incertitude de manière souhaitable et efficace, on croit fortement qu'une extension floue des MEBNs doit être développée qui tiennent compte de la connaissance floue liée à des événements probabilistes.

Nous avons exploité une technique déjà existante et connue pour la représentation et le raisonnement en présence du flou dans les réseaux bayésiens standards afin d'augmenter les capacités de modélisation et de raisonnement des MEBNs. En effet, nous avons proposé et développé une nouvelle extension qui tient compte de la connaissance floue dans les MEBNs. Ainsi, l'idée de notre approche se reflète dans :

- La modélisation de la connaissance sémantique floue dans les MEBNs. Ceci est fait en définissant un ensemble de constructeurs (nouveaux types de nœuds) pour distinguer entre la connaissance probabiliste précise et la connaissance probabiliste floue.
- Proposition d'un processus général pour faire l'inférence probabiliste en se basant sur des observations floues dans les MEBNs.
- Notre processus d'inférence proposé est basé sur la méthode de l'évidence virtuelle présentée dans le chapitre 2. En effet, notre choix se justifie par les avantages de cette méthode par rapport aux autres méthodes. La méthode de l'évidence virtuelle est générale, elle ne nécessite aucun changement dans les algorithmes d'inférences bayésiens classiques. Par conséquent, on peut utiliser n'importe quel algorithme pour propager les observations floues lors de l'inférence probabiliste. Ainsi, le choix du type d'algorithme dépend de la contribution des développeurs et de la complexité du problème en question.

4.3 Réseaux Bayésiens Multi Entités Flous

L'objectif de n'importe quel formalisme de représentation des connaissances est de représenter le monde et la réalité perçue avec des structures de connaissances qui reflètent bien les structures du monde. Ces structures facilitent l'interprétation et l'exploitation des connaissances pour inférer d'autres connaissances. En effet, le modèle MEBN représente le monde en se basant sur l'expressivité de la logique du premier ordre. Il capture les connaissances probabilistes liées aux entités du monde comme des fragments de RBs qu'on peut instancier au fur et à mesure pour déduire de nouvelles connaissances.

Dans ce qui suit, nous proposons une extension des MEBNs afin de soutenir le raisonnement probabiliste par la logique floue. Notre extension tient compte de la connaissance floue qui est une propriété inhérente de notre monde et qui est présente dans la plupart des problèmes réels qu'on cherche souvent à traiter.

4.3.1 Modélisation de l'incertitude via FzMEBN

Notre approche a pour originalité de séparer la partie précise et la partie floue du réseau. Cette séparation est considérée très importante pour la mise en œuvre de l'étape du raisonnement. En effet, une des lacunes des MEBNs standards est qu'ils ne sont pas capables à prendre en charge les variables aléatoires floues. Ces variables doivent être traitées de manière différente que les variables précises. Ceci s'explique par le fait que ces variables sont floues par nature et n'ont pas de bornes claires et précises. Pour remédier à ce problème, nous proposons donc l'ajout de nouveaux types de variable pour distinguer entre les éléments flous et les éléments précis du réseau MEBN.

Nous présentons ce qui suit des définitions relatives à notre proposition afin de mettre en évidence notre démarche théorique.

Un réseau bayésien multi-entités flou (FzMEBN : FuZzy Multi Entity Bayesian Networks) est une extension d'un réseau bayésien multi-entités classique. Son objectif est d'exploiter et de bénéficier des avantages de la logique floue. Il améliore également l'expressivité de MEBN classique pour mieux modéliser les situations et les problèmes du monde réel.

La connaissance dans FzMEBN est représentée comme des MEBN théories (MTheories). Chaque MTheorie est composée d'un ensemble de fragments de réseaux bayésiens flous (FzM-Frags).

Formellement, un FzMFrag est défini comme un **7-tuple** $\langle \mathbf{C}, \mathbf{R}, \mathbf{I}, \mathbf{FR}, \mathbf{FI}, \mathbf{G}, \mathbf{D} \rangle$, tels que :

- **C** représente un ensemble de nœuds de contexte,
- **R** représente un ensemble de nœuds résidents,
- **I** représente un ensemble de nœuds d'entrées (Input Nodes),
- **FR** représente un ensemble de nœuds résidents flous (Fuzzy Residents Nodes),
- **FI** représente un ensemble de nœuds d'entrées flous (Fuzzy Input nodes),

- **G** représente un graphe hybride (Fuzzy MFrag Graph),
- **D** représente les distributions locales.

La figure 4.1 montre les éléments de base de FzMEBN, en particulier, le FzMFRag. Il s'agit d'un ensemble de nœuds de contexte c_1 et c_2 afin de définir des variables ordinaires ou des conditions. En outre, il comporte des nœuds résidents classiques (comme le nœud R_1) ou bien des nœuds résidents flous (le nœud FR_1) qui sont attachés par des fonctions d'appartenance. Ces fonctions d'appartenance seront utilisées pour la fuzzification des SSBN au cours du processus d'inférence. Le FzMFRag contient également des nœuds d'entrées (le nœud I_1) qui réfèrent d'autres nœuds résidents, dans le cas où le nœud d'entrée fait référence à un nœud résident flou, il sera considéré comme un nœud d'entrée flou (le nœud FI_1).

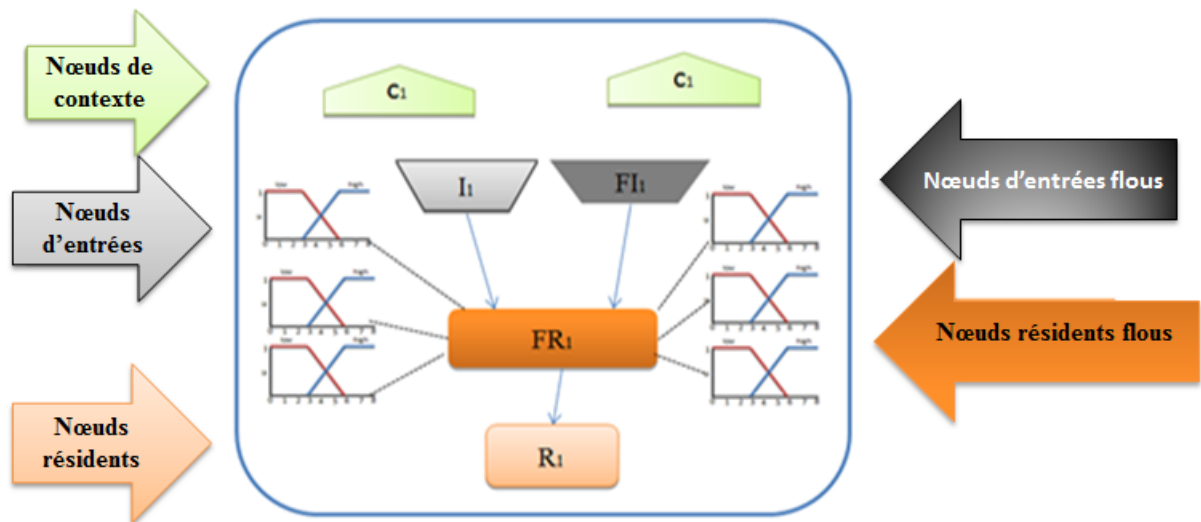


FIGURE 4.1 – Illustration d'un FzMFRag.

1. **Nœuds de contexte (C)** : ce type de nœuds représente les variables aléatoires booléennes permettant de définir des conditions et des contraintes qui doivent être vérifiées pour que la distribution dans le FzMFRag soit valide.
2. **Les Nœuds résidents (R)** : représentent des formules de la logique du premier ordre qui sont associées à l'incertitude. Un nœud résident peut être un prédicat, une fonction ou toute une formule appartenant à la logique du premier ordre.
3. **Les Nœuds d'entrées (I)** : ces nœuds peuvent être vus comme des nœuds étrangers ou des pointeurs qui réfèrent d'autres nœuds résidents définis dans un autre FzMFRag. Leurs propres distributions sont définies ailleurs dans leurs FzMFRag d'origines.
4. **Les Nœuds résidents flous (FR)** : sont des extensions des nœuds résidents classiques permettant au FzMFRag de faire face à des connaissances floues. Comme les nœuds résidents classiques, les nœuds résidents flous sont associés à des distributions de probabilités. En outre, ils peuvent représenter les connaissances floues liées aux définitions de leurs états en utilisant des fonctions d'appartenance.

5. **Les Nœuds d'entrées flous (FI)** : les nœuds d'entrées flous sont des nœuds résidents flous définis dans un autre FzMFRag. Les nœuds d'entrées flous peuvent également influencer sur la distribution de probabilité d'autres variables. Cependant, leurs distributions de probabilités et leurs propres fonctions d'appartenance sont définies dans leurs propres FzMFRag d'origines.

4.3.2 Observations dans FzMEBN

L'objectif des FzMEBNs est de faire l'inférence bayésienne en se basant sur les informations et les observations d'un observateur au sein du système étudié. La précision de ces observations joue un rôle très important lors du calcul des probabilités à posteriori pour une entité cible. Cependant, dans les cas pratiques, ces observations ne sont pas toujours exactes et fiables.

Dans ce qui suit nous présentons quelques définitions qui portent sur les observations dans les FzMEBNs.

Soient M et R deux variables qui représentent respectivement un FzMEBN, un nœud résident avec n variables ordinaires $\Phi = \{\theta_1, \dots, \theta_n\}$, telle que : $1 \leq j \leq n$ et $D = \{x_1, \dots, x_m\}$ l'ensemble contenant les états du nœud R .

Soit l'ensemble $E = \{o_1, \dots, o_a\}$ qui représente des observations (informations supplémentaires) dans M sur un nœud résident R .

Chaque observation $o_i \in E$ représente une instantiation de toutes les variables ordinaires $\theta_j \in \Phi$ de R . En effet, cette observation est obtenue en fixant pour chaque variable ordinaire θ_j un identifiant ω_{jk} qui représente l'identifiant d'une entité k qui participe dans l'observation o_i .

Dans FzMEBN, le résultat d'une observation o_i qui satisfait les nœuds de contexte définis dans son FzMFRag sera la création d'un nœud classique r_i dans le SSBN. Ainsi, l'observation o_i est caractérisée par une valeur d'observation V_i et par un vecteur d'observation floue de taille $m = |D|$, tel que m représente le nombre d'états du nœud R .

La figure 4.2 montre le processus d'instanciation et les observations produites sur un nœud résident R .

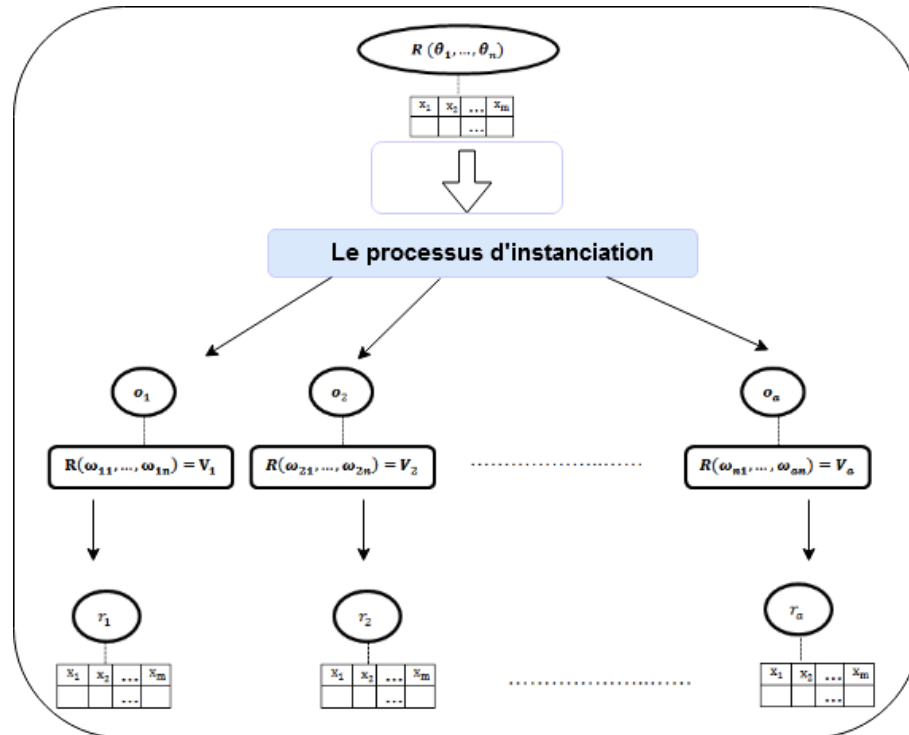


FIGURE 4.2 – Illustration des observations sur un nœud résident R.

4.3.2.1 Observation certaine

Si une observation o_i concerne un nœud résident précis, on appelle cette observation "observation certaine". Elle est caractérisée par :

- Une valeur d'observation V_i correspond à un et un seul état $x_k \in D$
- Un vecteur d'observation contient un unique 1 dans le vecteur à la position qui correspond à l'état x_k et des 0 dans les positions relatives aux autres états de r_i .

Cette observation représente l'instanciation de r_i avec la valeur x_k et elle est caractérisée par $P(r_i = x_k | o_i) = 1$. Autrement dit, lorsqu'une observation est certaine, la valeur d'observation appartient à un et un seul état x_k .

La figure 4.3 (a) donne une illustration graphique d'une observation certaine. En effet, les probabilités a posteriori du nœud r_i seront mises à zéro dans tous les états du nœud r_i , sauf l'état qui correspond à la valeur 1 dans le vecteur d'observation sera mise à 100% .

Exemple 4.3.6 Soit un FzMEBN M contenant le nœud résident « Sexe(p) » qui représente le sexe d'une personne p , avec l'ensemble d'états suivant : $D = \{H : Homme, F : Femme\}$.

Soit l'observation $o_1 = \ll p_1 \text{ est personne de sexe Femme} \gg$. La représentation formelle de cette observation est donc : $\text{Sexe}(p_1) = \text{Femme}$. Dans cette observation il s'agit d'instancier le nœud résidant Sexe(p) en utilisant l'entité p_1 avec **la valeur d'instanciation Femme**.

On obtient alors un nœud classique nommé Sexe $_{p_1}$ dans le SSBN avec les mêmes états que celles définis dans le nœud résident « Sexe(p) ». Par ailleurs, **son vecteur d'observation est le suivant** :

- $e = \langle P(\text{Sexe}_{p1} = H \mid o_1), P(\text{Sexe}_{p1} = F \mid o_1) \rangle$
- $e = \langle 0, 1 \rangle$.

Cette observation est illustrée dans la figure 4.3 (b).

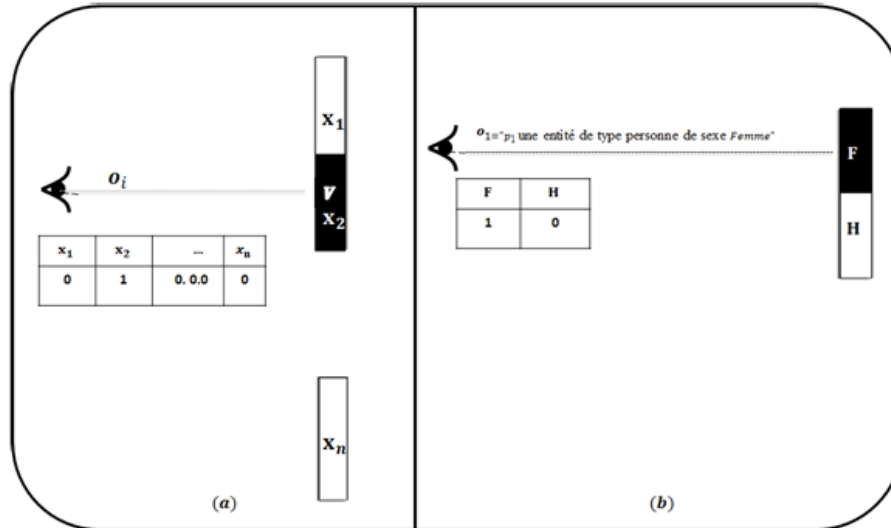


FIGURE 4.3 – Représentation graphique de l'observation o_1 .

Les probabilités a posteriori du noeud Sexe_{p1} après l'observation o_1 sont les suivantes :

- $P(\text{Sexe}_{p1} = H \mid o_1) = 0$,
- $P(\text{Sexe}_{p1} = F \mid o_1) = 100$.

4.3.2.2 Observation floue

Si une observation o_i concerne un noeud résident flou, on appelle cette observation "observation floue". Cette observation représente l'instanciation de r_i avec tous les états $x_k \in D$. Autrement dit, lorsqu'une observation est floue, la valeur d'observation peut appartenir à plusieurs états à la fois mais avec degrés d'appartenance progressifs. Elle est caractérisée par :

- Une valeur d'observation $V_i \in U$, tel que U est l'univers du discours de noeud r_i ,
- Un vecteur d'observation floue $fe_i(V_i) = \langle \mu_{x1}(V_i), \mu_{x2}(V_i), \dots, \mu_{xm}(V_i) \rangle$. Il contient dans chaque case k degré d'appartenance de V à l'état x_k de r_i dans l'intervalle $[0, 1]$.

Tel que, $\mu_{xk}(V_i)$ représente le degré d'appartenance de V à l'état x_k . La figure 4.4(a) illustre une observation floue.

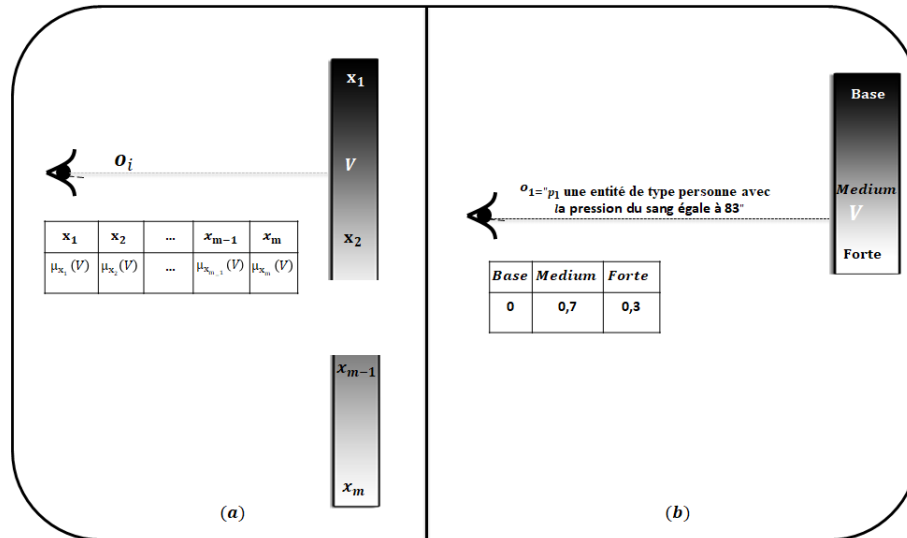


FIGURE 4.4 – Illustration d'une observation floue.

L'implémentation de la méthode de l'évidence virtuelle est simple et équivaut à créer un nœud enfant temporaire pour chaque nœud flou et à renseigner sa table de probabilités conditionnelles avec des valeurs en se basant sur le vecteur d'observation floue. L'interprétation théorique de ce nœud temporaire est qu'il fournit des informations floues sur son parent.

Pour montrer comment on peut incorporer le vecteur d'observation floue avec les probabilités marginales afin de calculer les probabilités a posteriori. Prenant la cas illustré dans la figure 4.5, qui fait rappel à un nœud r_i avec son nœud virtuel Vr_i . En effet, la probabilité a posteriori du nœud r_i pour un état x_k est calculée en utilisant le nœud virtuel Vr_i comme suit :

$$\begin{aligned}
 P(r_i = x_k | Vr_i = x_1) &= \frac{P(Vr_i = x_1 | r_i = x_k) \cdot P(r_i = x_k)}{P(Vr_i = x_1)} \\
 &= \frac{\mu_{xk}(V_i) \cdot P(r_i = x_k)}{\sum_k^m P(r_i = x_k) \cdot \mu_{xk}(V_i)} \\
 &= \lambda \cdot \mu_{xk}(V_i) \cdot P(r_i = x_k)
 \end{aligned} \tag{4.1}$$

Telle que λ est une constante de normalisation. $\lambda = 1 / \sum_k^m P(r_i = x_k) \cdot \mu_{xk}(V_i)$.

Avec,

- $P(r_i = x_1 | o_i) = \lambda \cdot P(r_i = x_1) \cdot \mu_{x1}(V_i)$.
- $P(r_i = x_2 | o_i) = \lambda \cdot P(r_i = x_2) \cdot \mu_{x2}(V_i)$.
- $P(r_i = x_k | o_i) = \lambda \cdot P(r_i = x_k) \cdot \mu_{xk}(V_i)$
- $P(r_i = x_m | o_i) = \lambda \cdot P(r_i = x_m) \cdot \mu_{xm}(V_i)$.

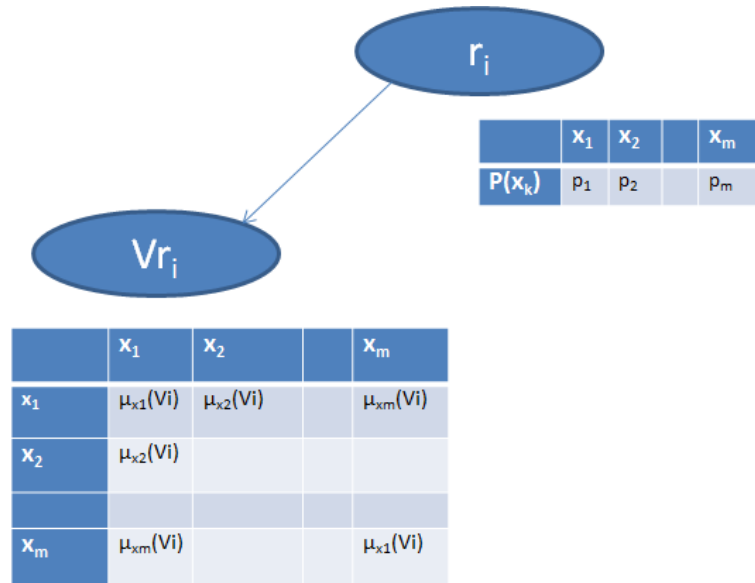


FIGURE 4.5 – Illustration d’une observation floue.

Exemple 4.3.7 Pour illustrer le concept de l’observation floue, on suppose qu’on a un FzMEBN M contenant la variable « TensionArtérielle(p) » qui représente la Tension Artérielle d’une personne p , avec l’ensemble des états défini par les intervalles suivants :

$$D = \{\text{Base}(B) : [0 - 60[, \text{Medium}(M) : [60 - 85], \text{Forte}(F) :] 85 - 200]\}.$$

En effet, la Tension Artérielle d’une personne p peut avoir une extension floue. Ce flou est omniprésent dans le langage naturel. Il est dû ainsi à la définition vague des ensembles des états qui sont exprimés par l’absence d’une connaissance rigoureuse.

Soit l’observation $o_1 = \ll \mathbf{p1}$ est une personne, sa tension artérielle correspond à la pression du sang égale à 83 mm Hg (millimètre de mercure) ». On obtient à partir de cette observation un nœud classique nommé TestionArtérielle-P-1 (pour faciliter l’écriture des formules on note TAp_1) dans le SSBN. Supposons que les probabilités marginales du nœud TAp_1 sont représentées dans le tableau suivant.

x_i	$\mathbf{P}(TAp_1 = x_i)$
Base	0.2
Medium	0.35
Forte	0.45

TABLE 4.1 – Les probabilités marginales du nœud TAp_1

Si on considère que cette observation est certaine alors la valeur d’observation se trouve dans l’intervalle de la classe médium. Elle se traduit par l’observation de l’état M avec une probabilité de 1 et l’état B avec une probabilité de 0 et l’état F avec une probabilité de 0. Sa représentation est donnée par le vecteur d’observation floue suivant :

$$— e = \langle P(TAp_1 = B | o_1), P(TAp_1 = M | o_1), P(TAp_1 = F | o_1) \rangle.$$

— $e = \langle 0, 1, 0 \rangle$.

Afin de tenir compte du flou lié à la définition des ensembles d'états D de la variable *TestionArtrielle_P_1*. Nous devons, tout d'abord associer chaque état avec sa propre fonction d'appartenance. Cette étape est nommée la fuzzification. En effet, le résultat de la fuzzification est illustré dans le tableau suivant.

Etat	Type et paramètres de la fonction
Base	Trapezoid("Base ",0,0,45,65)
Medium	Trapezoid("Medium ",45,65,75,90)
Forte	Trapezoid("Forte ",75,90,200,200)

TABLE 4.2 – La fuzzification du nœud TA_{p1} .

Cette observation sera considérée comme floue. En conséquence, nous pouvons constater que les degrés d'appartenance aux les états B, M et F sont respectivement à $\mu_{Base}(83)$, $\mu_{Medium}(83)$ et $\mu_{Forte}(83)$. En utilisant les fonctions d'appartenance liées aux états illustrées dans la figure 4.6, nous constatons que :

- $\mu_{Base}(83) = 0.$,
- $\mu_{Medium}(83) = 0.7,$
- $\mu_{Forte}(83) = 0.3.$

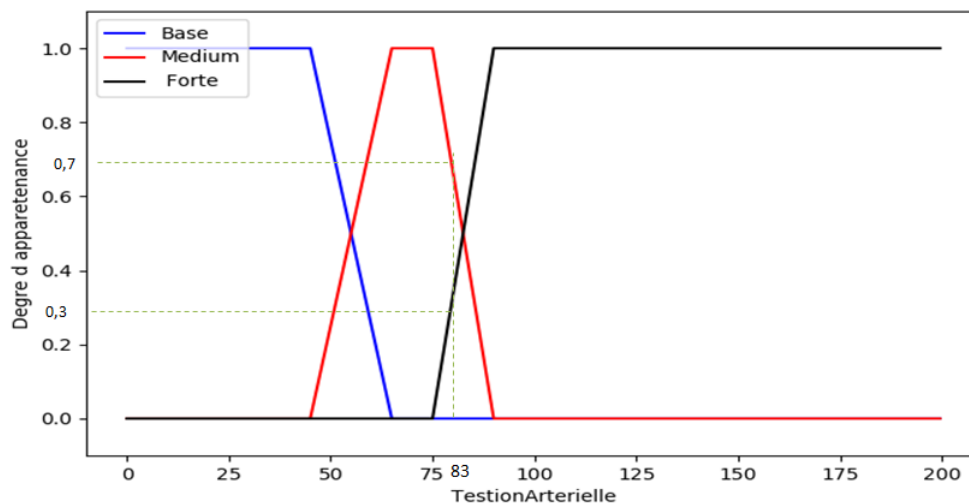


FIGURE 4.6 – Représentation floue du nœud TA_{p1} .

Cette observation est représentée par un vecteur fe contenant les degrés d'appartenance de l'observation de chaque état. Ainsi, cette observation est illustrée dans la figure 4.4 (b).

- $fe = \langle \mu_{Base}(83), \mu_{Medium}(83), \mu_{Forte}(83) \rangle$.
- $fe = \langle 0, 0.7, 0.3 \rangle$.

Les probabilités a posteriori du noeud TA_{p1} sont les suivantes :

$$\begin{aligned}\lambda &= \frac{1}{P(TA_{p1} = B) \cdot \mu_B(83) + P(TA_{p1} = M) \cdot \mu_M(83) + P(TA_{p1} = F) \cdot \mu_F(83)} \\ &= \frac{1}{0 + 0,245 + 0,135} \\ &= 2,63\end{aligned}\tag{4.2}$$

$$\begin{aligned}P(TA_{p1} = B \mid o_1) &= \lambda \cdot P(TA_{p1} = B) \cdot \mu_{Base}(83) \\ &= 2,63 \cdot 0 \\ &= 0\end{aligned}\tag{4.3}$$

$$\begin{aligned}P(TA_{p1} = M \mid o_1) &= \lambda \cdot P(TA_{p1} = M) \cdot \mu_{Medium}(83) \\ &= 2,63 \cdot 0,35 \cdot 0,7 \\ &= 0,64\end{aligned}\tag{4.4}$$

$$\begin{aligned}P(TA_{p1} = F \mid o_1) &= \lambda \cdot P(TA_{p1} = F) \cdot \mu_{Forte}(83) \\ &= 2,63 \cdot 0,45 \cdot 0,3 \\ &= 0,36\end{aligned}\tag{4.5}$$

4.3.3 Inférence dans FzMEBN

Afin de proposer et mettre en œuvre un formalisme de représentation et de raisonnement avec la connaissance probabiliste et la connaissance floue simultanément, il s'avère indispensable de développer un modèle qui permet à la fois, de représenter formellement et explicitement ces connaissances. Ce modèle doit permettre de décrire formellement le monde réel et supporter le raisonnement probabiliste basé sur des connaissances floues en calculant des probabilités à posteriori pour certaines entités cibles en se basant sur un ensemble d'observations floues sur d'autres entités.

Dans cette optique, nous proposons une solution pour offrir un mécanisme d'inférence probabiliste basé sur les observations floues dans FzMEBN. Notre objectif est d'exploiter la connaissance floue modélisée dans FzMEBN pour améliorer ces capacités de raisonnement. Ainsi, nous avons opté pour un processus général afin d'offrir une inférence probabiliste basée sur des observations floues. En effet, notre démarche consiste à générer un SSBN classique, ensuite fuzzifier ce dernier et exploiter la technique d'évidence virtuelle pour incorporer les des observations floues lors de l'inférence probabiliste. Par ailleurs, elle comprend quatre étapes telles qu'elles sont illustrées dans la figure 4.7 :

- Étape 1 : Générer un SSBN flou.

- Étape 2 : Déterminer les vecteurs des observations floues.
- Étape 3 : Ajouter les nœuds virtuels.
- Étape 4 : Lancer l'inférence probabiliste floue.

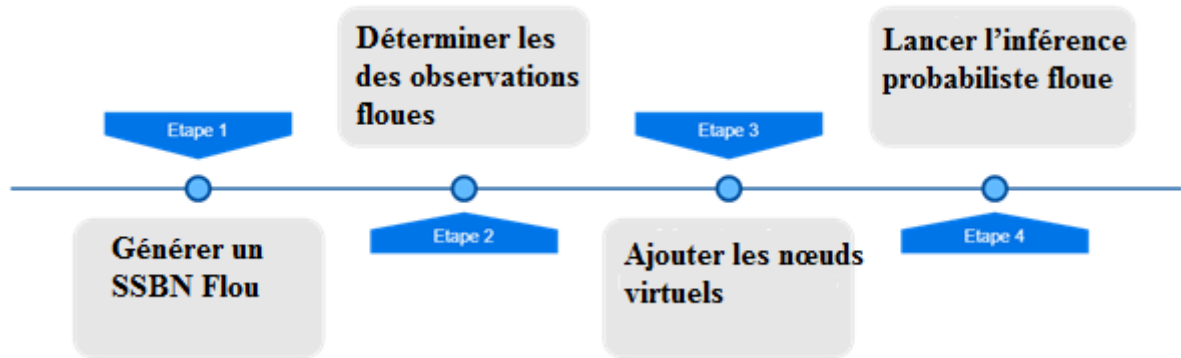


FIGURE 4.7 – Le processus général de l'inférence dans FzMEBN.

Pour pouvoir lancer l'exécution de l'inférence probabiliste floue dans FzMEBN il faut introduire un ensemble de nœuds de requêtes Q et un ensemble de nœuds d'observations E représentés dans la base de connaissance K .

4.3.3.1 Étape 1 : Générer un SSBN Flou

Cette phase vise à générer un réseau bayésien flou spécifique à la situation enregistrée. Cette phase est déclinée en actions présentées ci-dessous :

- Générer un SSBN standard avec des observations certaines ;
- Considérer nœuds qui sont instanciés à partir des nœuds résidants flous comme des nœuds flous ;
- Supprimer les observations certaines pour chaque nœud flou ;
- Attacher les nœuds flous par des fonctions d'appartenance pour fuzzifier leurs états.

Le pseudo-code suivant « Algorithm 3 » formalise la première phase du processus d'inférence pour la génération d'un SSBN flou pour traiter une situation particulière enregistrée dans une base de connaissances K . En effet, il s'agit de créer un SSBN standard dans un premier temps, un réseau bayésien standard spécifique à une situation particulière. Ensuite, ce SSBN sera fuzzifié dans la deuxième étape. L'algorithme prend en entrée un réseau bayésien multi-entités flou m , une liste L qui contient les nœuds requêtes, c'est à dire, les nœuds cibles de la requête d'analyse et FL représente tous les nœuds résidents flous du réseau.

En effet, nous devons tout d'abord construire un SSBN en se basant sur des observations certaines. Par la suite, il s'agit de supprimer les observations certaines incorporées dans les nœuds flous du SSBN et les remplacer par des observations floues (le remplissage des observations floues sera effectué dans l'étape 3). Ainsi, la ligne numéro 2 vise à construire un SSBN

classique en appliquant l'algorithme de laskey [74], la boucle "for" définie après a pour objectif de fuzzifier cet SSBN. La liste de nœuds flous avec leurs fonctions d'appartenance sera construite en se basant sur l'hypothèse suivante : « les nœuds instanciés à partir d'un nœud flou sont considérés aussi comme flous ». La ligne 05 vise à tester cette hypothèse, ensuite, si elle est vérifiée, c'est-à-dire, le nœud est flou, les observations certaines du nœud en cours de traitement seront supprimées (la ligne 06) et on le rattache avec ses propres fonctions d'appartenance afin de fuzzifier ses états.

Algorithm 3: Générer un SSBN Flou

Input: Liste des nœuds Requêtes : Q ; knowledge base : k ; FzMEBN : m ; liste des nœuds residents flous : FL

Output: SSBN flou Net ; Fuzzy Nodes List : Γ

```

1  $\Gamma \leftarrow \emptyset$ 
2  $Net \leftarrow LaskeySSBNGenerator(Q, k, m)$ 
3 foreach  $n_i \in Ne$  do
4    $R \leftarrow getResidentNodeName(n_i)$ 
5   if  $(R \in FL)$  then
6     supprimer l'observation certaine de  $n_i$ 
7     Attacher  $n_i$  avec ses fonctions d'appartenance
8      $\Gamma \leftarrow \Gamma \cup n_i$ 

```

A la fin de cette phase, on obtient un SSBN Flou Net et une liste Γ contenant les nœuds flous qui sont attachés à des fonctions d'appartenance.

4.3.3.2 Étape 2 : Déterminer les vecteurs d'observations floues.

L'inférence bayésienne floue dans FzMEBN vise à répondre à des requêtes pour calculer le degré de la croyance de certaines entités cibles étant donné les observations floues et précises pour d'autres entités. Le but ici est de construire une liste globale contenant tous les vecteurs des observations floues, pour les nœuds flous afin de les incorporer dans l'inférence bayésienne dans la prochaine phase. Ainsi, l'algorithme 4 formalise les étapes de cette phase.

L'ensemble $\Psi = \{fen_1, fen_2, \dots, fen_n\}$ représente le résultat de cet algorithme, il contient tous les vecteurs des observations floues relatifs à chaque nœud flou. Tels que :

- $feni = \langle \mu_{s_1}(x), \mu_{s_2}(x), \dots, \mu_{s_m}(x) \rangle$, représente un vecteur d'observation floue, qui contient les degrés d'appartenance pour chaque état s_j étant donné la valeur d'observation du nœud $_i$ est x .
- n représente le nombre d'observations floues.

Algorithm 4: Déterminer les vecteurs d'observations floues

Input: L : Une liste de valeurs observées; Γ : Une liste de nœuds flous

Output: Ψ : Un ensemble contenant tous les vecteurs d'observations floues

```

1  $\Psi \leftarrow \emptyset$ 
2 foreach  $n_i \in \Gamma$  do
   | /* récupérer la valeur observée pour le nœud i */
3    $ov_i \leftarrow L.getValue(n_i)$ 
   | /* Calculer les degrés d'appartenance de chaque état  $j \in s_i$  */
4    $fen_i \leftarrow \langle \mu_{s_1}(ov_i), \mu_{s_2}(ov_i), \dots, \mu_{s_m}(ov_i) \rangle$ 
5   if ( $fen_i$  n'est pas normalisé) then
6     | Normalisation ( $fen_i$ ); /* Faire appel à l'algorithme 5 */
7    $\Psi \leftarrow \Psi \cup fen_i$ 

```

Dans la plus part des cas, le vecteur d'observation floue n'est pas normalisé. C'est à dire, la somme des degrés d'appartenance d'un nœud donné pour une observation donnée n'est pas égale à 1.

Soient n_i, S, fen_i et U des variables qui représentent respectivement un nœud flou, l'ensemble de ses états, son vecteur d'observation floue et l'univers du discours du nœud n_i .

Dans le cas où, $\sum_{j=0}^m \mu_{s_j}(x) = 1$, $\forall s_j \in S$ et $\forall x \in U$, alors l'espace de n_i est bien défini et garanti que la propriété des probabilités reste valide. Par contre, si cette condition n'est pas vérifiée, nous adoptons un algorithme (Algorithme 5 : Normalisation) de normalisation basé sur l'équation 6 présentée dans (Waltman et al., 2005). L'observation floue normalisée d'un état s_i est donnée par la formule ci-dessous :

$$\mu_{s_i}(x) = \frac{\mu_{s_i}(x)}{\sum_{j=0}^m \mu_{s_j}(x)} \quad (4.6)$$

Algorithm 5: Normalisation

Input: fen_i : Un vecteur d'observation floue non normalisé

Output: fen_i : Un vecteur d'observation floue normalisé

```

1  $somme \leftarrow 0$ 
2  $j \leftarrow 1$ 
3 while  $j \leq |fen_i|$  do
4   |  $somme \leftarrow somme + fen_i[j]$ 
5   |  $j \leftarrow j + 1$ 
6  $k \leftarrow 1$ 
7 while  $k \leq |fen_i|$  do
8   |  $fen_i[k] \leftarrow fen_i[k] | somme$ 
9   |  $k \leftarrow k + 1$ 

```

4.3.3.3 Étape 3 : Ajouter les nœuds virtuels

Dans cette phase, nous exploitons les résultats obtenus dans les phases précédentes. Cette phase vise à ajouter un nœud virtuel à chaque nœud flou. Par la suite, les vecteurs des observations floues élaborés dans l'étape précédente doivent être remplis dans les CPTs des nœuds virtuels pour pouvoir lancer l'inférence dans la prochaine phase. Par ailleurs, l'algorithme 6 décrit de manière formelle cette phase.

L'algorithme 6 prend en entrée un SSBN flou Net, la liste Γ contenant tous les nœuds flous et la liste ψ qui englobe toutes les observations floues pour tous les nœuds flous. Dans la ligne 2, l'algorithme commence à traiter les nœuds flous en ajoutant à chacun un nœud virtuel comme un fils. Ensuite il s'agit de récupérer le vecteur d'observation floue du nœud en cours de traitement et l'incorpore dans la CPT de son nœud virtuel en faisant appel à l'algorithme 7. Le résultat final sera un SSBN flou avec des nœuds virtuels.

Algorithm 6: Ajouter les nœuds virtuels

Input: Net : SSBN Flou; Γ : Une liste de nœuds flous; Ψ : la liste de tous les vecteurs d'observations floues

```

1 foreach  $n_i \in \Gamma$  do
  | /* créer un noeud virtuel fils pour  $n_i$  */
2    $vn_i \leftarrow L.getValue(Net, n_i)$ 
3    $fen_i \leftarrow \psi.récupérerVectObservationFloue(n_i)$ 
4   Remplissage( $vn_i, fen_i$ ) /* Faire appel à l'algorithme 7 */

```

L'algorithme 7 prend en entrée un nœud virtuel vn_i et un vecteur d'observation floue normalisé fen_i . Il s'agit dans cet algorithme de remplir la CPT du nœud vn_i (est une matrice carrée) de telle sorte que la matrice soit bi-stochastique. c'est-à-dire, la somme des lignes égale à 1 et la somme des colonnes égale aussi 1.

Algorithm 7: Remplissage

Input: vn_i : Un noeud virtuel; fen_i : Un vecteur d'observation floue.

```

1  $k \leftarrow 1$ 
2  $taille \leftarrow |fen_i|$ 
3 while  $k \leq taille$  do
4   |  $j \leftarrow 1$ 
5   | while  $j \leq taille$  do
6   |   |  $vn_i.Cpt[j][k] \leftarrow fen_i[((k + j)\%taille) + 1]$ 
7   |   |  $j \leftarrow j + 1$ 
8   |  $k \leftarrow k + 1$ 

```

4.3.3.4 Étape 4 : Lancer l'inférence probabiliste floue

Après avoir construit un SSBN flou pour la requête d'analyse, il s'agit maintenant de propager les observations certaines et les observations floues dans les réseaux bayésiens. Ceci en appliquant un algorithme d'inférence classique pour propager toutes les observations dans le SSBN flou. Dans la littérature, divers algorithmes ont traité l'inférence probabiliste dans les RBs. Dans cette étape, il s'agit d'utiliser un de ces algorithmes pour propager les observations, notamment les observations floues dans le SSBN flou. Le choix d'un algorithme dépend fortement de la contribution du développeur et de la complexité de la requête en question.

4.3.4 Exemple Illustratif

Afin de mettre en évidence les éléments de base de FzMEBN, prenons l'exemple illustré dans la figure 4.8, modélisant la relation entre la présence de l'anémie et le nombre de cellules dans le sang du patient.

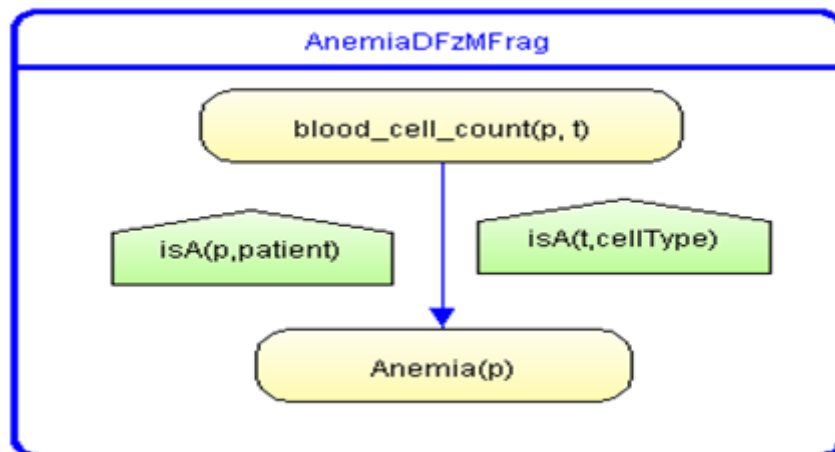


FIGURE 4.8 – Un FzMfrag pour l'anémie.

Dans cet FzMfrag, deux variables ordinaires ont été définies pour représenter des arguments aux différents nœuds. La variable p est du type *Patient* et la variable t est du type *CellType*. Ces variables ont été définies grâce à deux nœuds de contexte. En effet, le nœud $blood_cell_count(p, t)$ possède deux états qui sont *low* et *high* comme valeurs. En outre, la variable p peut représenter deux entités *red* et *white*, c'est pour cette raison nous devons attacher le nœud $blood_cell_count(p, t)$ à quatre fonctions d'appartenance comme il est montré dans les figures 4.9 et 4.10.

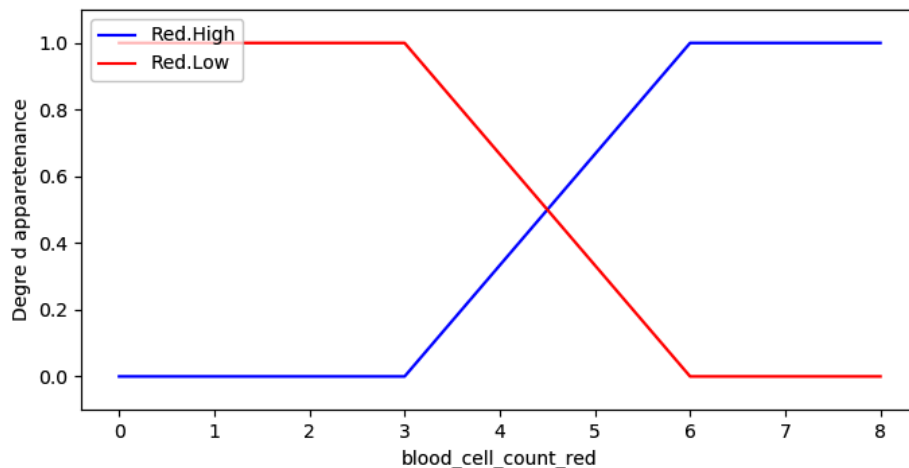


FIGURE 4.9 – Les fonctions d'appartenance attachées au nœud "blood _ cell_count(p,t)" .

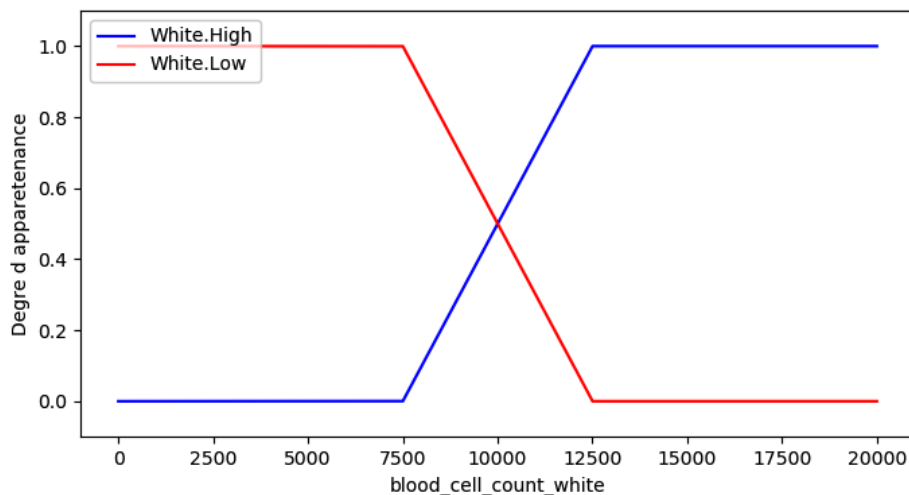


FIGURE 4.10 – Les fonctions d'appartenance attachées au nœud "blood _ cell_count(p,t)" .

Pour illustrer le processus d'inférence dans FzMEBN, on suppose que l'on a les observations suivantes dans notre base de connaissances. Soit l'entité **Amine** qui représente un patient.

$o_1 = \text{'Amine a } 3,2 \text{ millions/mm}^3 \text{ Globules rouges}'$.

$o_2 = \text{'Amine a } 7600 \text{/mm}^3 \text{ Globules blancs}'$.

Objectif : Calculer la probabilité que le patient Amine soit touché par l'anémie.

Nous avons généré dans un premier temps un SSBN flou en utilisant l'algorithme 3 en considérons le nœud Anemia comme un nœud de requête pour l'entité **Amine**, le SSBN flou généré est illustré dans la figure 4.11 (b). Il contient trois nœuds :

- Un nœud de requête *Anemia_Amine* qui est une instance du nœud résident *Anemia(p)* en substituant la variable ordinaire **p** par l'entité **Amine**.

- Le nœud *blood_cell_count_red* est résultat de l'observation o_1 . Il représente une instance du nœud résident $blood_cell_count(p,t)$ en substituant la variable ordinaire p par l'entité **Amine** et la variable t par l'entité *red*.
- Le nœud *blood_cell_count_white* est le résultat de l'observation o_2 , il représente une instance du nœud résident $blood_cell_count(p,t)$ en substituant la variable ordinaire p par l'entité **Amine** et la variable t par l'entité *white*.

Ainsi, la valeur d'observation accordée à o_1 est 3,2 millions/mm³ et la valeur d'observation de o_2 est 7600 cells /mm³. On écrit plus formellement :

- $blood_cell_count(\mathbf{Amine}, red) = 3,2 \text{ millions/mm}^3$
- $blood_cell_count(\mathbf{Amine}, white) = 7600 \text{ cells/mm}^3$.

Dans la deuxième étape, nous avons déterminé le vecteur d'observation floue *rfe* pour le nœud *blood_cell_count_red* en utilisant les fonctions d'appartenance attachées à ses états.

- $rfe = \langle \mu_{high}(3,2); \mu_{low}(3,2) \rangle$
- $rfe = \langle 0,3; 0,7 \rangle$

La même chose avec le vecteur d'observation floue *wfe* pour le nœud *blood_cell_count_white* comme suit :

- $wfe = \langle \mu_{high}(7600); \mu_{low}(7600) \rangle$
- $wfe = \langle 0,1; 0,9 \rangle$

Dans la troisième étape, nous avons ajouté les nœuds virtuels pour chaque nœud flou. En effet, les nœuds V1 (figure 4.11 (c)) et V2 (figure 4.11 (c)) ont été ajoutés pour stocker les vecteurs des observations floues relatives respectivement aux observations o_1 et o_2 . Les tableaux 4.3 et 4.4 présentent les tables des probabilités conditionnelles respectivement pour le nœud V1 et V2.

V1 blood_cell_count_red	High	Low
High	0.3	0.7
Low	0.7	0.3

TABLE 4.3 – La table des probabilités conditionnelles V1

V2 blood_cell_count_white	High	Low
High	0.1	0.9
Low	0.9	0.1

TABLE 4.4 – La table des probabilités conditionnelles V2

Finalement, dans la dernière étape nous appliquons un algorithme d'inférence classique pour propager les observations dans tous le SSBN Flou. Dans notre exemple nous avons exécuté l'algorithme Arbre de jonction, les résultats obtenus sont indiqués dans la figure 4.11(c). D'après

ces résultats, nous pouvons constater que le patient **Amine** développe l'anémie avec une probabilité de 79.4%.

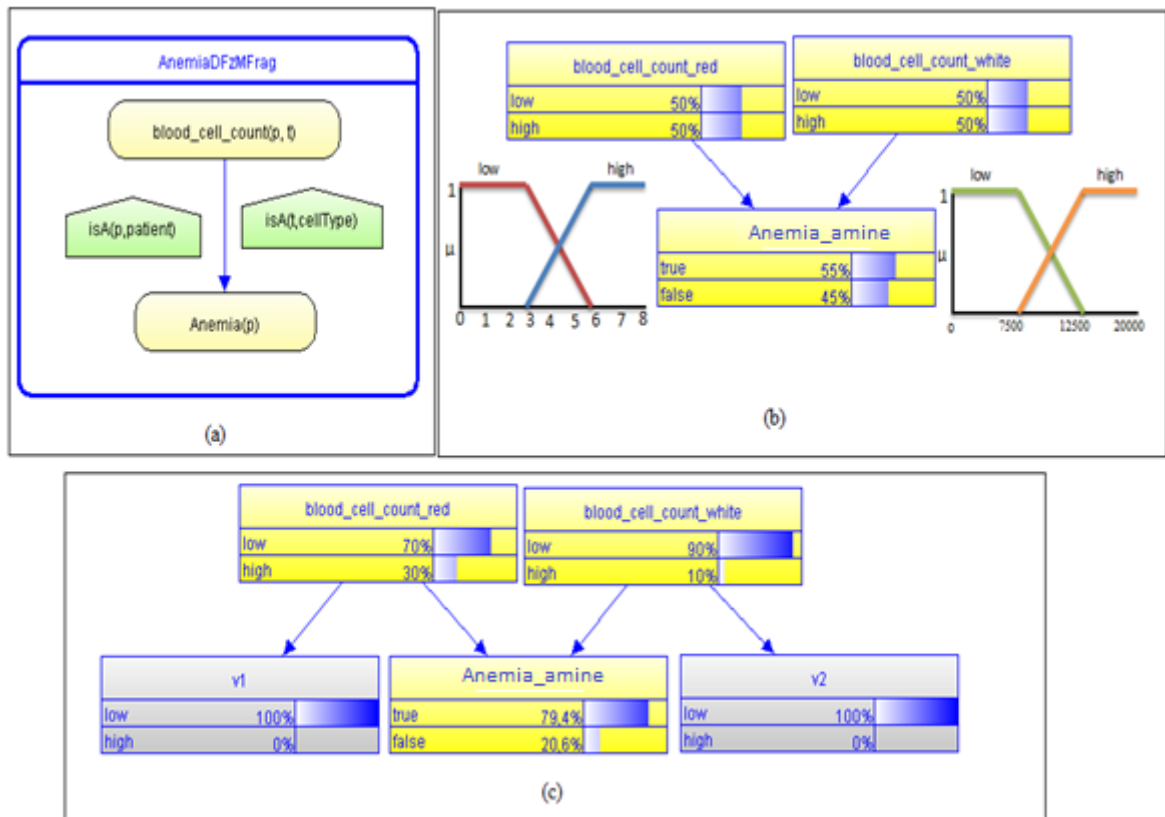


FIGURE 4.11 – Illustration du processus d'inférence.

4.4 Synthèse

La solution que nous proposons dans ce chapitre vise à permettre la modélisation et le raisonnement avec des connaissances probabilistes dans MEBNs en tenant compte la connaissance floue. Afin de représenter la sémantique floue des variables aléatoires dans MEBNs, une distinction entre les variables floues et les variables précises a été proposée pour construire des FzMFrags (MFraggs flous). Ces FzMFrags flous peuvent être vus comme des motifs de réseaux bayésiens flous qu'on peut instancier complètement ou partiellement plusieurs fois selon les situations observées. Associée à cette modélisation, nous proposons un processus général pour fournir une inférence probabiliste basée sur des observations floues. Le travail présenté dans [77, 78] vise à considérer la connaissance floue dans les MEBNs. Il s'agit d'un MEBN attaché à des règles d'inférences floues. Ce travail présente plusieurs inconvénients, notamment :

- Dans la modélisation, leur approche ne fait aucune distinction entre les éléments précis et flous du réseau, ce qui va compliquer le processus d'inférence probabiliste floue par la suite.

- L'inférence probabiliste floue proposée dans cette approche est basée sur des formules pondérées et l'algorithme l'arbre de jonction modifié pour propager les observations floues. Cette restriction peut poser de vrais problèmes quand la complexité du problème augmente, c'est-à-dire la taille du réseau augmente, car l'algorithme de l'arbre de jonction est un algorithme d'inférence exacte qui est un problème NP-complet.

En effet, l'originalité de notre approche par rapport au travail de [77, 78] réside dans les caractéristiques suivantes :

- Nous modélisons les connaissances comme des fragments de réseaux bayésiens flous. Autrement dit, notre approche permet de modéliser des structures répétitives des RBs flous. Les nœuds flous sont étiquetés et attachés par des fonctions d'appartenance.
- Les fragments des réseaux bayésien flous peuvent être instanciés pour générer des réseaux bayésiens flous selon la situation observée.
- L'inférence probabiliste floue est basée sur la méthode d'évidence virtuelle qui est une méthode générale et ne nécessite aucun changement et aucune personnalisation dans les algorithmes d'inférences.

La figure 4.12 synthétise le processus de la modélisation et du raisonnement dans FzMEBN et MEBN ; l'avantage de FzMEBN par rapport au MEBN classique réside dans le fait qu'il tient compte la connaissance probabiliste et la connaissance floue associées à ses nœuds dans l'étape de la modélisation. En outre, il présente un processus générique pour l'inférence probabiliste floue en incorporant les observations floues dans les probabilités.

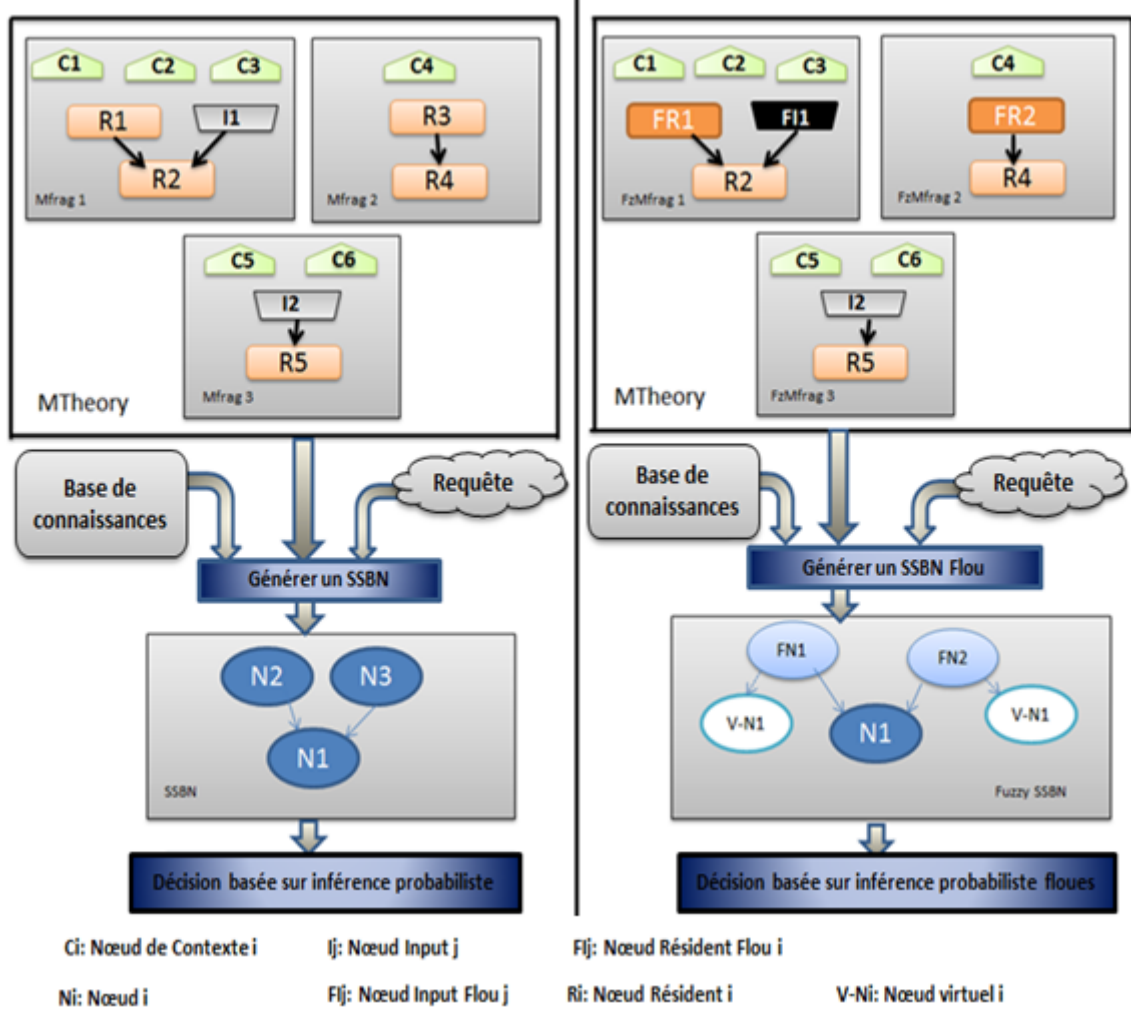


FIGURE 4.12 – Différence entre MEBN et FzMEBN.

4.5 Conclusion

Dans ce chapitre, nous avons exposé une approche pour faire face au problème de la représentation et le raisonnement avec les connaissances floues dans les MEBNs. Nous avons proposé une extension des MEBNs afin d’améliorer les MEBNs classiques et permettre la représentation de la connaissance floue de manière expressive. Nous avons proposé également un processus générique pour faire le raisonnement probabiliste basé sur des observations floues. En tant que caractéristique de notre approche c’est qu’elle est applicable dans n’importe quel domaine où les connaissances probabilistes et floues sont liées.

Dans le chapitre suivant, nous allons exploiter notre approche FzMEBN pour faire face à l’incertitude dans des ontologies probabilistes formalisées dans le langage PR-OWL 2.

Chapitre 5

Approche pour la gestion de la connaissance floue dans les ontologies probabilistes

“The process of scientific discovery is, in effect, a continual flight from wonder”

— Albert Einstein

Sommaire

5.1	Introduction	95
5.2	Contexte et motivation	96
5.3	Ontologie probabiliste floue	98
5.4	FuZzy-PRWOL2	99
5.4.1	Sémantique de FuZzy-PR-OWL2	100
5.4.2	Raisonnement dans l’ontologie probabiliste floue	108
5.4.3	Exemple Illustratif	110
5.5	Comparaison avec d’autres approches	115
5.6	Conclusion	118

5.1 Introduction

Le Web sémantique a pour objectif de rendre le contenu de World Wide Web courant compréhensible et traitable par des machines. En effet, le Web sémantique est basé principalement sur les capacités des ontologies qui permettent de modéliser des conceptualisations partagées dans un langage formel.

Le langage ontologique probabiliste PR-OWL 2 est basé sur la logique des réseaux bayésiens multi-entités pour remédier aux défauts des ontologies classiques. Il permet de modéliser la connaissance probabiliste et l’incorporer sous forme de dépendances et des régularités statistiques dans les ontologies classiques. Au cours des dernières années, beaucoup de travaux se

sont basés sur le langage PR-OWL 2 et ont prouvé leurs efficacités afin de modéliser et faire des raisonnements à base sémantique incertaine. Le langage PR-OWL 2 quant à lui ne fournit pas des alternatives pour modéliser et de faire des raisonnements avec la connaissance floue qui caractérise la plus part des domaines du monde réel.

Dans ce travail de thèse, nous nous intéressons au problème de la représentation et du raisonnement avec l'incertitude dans les ontologies riches en termes d'incertitude. En effet, notre démarche vise à considérer la connaissance probabiliste et la connaissance floue qui sont impliquées simultanément dans les ontologies classiques. C'est pour cette raison, nous proposons une extension du langage probabiliste PR-OWL 2 pour augmenter ses capacités de raisonnement et de la modélisation. En effet, notre idée principale est d'étendre le langage PR-OWL 2 pour prendre en charge les connaissances floues. Ceci en mettant à jour son ontologie de haut-niveau en ajoutant de nouvelles classes et propriétés en permettant de représenter la sémantique de FzMEBN et fournir une inférence probabiliste floue basée sur des observations floues.

Le présent chapitre a pour objectif de présenter notre deuxième contribution. Dans la première section, nous présentons le contexte et la motivation de notre travail. Dans la deuxième section nous introduisons les éléments de base de notre nouvelle extension pour la modélisation et le raisonnement avec la connaissance floue représentée dans le langage PR-OWL 2. Ensuite, nous discutons notre approches par rapport aux travaux développés dans l'état de l'art présentés dans le chapitre 3.

5.2 Contexte et motivation

L'émergence des technologies du Web sémantique, notamment les langages ontologiques a donné naissance à de nombreuses applications. Malgré le succès indéniable de ces langages dans des applications réelles dans plusieurs domaines afin de représenter les connaissances, de les partager et d'assurer l'interopérabilité entre différents systèmes. Ces langages restent primitifs et nécessitent d'être enrichis afin de représenter le monde réel comme il se doit en tenant en compte l'incertitude. Ceci est dû au fait que ces langages sont basés sur des Frameworks fondés sur une logique booléenne.

Selon la taxonomie présentée dans [15], l'incertitude dans une ontologie est attachée à deux catégories de composants comme illustré dans la figure 5.1, les composants bien-définis et les composants mal-définis. Sur la base de l'ontologie d'incertitude proposée par le W3C, plusieurs extensions ont été proposées dans la littérature afin de traiter l'incertitude dans les ontologies. La plupart d'entre elles reposent sur des fondements mathématiques, notamment les RBs et la logique floue.

D'une part, les ontologies floues ont été développées afin de faire face à la connaissance floue dans les ontologies classiques. Elles s'appuient sur la logique floue pour traiter les composants mal-définis des ontologies.

D'autre part, les ontologies probabilistes ont été proposées afin de remédier aux défauts des ontologies classiques, visant la permission de la modélisation de la connaissance probabiliste liée aux composants bien-définis dans les ontologies. Elles fournissent des taches de raisonnement basées sur les capacités des réseaux bayésiens.

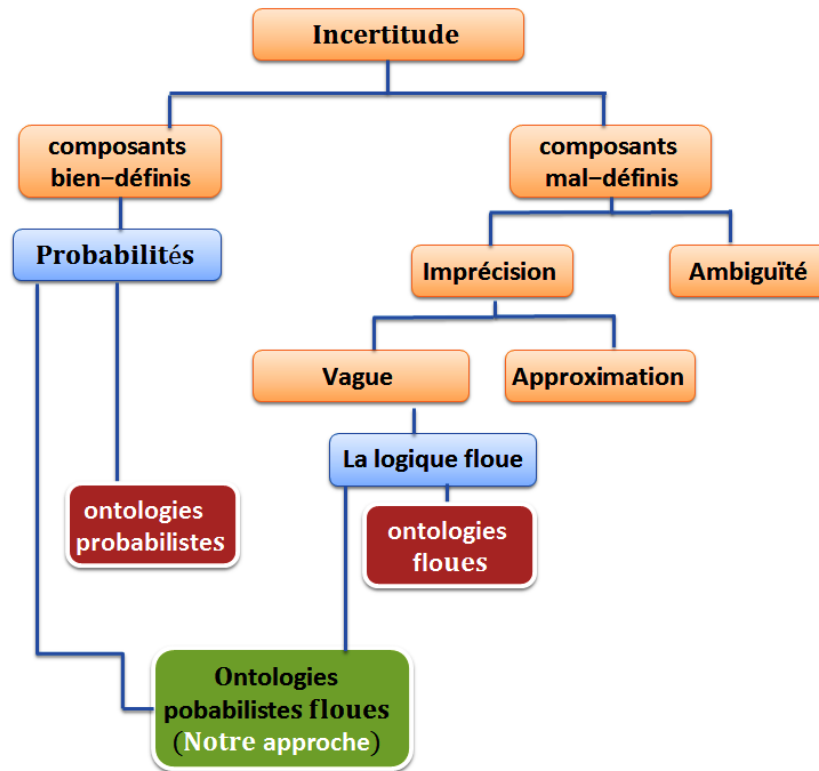


FIGURE 5.1 – Positionnement de notre approche.

Par ailleurs, la combinaison de la logique floue et les RBs a prouvé son efficacité dans plusieurs domaines tel que, l'analyse de risques, les systèmes de recommandation, etc. Néanmoins, aucune attention n'est portée à l'utilisation des modèles hybrides pour faire face à l'incertitude du Web sémantique. Plusieurs formes d'incertitude peuvent apparaître simultanément dans un domaine bien particulier où les connaissances probabilistes et les connaissances floues sont très souvent liées. Par conséquent, il s'avère nécessaire d'adopter des modèles hybrides combinant la logique floue et les RBs qui doivent être riches, expressifs et puissants en termes de modélisation et du raisonnement afin de bien gérer l'incertitude de notre monde en traitant les composants bien-définis et les composants mal-définis simultanément dans les ontologies.

PR-OWL 2 est l'une des solutions candidates pour faire face à la connaissance probabiliste qui traite les composants bien-définis dans les ontologies. Les ontologies probabilistes formalisées en PR-OWL 2 sont robustes car elles se basent sur les capacités des MEBNs et ces derniers présentent plusieurs avantages à savoir :

- Partage plusieurs similitudes avec les ontologies et conviennent pour représenter des systèmes complexes, puisqu'ils introduisent plusieurs aspects de la modélisation à savoir : un

niveau d'expressivité très élevé fondé sur la logique du premier ordre pour mieux représenter le monde réel et la réalité perçue. En effet, MEBN modélise le problème en question sous forme d'un ensemble de fragments (sous problèmes) qui peuvent être exploités de manière globale ou partielle.

- Représente un outil simple de modélisation rapide et efficace.
- Permettent une représentation plus compacte des connaissances et surtout lorsqu' il s'agit d'un problème qui contient beaucoup de structures de connaissances répétitives. Donc, il offre un moyen de réutilisation des connaissances déjà modélisées.
- Permettre ainsi au concepteur le choix entre visualiser globalement son système ou entrer dans les détails de la modélisation d'un sous-système bien précis. Autrement dit, il permet de manipuler partiellement ou complètement les connaissances modélisées dans le modèle.
- Offre des mécanismes d'inférences très flexibles en se basant sur la génération des SSBNs, des RBs standards qui représentent une situation précise enregistrée et capturée par un observateur dans le système.
- La modularité, les MFragments peuvent être facilement ajoutés ou supprimés du système modélisé sans aucune perte dans la cohérence structurelle du réseau.

Malgré ces avantages, PR-OWL 2 reste incapable de traiter les composants mal-définis car il se base sur les MEBNs qui ne tiennent pas en considération des variables floues du domaine. Par ailleurs, FzMEBN (Riali et al., 2017) est une extension floue aux réseaux bayésiens multi-entités classiques permettant de gérer le flou en se basant sur la logique floue. Ce modèle est utile pour traiter l'incertitude due à l'incomplétude, au caractère aléatoire et tient compte également des connaissances vagues.

Nous proposons dans le reste de ce chapitre une nouvelle approche pour faire face aux composants mal-définis et aux composants bien-définis simultanément. En effet, nous proposons de modéliser la connaissance floue dans les ontologies probabilistes en se basant sur FzMEBN.

5.3 Ontologie probabiliste floue

Une des lacunes des ontologies c'est qu'elles sont basées sur une logique booléenne et déterministe qui n'est pas capable de faire face aux connaissances mal-structurées, floues ou probabilistes impliquées dans les applications du monde réel.

Pour une meilleure manipulation de la connaissance incertaine dans les ontologies. Nous nous intéressons dans notre travail à l'intégration de la logique floue dans l'ontologie probabiliste pour définir un nouveau paradigme théorique appelé ontologie probabiliste floue (*FPO : Fuzzy Probabilistic Ontology*) capable de gérer la connaissance floue et la connaissance probabiliste simultanément. En effet, une quantification du flou lié aux connaissances par l'utilisation de la logique floue sera exploitée pour améliorer la puissance des ontologies probabilistes afin

qu'elles prennent en considération la connaissance floue et probabiliste à la fois dans les ontologies. Ainsi, une FPO est une extension qui améliore les ontologies probabilistes en intégrant des modèles hybrides (réseaux bayésiens flous). Elles combinent à la fois la logique floue et la théorie des probabilités dans les ontologies pour bénéficier des avantages de chaque formalisme.

Avant de présenter les détails de notre extension, nous allons définir notre nouveau paradigme de manière générale par la mise à jour de la définition de l'ontologie probabiliste proposée par Costa et al [93].

Définition 5.3.4 *une ontologie probabiliste floue est une représentation de la connaissance formelle explicite qui exprime des connaissances sur un domaine d'application. Elle comprend :*

- *Les types des entités existant dans le domaine ;*
- *Les propriétés de ces entités ;*
- *Les relations entre les entités ;*
- *Les processus et les événements qui se passent avec ces entités ;*
- *Les régularités statistiques qui caractérisent le domaine ;*
- *Les connaissances peu concluantes, ambiguës, incomplètes, **imprécises**, **vagues**, peu fiables, et dissonantes liées aux entités du domaine ;*
- *L'incertitude sur toutes les formes de connaissance citées ci-dessus .*

Ainsi, le terme entité fait référence à tout concept réel ou fictif, concret ou abstrait qui est caractérisé par son unicité.

5.4 FuZzy-PRWOL2

L'ontologie probabiliste floue proposée dans notre étude est principalement basée sur l'extension du langage probabiliste PR-OWL 2 pour traiter les connaissances vagues. Il s'agit d'un langage probabiliste très flexible, expressif et puissant. Bien que PR-OWL 2 a prouvé son efficacité dans de nombreux domaines où l'incertitude est prédominante, il ne peut pas gérer les connaissances vagues qui apparaissent dans l'ontologie.

Nous proposons donc dans les sections suivantes une nouvelle extension de PR-OWL 2 en tant que langage Web d'ontologies probabilistes floues. Notre idée principale est d'étendre les ontologies probabilistes formalisées dans PR-OWL2 par la substitution de la sémantique de MEBN par la sémantique de notre modèle FzMEBN afin d'augmenter ses capacités de modélisation et de raisonnement. Notre extension est nommée FuZzy-PR-OWL 2. Notre démarche vise à mettre à jour l'ontologie de haut niveau de PR-OWL 2 classique en ajoutant de nouveaux composants afin de représenter la sémantique de FzMEBN. Il convient de noter que FuZzy-PR-OWL 2 étend PR-OWL 2 au lieu de le remplacer. i.e, FuZzy-PR-OWL 2 est une extension de PR-OWL 2. Nous illustrons dans la figure 5.2 le processus proposé.

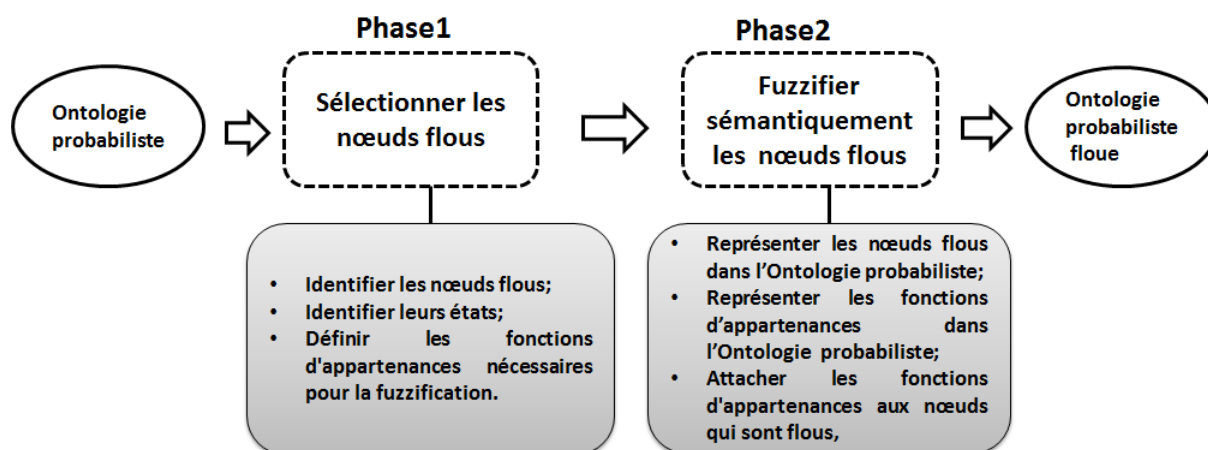


FIGURE 5.2 – Le processus proposé pour la fuzzification des ontologies probabilistes.

Le but ultime étant de fuzzifier l'ontologie probabiliste pour tenir compte du flou lié à ses composants. pour ce faire, deux phases sont requises :

Phase 1 : l'objectif de cette phase est de déterminer les composants qui présentent du flou, elle se déroule en trois activités successives :

- Identifier les noeuds qui sont flous,
- Identifier leurs états ,
- Définir les fonctions d'appartenances pour fuzzifier les états.

Phase 2 : cette phase est constituée de trois activités successives :

- Représenter les noeuds qui présentent du flou sémantiquement comme des instances de la classe *FuzzyResidentNode* et de la classe *FuzzyInputNode*.
- Représenter les fonctions d'appartenance comme des instances de la classe *MembershipFunction*.
- Lier les fonctions d'appartenance avec les noeuds qui sont flou via la propriété d'objets *isAttachedWith*.

Nous illustrons dans ce qui suit en détails notre démarche en mettant l'accent sur les nouvelles classes ajoutées pour modéliser sémantiquement le flou attaché aux composants de l'ontologie probabiliste.

5.4.1 Sémantique de FuZzy-PR-OWL2

La figure 5.3 met l'accent sur les classes que nous avons ajoutées dans l'ontologie de haut-niveau. Ces classes (sont en couleur verte) permettent la modélisation de la connaissance floue de l'ontologie probabiliste. Par ailleurs, nous présentons les classes de l'ancienne ontologie de haut-niveau de PR-OWL 2 qui sont en couleur grise dans l'annexe A.

Comme illustré dans la figure 5.4, l'ontologie probabiliste floue (FuZzy-PR-OWL 2) est modélisée à l'aide de la classe MTheory qui comprend un ensemble de MFrag flous (FuZzyMFrag) $FzM = \{FzM_1, FzM_2, \dots, FzM_n\}$. En effet, deux types de MFrag flous sont distingués :

- Un ensemble de MFrag flous de domaines $FzFM$ (FuZzyDomainMFrag), avec :
 $FzDM = \{FzDM_1, FzDM_2, \dots, FzDM_a\}$,
- Un ensemble de MFrag flous d'observations $FzFM$ (FuZzyFindingMFrag), avec :
 $FzFM = \{FzFM_1, FzFM_2, \dots, FzFM_b\}$.

Où : $FzFM \subseteq FzM$, $FzDM \subseteq FzM$ et $FzFM \cup FzDM = FzM$.

En effet, la classe FuZzyFindingMFrag est une sous-classe de la classe FuZzyMFrag qui permet de représenter les informations sur les observations. La classe FuZzyDomainMFrag est une sous-classe de la classe FuZzyMFrag qui comprend tous les MFrag flous spécifiques au domaine. Les membres de cette classe sont les MFrag flous créés par l'ingénieur ontologique.

Chaque $FzDM_i \in FzDM$ avec $1 \leq i \leq a$ est construit à partir d'un ensemble de nœuds X pouvant être précis ou flous. La sémantique des MFrag Flous est exprimée grâce à un ensemble de classes et propriétés qui sont :

- Un ensemble de nœuds de contextes dénotés par : $C = \{C_1, C_2, C_3, \dots, C_l\} \subseteq X$.
- Un ensemble de nœuds résidents dénotés par : $R = \{R_1, R_2, R_3, \dots, R_m\} \subseteq X$.

Un nœud résident peut définir la distribution d'une variable aléatoire au sein du FzM-Frag où il appartient, un nœud résident est représenté comme un individu de la classe **ResidentNode**.

- Un ensemble de nœuds d'entrée dénotés par : $I = \{I_1, I_2, I_3, \dots, I_p\} \subseteq X$.
- Un ensemble de nœuds résidents flous dénotés par : $F_R = \{F_{R1}, F_{R2}, F_{R3}, \dots, F_{Rk}\} \subseteq X$
- Un ensemble de nœuds d'entrée flous dénotés par : $F_I = \{F_{I1}, F_{I2}, F_{I3}, \dots, F_{It}\} \subseteq X$

Avec, $C \cup R \cup I \cup F_R \cup F_I = X$.

- **Un nœud de contexte** est une variable aléatoire qui vise à définir le contexte et les conditions qui doivent être vérifiés pour que les distributions de probabilités définies dans un FzM-Frag soient valides. Un nœud de contexte est représenté comme un individu de la classe **ContextNode**.
- **Un nœud résident** peut définir la distribution d'une variable aléatoire au sein du FzM-Frag où il appartient. Un nœud résident est représenté comme un individu de la classe **ResidentNode**.
- **Un nœud d'entrée** est une variable aléatoire qui influe sur la distribution des nœuds au sein du FzM-Frag où il appartient. Cependant, sa distribution est définie ailleurs dans son FzM-Frag où il est défini comme nœud résident. Un nœud d'entrée est représenté comme un élément de la classe **InputNode**.

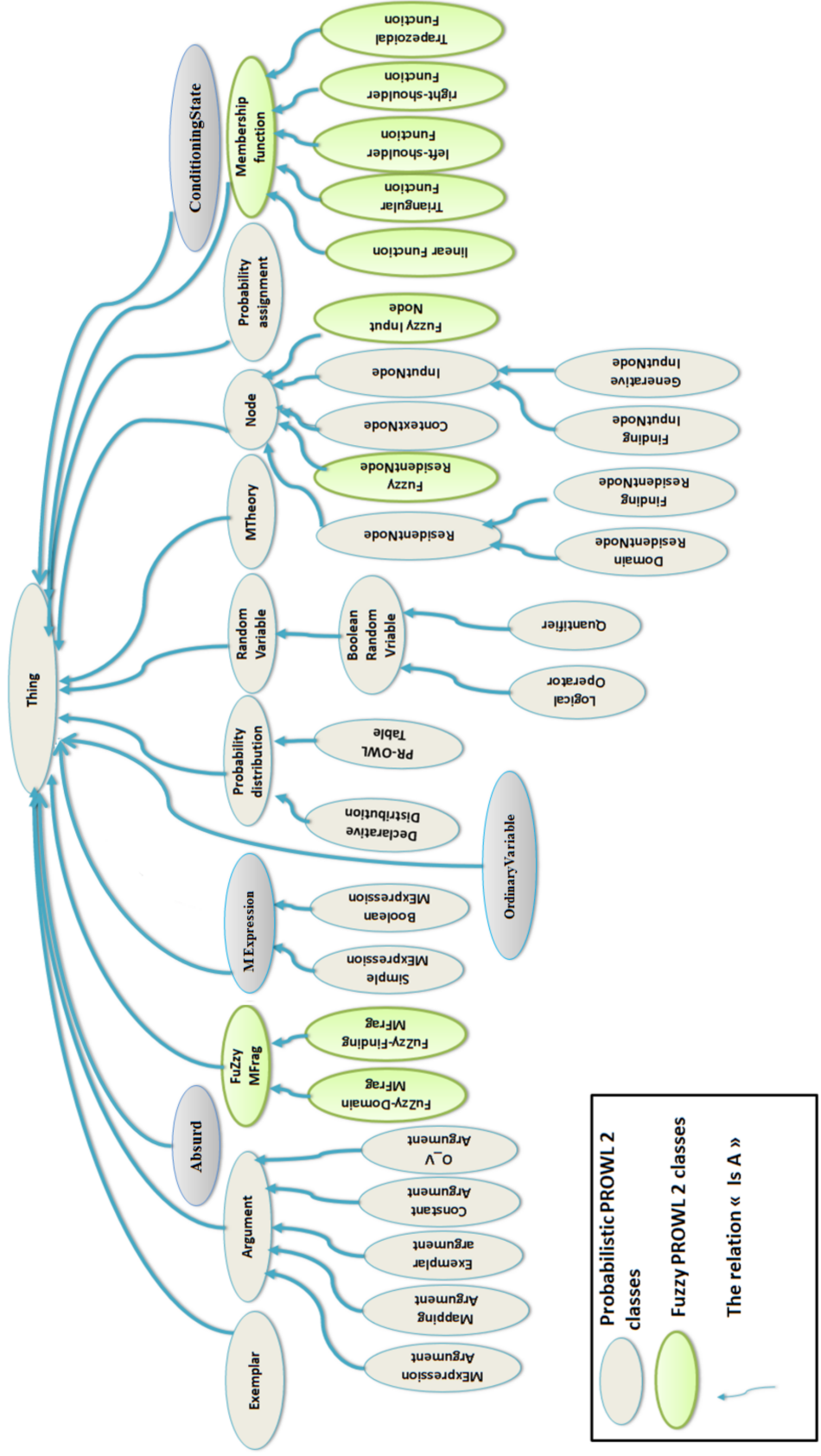


FIGURE 5.3 – L'ontologie de haut niveau de FuZzy-PR-OWL2.

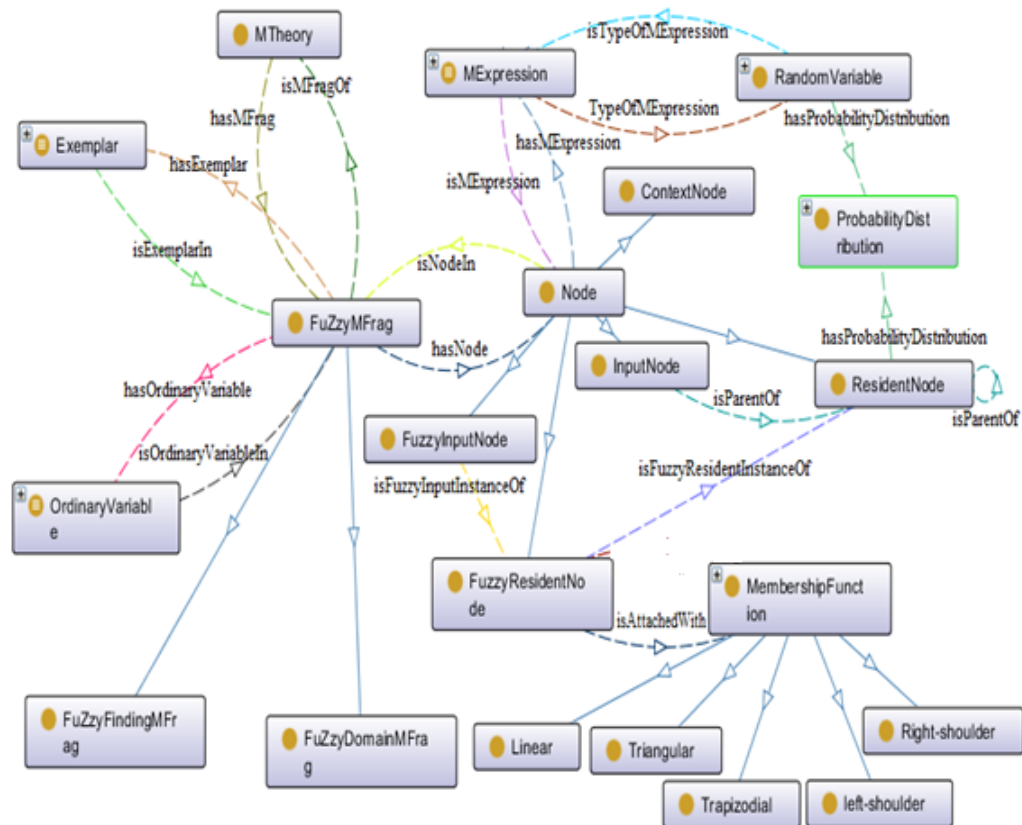


FIGURE 5.4 – Les éléments nécessaires pour définir une MThéorie.

Les classes **ContextNode**, **ResidentNode** et **InputNode** permettent de construire des ontologies probabilistes à base du MEBN. Dans les sections suivantes nous introduisons les classes ajoutées afin de capturer la connaissance floue liées aux différentes variables du domaine.

- **Les nœuds résidents flous** visent à représenter les nœuds résidents qui comportent du flou. En effet, pour chaque nœud résident R_i sélectionné comme flou, une instance IFR_i de la classe *FuzzyResidentNode* est créée pour le représenté sémantiquement. En outre, l'instance IFR_i est liée avec l'instance IR_i ¹ via une propriété d'objet appelée «*IsFuzzyResidentInstanceOf*».

Par ailleurs, l'instance IFR_i est liée avec un ensemble de fonctions d'appartenance qui seront exploiter pour fuzzifier le nœud R_i . Les liens entre ces individus sont illustrés dans la figure 5.5 .

1. L'instance IR_i est de type *ResidentNode*, elle représente la sémantique du nœud R_i .

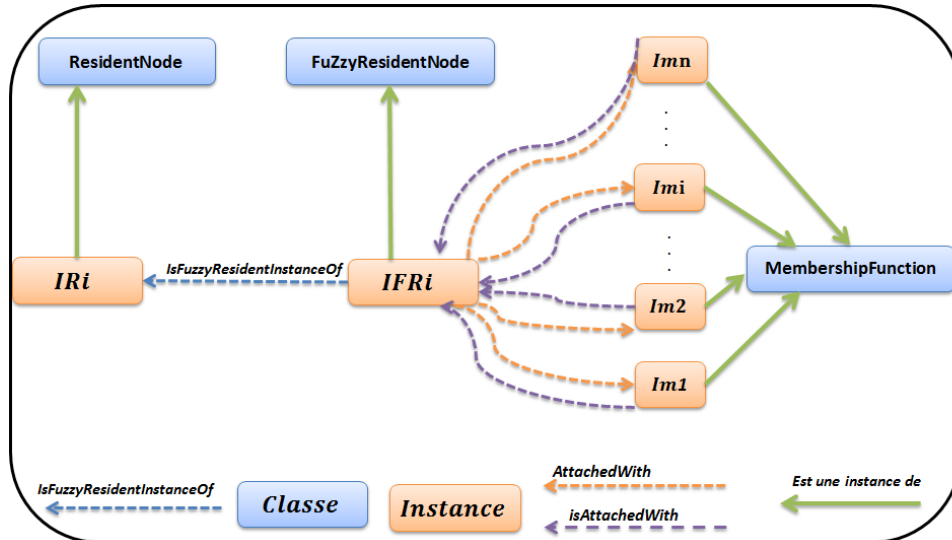


FIGURE 5.5 – Illustration graphique des liens entre les nœuds résidents flous et les nœuds résidents.

La figure 5.6 présente le code en OWL de la classe *FuzzyResidentNode*.

```
<owl:Class rdf:about="http://www.pr-owl.org/pr-owl2.owl#FuzzyResidentNode">
  <rdfs:subClassOf rdf:resource="http://www.pr-owl.org/pr-owl2.owl#Node"/>
</owl:Class>
```

FIGURE 5.6 – Le code OWL de la classe *FuzzyResidentNode*.

Le code en OWL de la propriété d'objets «*IsFuzzyResidentInstanceOf*» est montré dans la figure 5.7.

```
<owl:ObjectProperty rdf:about="http://www.pr-owl.org/pr-owl2.owl#isFuzzyInputInstanceOf">
  <rdfs:domain rdf:resource="http://www.pr-owl.org/pr-owl2.owl#FuzzyInputNode"/>
  <rdfs:range rdf:resource="http://www.pr-owl.org/pr-owl2.owl#FuzzyResidentNode"/>
</owl:ObjectProperty>
```

FIGURE 5.7 – Le code en OWL de la propriété d'objets «*IsFuzzyResidentInstanceOf*».

- La **classe FuzzyInputNode** permet de représenter les nœuds d'entrées flous qui sont des nœuds résidents flous définis dans un autre FuZzy-MFrag. En effet, si un nœud d'entrée $I_k \in I$ est flou, alors une instance $FI_k \in FI$ de la classe *FuzzyInputNode* est créée pour le représenter sémantiquement. En outre, l'instance FI_k est liée avec l'instance I_k en utilisant la propriété d'objet «*isFuzzyInputInstanceOf*».

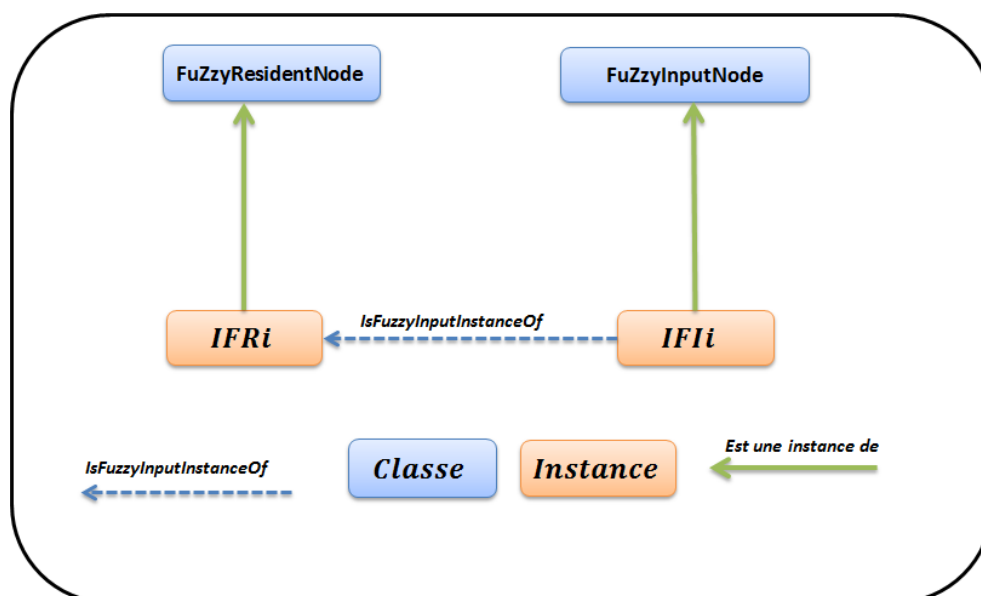


FIGURE 5.8 – Illustration graphique des liens entre les noeuds résidents flous et les noeuds inputs flous.

Le code en OWL de la classe *FuzzyInputNode* est présenté dans la figure 5.9.

```
<owl:Class rdf:about="http://www.pr-owl.org/pr-owl2.owl#FuzzyInputNode">
  <rdfs:subClassOf rdf:resource="http://www.pr-owl.org/pr-owl2.owl#Node"/>
</owl:Class>
```

FIGURE 5.9 – Le code en OWL de la classe *FuzzyInputNode*.

Le code en OWL de la propriété d’objets «*IsFuzzyInputInstanceOf*» est montré dans la figure 5.10.

```
<owl:ObjectProperty rdf:about="http://www.pr-owl.org/pr-owl2.owl#IsFuzzyInputInstanceOf">
  <rdfs:domain rdf:resource="http://www.pr-owl.org/pr-owl2.owl#FuzzyInputNode"/>
  <rdfs:range rdf:resource="http://www.pr-owl.org/pr-owl2.owl#FuzzyResidentNode"/>
</owl:ObjectProperty>
```

FIGURE 5.10 – Le code en OWL de la propriété d’objets «*IsFuzzyInputInstanceOf*».

- La classe **MembershipFunction** permet de représenter les fonctions d’appartenance pouvant être utilisées dans FuZzy-PR-OWL 2. Leur rôle est de fuzzifier les noeuds résidents flous. Ainsi, cette classe contient cinq sous-classes (*Left_Shoulder*, *Right_Shoulder*, *Triangular*, *Linear* et *Trapezoidal*), chacune de ces sous-classes a un ensemble d’arguments. En effet, pour représenter les arguments des fonctions d’appartenance. Nous avons ajouté des propriétés de données (*A*, *B*, *C*, *D* et *E*) à chaque classe et / ou sous-classe comme illustré dans la figure 5.11.

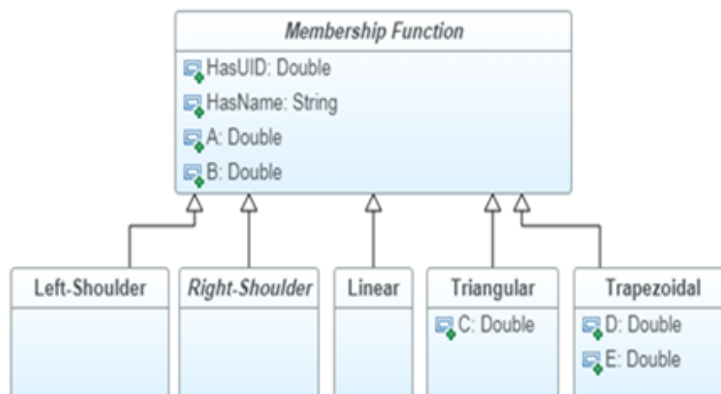


FIGURE 5.11 – Les fonctions d'appartenance utilisées dans FuZzy-PR-OWL2.

Le code en OWL qui formalise la classe *Membershipfunction* est illustré dans la figure 5.12. Cette classe possède deux propriétés des données nommées respectivement *A* et *B*.

```

<owl:Class rdf:about="http://www.pr-owl.org/pr-owl2.owl#MembershipFunction" />
<DataPropertyDomain>
  <DataProperty IRI="#A"/>
  <Class IRI="http://www.pr-owl.org/pr-owl2.owl#MembershipFunction" />
</DataPropertyDomain>
<DataPropertyRange>
  <DataProperty IRI="#A"/>
  <Datatype abbreviatedIRI="xsd:double" />
</DataPropertyRange>
<DataPropertyDomain>
  <DataProperty IRI="#B"/>
  <Class IRI="http://www.pr-owl.org/pr-owl2.owl#MembershipFunction" />
</DataPropertyDomain>
<DataPropertyRange>
  <DataProperty IRI="#B"/>
  <Datatype abbreviatedIRI="xsd:double" />
</DataPropertyRange>
  
```

FIGURE 5.12 – Le code en OWL qui décrit la classe *Membershipfunction* .

Les fonctions d'appartenance de type *Left_Shoulde*, *Right_Shoulder* et *Linear* sont définies comme des sous classes de la classe *Membershipfunction* comme le montre la figure 5.13 .

```

<owl:Class rdf:about="http://www.pr-owl.org/pr-owl2.owl#Linear">
  <rdfs:subClassOf rdf:resource="http://www.pr-owl.org/pr-owl2.owl#MembershipFunction"/>
</owl:Class>

<owl:Class rdf:about="http://www.pr-owl.org/pr-owl2.owl#Right-shoulder">
  <rdfs:subClassOf rdf:resource="http://www.pr-owl.org/pr-owl2.owl#MembershipFunction"/>
</owl:Class>

<owl:Class rdf:about="http://www.pr-owl.org/pr-owl2.owl#Left-shoulder">
  <rdfs:subClassOf rdf:resource="http://www.pr-owl.org/pr-owl2.owl#MembershipFunction"/>
</owl:Class>

```

FIGURE 5.13 – Le code en OWL qui décrit les classes *Left_Shoulder*, *Right_Shoulder* et *Linear*.

La fonction d'appartenance de type triangulaire est définie comme une sous-classe de la classe *Membershipfunction*. De plus, elle possède une autre propriété des données pour représenter le troisième paramètre (voir la figure 5.14).

```

<owl:Class rdf:about="http://www.pr-owl.org/pr-owl2.owl#Triangular">
  <rdfs:subClassOf rdf:resource="http://www.pr-owl.org/pr-owl2.owl#MembershipFunction"/>
</owl:Class>

<DataPropertyDomain>
  <DataProperty IRI="#C"/>
  <Class IRI="http://www.pr-owl.org/pr-owl2.owl#Triangular"/>
</DataPropertyDomain>

<DataPropertyRange>
  <DataProperty IRI="#C"/>
  <Datatype abbreviatedIRI="xsd:double"/>
</DataPropertyRange>

```

FIGURE 5.14 – Le code en OWL qui décrit la classe *Triangular*.

La fonction d'appartenance de type *Trapèze* est définie comme une sous-classe nommée *Trapezoidal* de la classe *Membershipfunction* comme le montre la figure 5.15. En effet, cette classe possède deux propriétés de données supplémentaires pour représenter le troisième paramètre **D** et le quatrième paramètre **E**.

```

<owl:Class rdf:about="http://www.pr-owl.org/pr-owl2.owl#Trapezoidal">
  <rdfs:subClassOf rdf:resource="http://www.pr-owl.org/pr-owl2.owl#MembershipFunction"/>
</owl:Class>

<DataPropertyDomain>
  <DataProperty IRI="#D"/>
  <Class IRI="http://www.pr-owl.org/pr-owl2.owl#Trapezoidal"/>
</DataPropertyDomain>

<DataPropertyRange>
  <DataProperty IRI="#D"/>
  <Datatype abbreviatedIRI="xsd:double"/>
</DataPropertyRange>

<DataPropertyDomain>
  <DataProperty IRI="#E"/>
  <Class IRI="http://www.pr-owl.org/pr-owl2.owl#Trapezoidal"/>
</DataPropertyDomain>

<DataPropertyRange>
  <DataProperty IRI="#E"/>
  <Datatype abbreviatedIRI="xsd:double"/>
</DataPropertyRange>

```

FIGURE 5.15 – Le code en OWL qui décrit la classe *Trapezoidal*.

Les liens entre les noeuds résidents flous et les fonctions d'appartenance sont établis à l'aide de la propriété d'objet *isAttachedWith*(voir la figure 5.16).

```

<owl:ObjectProperty rdf:about="http://www.pr-owl.org/pr-owl2.owl#isAttachedWith">
  <rdfs:domain rdf:resource="http://www.pr-owl.org/pr-owl2.owl#FuzzyResidentNode">
  <rdfs:range rdf:resource="http://www.pr-owl.org/pr-owl2.owl#MembershipFunction">
</owl:ObjectProperty>

```

FIGURE 5.16 – Le code OWL de la propriété d'objet *isAttachedWith*.

5.4.2 Raisonnement dans l'ontologie probabiliste floue

Le fait que FuZzy-PR-OWL2 est une extension floue du langage PR-OWL 2, l'ensemble des tâches du raisonnement pouvant être appliquées sur les ontologies codées en PR-OWL 2 sont toujours valables pour FuZzy-PR-OWL2. Outre que les tâches de raisonnement appliquées sur les ontologies classiques telles que la classification d'individus, la vérification de la subsumption, la satisfiabilité des concepts, etc. PR-OWL2 permet d'effectuer plusieurs tâches de raisonnement incertaines en se basant sur l'inférence bayésienne fournie par les MEBNs. Selon [121], trois tâches de raisonnement principales doivent être implémentées lorsqu'on traite les connaissances probabilistes dans les ontologies.

1. **L'incertitude existentielle** : il faut modéliser l'incertitude autour de l'existence réelle d'un individu dans l'ontologie. Par ailleurs, il s'agit de vérifier si un individu correspond à une description qui existe réellement.

Selon [122], le raisonnement sur l'existence d'un individu revient à vérifier la véracité de la propriété *Existe (individu)* pour l'individu requête. Si sa valeur est vraie alors l'individu existe réellement sinon il n'existe pas. Cette tâche peut être formulée comme suit :

Input : Une ontologie probabiliste floue FPO, un individu I

Tache : Vérifier si la propriété **Exist** (I) est satisfaite ou non dans l'ontologie probabiliste floue FPO

Output : Une valeur booléenne.

2. **L'incertitude relative aux types d'un individu** : cette tâche de raisonnement permet de répondre à la question suivante : quelle est la probabilité pour qu'un individu appartienne à une classe bien déterminée ?

Cette connaissance probabiliste peut être modélisée via des relations d'appartenance non déterministes entre un individu probabiliste et une classe probabiliste. En effet, la relation **isA(ressource, classe)** est définie comme un type particulier de variable aléatoire booléenne. Cette variable est munie d'une distribution probabiliste pour définir l'incertitude relative à l'appartenance de l'individu (*ressource*) à une classe. Cette tâche peut être formulée comme suit :

Input : Une ontologie probabiliste floue FPO, un concept C , un individu I .

Tache : Calculer la probabilité que l'individu I appartienne à la classe C dans FPO. Formellement, calculer $P(\text{isA}(I, C) = \text{true})$

Output : Un couple $\langle \text{isA}(I, C), p \rangle$, avec $p \in [0, 1]$.

A titre d'exemple : supposons qu'on veuille modéliser l'appartenance de l'entité p_1 à la classe *Diabétique* avec une probabilité de 0.7. Cette appartenance est représentée par la relation suivante : $\text{isA}(p_1, \text{Diabétique}) = 0,7$.

3. **L'incertitude liée aux valeurs des propriétés** : quelle est la probabilité qu'un individu possède une propriété avec une valeur donnée ?

Cette connaissance a pour objectif de capturer l'incertitude liée aux propriétés à savoir les propriétés d'objets et les propriétés des données.

- **L'incertitude liée aux propriétés d'objets**, cette tâche vise à calculer la probabilité que deux individus I_1 et I_2 soient liés via une propriété d'objets R avec une valeur donnée V .

Input : Une ontologie probabiliste floue FPO, Une propriété d'objets R , Deux individus I_1 and I_2 .

Tache : Calculer la probabilité que deux individus I_1 et I_2 soient liés avec la propriété d'objets R . Formellement, calculer $P(R(I_1, I_2) = V)$

Output : Un couple $\langle R(I_1, I_2) = V, p \rangle$, avec $p \in [0, 1]$.

Par exemple : "Aymen" (instance de type Personne) est intéressé par le livre "Web sémantique" (instance de type livre) avec une probabilité de 0.8.

- **L'incertitude liée aux propriétés de données**, cette tâche vise à calculer la probabilité qu'un individu I ait une propriété de données D avec une valeur donnée V

Input : Une ontologie probabiliste floue FPO, Une propriété de données D , Un individu I .

Tache : Calculer la probabilité que l'individu I ait une propriété de données D avec une valeur donnée V dans FPO. Formellement, calculer $P(D(I, V) = \text{true})$

Output : Un couple $\langle D(I, V), p \rangle$, avec $p \in [0, 1]$.

Par exemple : la tension artérielle de "Aymen" est de 83 mm Hg (millimètre de mercure) avec une probabilité de 0.75.

Par ailleurs, FuZzy PR-OWL 2 permet d'effectuer toutes ces tâches de raisonnement et tirer des conclusions en se basant sur la puissance de FzMEBN. En effet, FzMEBN permet de calculer des probabilités a posteriori en prenant en compte le flou lié aux observations accordées aux entités du modèle.

5.4.3 Exemple Illustratif

Nous présentons dans cette section un exemple qui illustre le principe de notre approche. Prenons un simple exemple où nous avons une ontologie qui contient la classe *Patient* qui sert à représenter l'ensemble des patients. Elle comporte deux propriétés des données qui sont définies comme suit :

- **hasHémoglobine** : Patient \implies Double.
- **hasAnemia** : Patient \implies Boolean.

Notre objectif est de modéliser cette situation en tenant en compte de l'incertitude liée à la présence de l'anémie et le taux d'hémoglobine en sang d'un patient. Nous avons modélisé cette situation d'incertitude par un FuZzyMFrag nommé FzDMFrag-Anemia qui comporte :

- Un nœud résident flou nommé $hasHmoglobine(p)$.
- Un nœud résident booléen nommé $hasAnemia(p)$.
- Une variable ordinaire p de type patient.

La figure 5.17 montre les éléments de ce FuZzyMFrag.

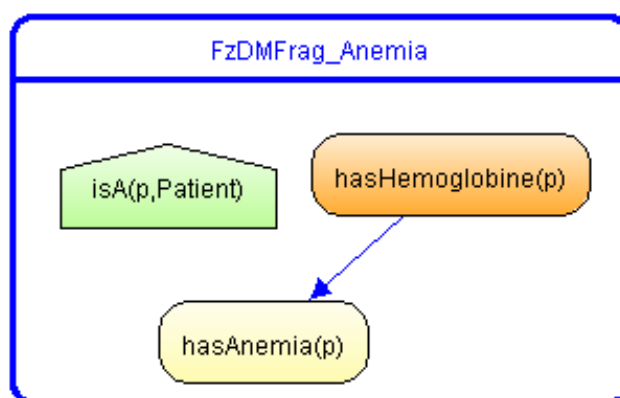


FIGURE 5.17 – La représentation graphique de FzDMFrag-Anemia

Dans FzDMFrag-Anemia, le noeud *hasHemoglobine* est flou, le taux d'hémoglobine dans le sang "médium".

Dans un premier temps, nous décrivons le noeud *hasHemoglobine* sémantiquement par une instance de la classe *ResidentNode*, nommée *Domain_Res.hasHemoglobine* (voir la figure 5.18).

```
<ClassAssertion>
  <Class IRI="http://www.pr-owl.org/pr-owl2.owl#DomainResidentNode"/>
  <NamedIndividual IRI="#Domain_Res.hasHemoglobine"/>
</ClassAssertion>
```

FIGURE 5.18 – Le code en OWL qui représente le noeud *hasHemoglobine*.

En effet, le noeud *hasHemoglobine* possède deux états qui sont *lowHb* et *highHb*. Ces états sont décrits via une propriété de données *hasPossibleValue* comme le montre la figure 5.19.

```
<DataPropertyAssertion>
  <DataProperty IRI="http://www.pr-owl.org/pr-owl2.owl#hasPossibleValues"/>
  <NamedIndividual IRI="#RV_hasHemoglobine"/>
  <Literal datatypeIRI="&xsd:anyURI">file:/I:/anemia.owl#lowHb</Literal>
</DataPropertyAssertion>

<DataPropertyAssertion>
  <DataProperty IRI="http://www.pr-owl.org/pr-owl2.owl#hasPossibleValues"/>
  <NamedIndividual IRI="#RV_hasHemoglobine"/>
  <Literal datatypeIRI="&xsd:anyURI">file:/I:/anemia.owl#highHb</Literal>
</DataPropertyAssertion>
```

FIGURE 5.19 – Les états du noeud *hasHemoglobine*

En outre, le noeud possède une *hasHemoglobine_Table*, via la propriété d'objets *hasProbabilityDistribution*. En effet, cette instance représente sémantiquement la LPD du noeud *hasHemoglobine* (voir figure 5.20).

```
<ClassAssertion>
  <Class IRI="http://www.pr-owl.org/pr-owl2.owl#DeclarativeDistribution"/>
  <NamedIndividual IRI="http://www.pr-owl.org/pr-owl2.owl#hasHemoglobine_Table"/>
</ClassAssertion>

<DataPropertyAssertion>
  <DataProperty IRI="http://www.pr-owl.org/pr-owl2.owl#hasDeclaration"/>
  <NamedIndividual IRI="http://www.pr-owl.org/pr-owl2.owl#hasHemoglobine_Table"/>
  <Literal datatypeIRI="&xsd:string">
    lowHb = 0.42,
    highHb = 0.58
  </Literal>
</DataPropertyAssertion>
```

FIGURE 5.20 – La LPD du noeud *Domain_Res.hasHémoglobine*.

Après avoir représenté d'une manière sémantique la partie probabiliste du noeud *hasHemoglobine*, en utilisant la classe *ResidentNode*, nous allons maintenant passer au processus de la fuzzification afin de modéliser la connaissance floue liée au noeud *hasHemoglobine*.

Il s'agit dans un premier temps de créer une instance de la classe *FuzzyResidentNode*, nommée *Fz.hasHemoglobine*. Cette instance est illustrée dans la figure 5.21, elle est liée au noeud *hasHemoglobine* (qui est représenté par l'instance *Domain.Res.hasHemoglobine*) via la propriété d'objets *IsFuzzyResidentInstanceOf* comme le montre la figure 5.22. Cette opération vise à annoter l'instance *Domain.Res.hasHemoglobine* et dire qu'elle est floue.

```
<ClassAssertion>
  <Class IRI="http://www.pr-owl.org/pr-owl2.owl#FuzzyResidentNode"/>
  <NamedIndividual IRI="#Fz.hasHemoglobine"/>
</ClassAssertion>
```

FIGURE 5.21 – Le code en OWL de l'instance *Fz.hasHemoglobine*

```
<ObjectPropertyAssertion>
  <ObjectProperty IRI="http://www.pr-owl.org/pr-owl2.owl#IsFuzzyResidentInstanceOf"/>
  <NamedIndividual IRI="#Fz.hasHemoglobine"/>
  <NamedIndividual IRI="#Domain Res.hasHemoglobine"/>
</ObjectPropertyAssertion>
```

FIGURE 5.22 – Le code en OWL qui décrit le lien entre l'instance *Fz.hasHemoglobine* et l'instance *Domain.Res.hasHemoglobine*

Par la suite, il faut spécifier les fonctions d'appartenance convenables pour l'opération de la fuzzification et les représenter sémantiquement dans l'ontologie. Pour ce faire, nous avons créé deux instances nommées *Low* et *High* de la classe *Trapizodial* comme il est illustré dans la figure 5.23.

```
<ClassAssertion>
  <Class IRI="http://www.pr-owl.org/pr-owl2.owl#Trapizodial"/>
  <NamedIndividual IRI="#High"/>
</ClassAssertion>

<ClassAssertion>
  <Class IRI="http://www.pr-owl.org/pr-owl2.owl#Trapizodial"/>
  <NamedIndividual IRI="#Low"/>
</ClassAssertion>
```

FIGURE 5.23 – Les fonctions d'appartenance *Low* et *High*.

La fonction d'appartenance *Low* possède quatre propriétés de données *A*, *B*, *D* et *E*, permettant de décrire sémantiquement ses arguments. Leur définition est illustrée dans la figure 5.24

```

<DataPropertyAssertion>
  <DataProperty IRI="#A"/>
  <NamedIndividual IRI="#High"/>
  <Literal datatypeIRI="&xsd;double">10</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#B"/>
  <NamedIndividual IRI="#High"/>
  <Literal datatypeIRI="&xsd;double">16</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#D"/>
  <NamedIndividual IRI="#High"/>
  <Literal datatypeIRI="&xsd;double">20</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#E"/>
  <NamedIndividual IRI="#High"/>
  <Literal datatypeIRI="&xsd;double">20</Literal>
</DataPropertyAssertion>

```

FIGURE 5.24 – Les propriétés des données de la fonction d'appartenance *Low*.

La fonction d'appartenance *High* contient quatre propriétés de données *A*, *B*, *D* et *E* qui représentent ses arguments. Leur définition est montrée dans la figure 5.25

```

<DataPropertyAssertion>
  <DataProperty IRI="#A"/>
  <NamedIndividual IRI="#High"/>
  <Literal datatypeIRI="&xsd;double">10</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#B"/>
  <NamedIndividual IRI="#High"/>
  <Literal datatypeIRI="&xsd;double">16</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#D"/>
  <NamedIndividual IRI="#High"/>
  <Literal datatypeIRI="&xsd;double">20</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#E"/>
  <NamedIndividual IRI="#High"/>
  <Literal datatypeIRI="&xsd;double">20</Literal>
</DataPropertyAssertion>

```

FIGURE 5.25 – Les propriétés des données de la fonction d'appartenance *High*.

Afin de mettre en lien l'instance *Fz.hasHemoglobine* et ses fonctions d'appartenance, nous avons utilisé la propriété des données *isAttachedWith*, voir la figure 5.26.

```

<ObjectPropertyAssertion>
  <ObjectProperty IRI="http://www.pr-owl.org/pr-owl2.owl#isAttachedWith"/>
  <NamedIndividual IRI="#Fz.hasHemoglobine"/>
  <NamedIndividual IRI="#High"/>
</ObjectPropertyAssertion>

<ObjectPropertyAssertion>
  <ObjectProperty IRI="http://www.pr-owl.org/pr-owl2.owl#isAttachedWith"/>
  <NamedIndividual IRI="#Fz.hasHemoglobine"/>
  <NamedIndividual IRI="#Low"/>
</ObjectPropertyAssertion>

```

FIGURE 5.26 – Le code en OWL qui décrit la relation entre l’instance Fz.hasHemoglobine et les fonctions d’appartenances *Low* et *High*.

Supposons qu’on a un patient *Patient_1* qui a une valeur d’hémoglobine de 14 gms. Cette valeur est représentée grâce à la propriété des données hasHemoglobine comme le montre la figure 5.27.

FIGURE 5.27 – Observation sur l’entité *Patient_1*.

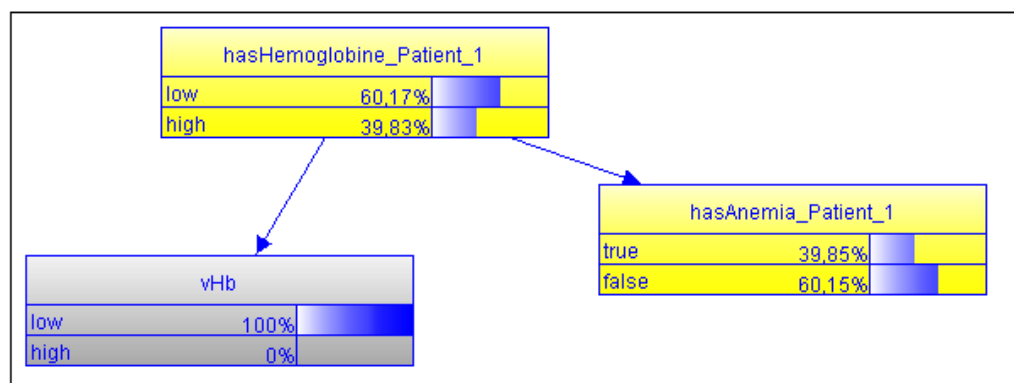
Cette observation est représentée par un vecteur \mathbf{fe} contenant les degrés d’appartenance de l’entité à chaque état du noeud *hasHemoglobine*.

$$\text{— } \mathbf{fe} = \langle \mu_{Low}(14), \mu_{High}(14) \rangle.$$

$$\text{— } \mathbf{fe} = \langle 0.67, 0.33 \rangle.$$

Avec FuZzy-PR-OWL2 on peut déduire de nouvelles connaissances sur le patient nommé *Patient_1* en utilisant l’inférence probabiliste basée sur des observations floues de FzMEBN.

En effet, sachant que la valeur d’hémoglobine est de 14 gms du *patient_1*, nous pouvons constater que ce patient peut développer l’anémie avec une probabilité de 60 %, comme le montre la figure 5.28. Cette déduction est établie en générant un SSBN flou et en appliquant une inférence probabiliste floue.

FIGURE 5.28 – Le SSBN flou pour le *Patient_1*.

5.5 Comparaison avec d'autres approches

Comme nous l'avons précisé dans le chapitre 3, plusieurs solutions ont été développées pour traiter la connaissance incertaine dans les ontologies. Ces solutions s'accordent sur certaines propriétés. Cependant, leur étude nous a permis de répertorier un ensemble de limites :

1. La plupart des approches existantes (Ding et al., 2004; Ding et al., 2006; Yang et al., 2005; Fenz, 2012; Mouenis et al., 2014; Emna et al., 2016a; Emna et al., 2016b; Emna et al., 2017) se concentrent sur l'intégration des réseaux bayésiens classiques dans les ontologies classiques. Cependant, ces derniers comportent de nombreuses limites, telle que le fait elles ne sont pas assez expressives et elles ne traitent pas les connaissances vagues attachées aux différents noeuds. Notre solution prend en compte ces problèmes et utilise un modèle hybride combinant logique floue et MEBN. En effet, notre FzMEBN est flexible, expressif et peut traiter des événements probabilistes en tenant compte des connaissances floues.
2. Bien que l'extension de (Mohammed et al., 2016) se base sur les réseaux bayésiens hybrides pour traiter à la fois les concepts continus et discrets dans les ontologies classiques, permettant de traiter la connaissance probabiliste dans les ontologies, elle ne peut pas traiter les connaissances floues dans les ontologies.
3. La solution présentée dans (Zheng et al., 2016) est la seule extension qui intègre la logique floue dans les ontologies probabilistes. Néanmoins, ce travail a des limites, notamment :
 - **Dans l'étape de modélisation**, leur extension se base sur des MFrag classiques attachés à un ensemble de règles floues. En revanche, notre extension se base sur FuZzy MFrag, qui contient des noeuds flous et des noeuds précis et peut être instanciée plusieurs fois afin de générer des réseaux Bayésiens flous permettant de réaliser une inférence probabiliste floue.

- **Dans l'étape de raisonnement**, l'inférence probabiliste floue n'est pas considérée dans leur extension. En revanche, l'inférence probabiliste floue de FzMEBN est utilisée dans notre solution qui se repose sur la méthode de l'évidence virtuelle pour incorporer les observations floues lors de l'inférence probabiliste. Notre solution permet de faire face à des événements probabilistes liés à des connaissances floues.

La plupart des solutions proposées pour traiter l'incertitude dans les ontologies se reposent sur l'intégration des RBs classiques dans les ontologies. Néanmoins, il est possible d'améliorer encore la représentation et le raisonnement avec l'incertitude en combinant la logique floue et les RBs pour faire face à l'incertitude dans les ontologies du domaine riches en termes d'incertitude. Dans cette perspective, notre solution utilise à la fois la logique floue et les RBs basés sur la sémantique FzMEBN afin de traiter à la fois les connaissances probabilistes et les connaissances floues. Le mérite de notre solution réside dans le fait que :

- Elle traite des événements probabilistes même avec des observations incomplètes,
- Elle modélise les événements dont les résultats ne peuvent pas être prédits avec certitude (caractère aléatoire du phénomène).
- Elle fournit une inférence probabiliste floue en se basant sur des observations floues.

Ainsi, le tableau 5.1 résume les différences entre les approches proposées dans la littérature pour traiter l'incertitude dans les ontologies et notre solution. Les critères de la comparaison sont :

- L'entrée de chaque approche,
- L'inférence fournie par chaque approche en tenant compte de l'incertitude,
- Les modèles mathématiques utilisés pour faire face à l'incertitude,
- L'incertitude traitée par chaque approche.

Approche	Input	Modèles utilisés	Connaissance modélisée	Inférence fournie en présence de l'incertitude
-P-SHOQ (Giugno et al., 2002) -Creedal ALC (Polastro et al., 2012) -BORN Ceylan et al., 2015; -OntoBayes (Yang et al., 2005). -BayesOWL (Ding et al., 2006) -PODM (Emna et al., 2016a, Emna et al., 2016b) -Mouenis et al., 2014 -Fenz, 2012 -Emna et al., 2017	-Logique de Description - Ontologie classique	-Théorie des probabilités.. - Réseaux bayésiens classiques.	- Connaissance probabiliste -Connaissance probabiliste	- Inférence probabiliste -Inférence probabiliste.
HyProb-Ontology (Mohammed et al., 2016)	-Ontologie classique.	-Réseaux bayésiens hybrides.	- Connaissance probabiliste -Connaissance probabiliste -Connaissance floue	- Inférence probabiliste -Inférence probabiliste.
Zheng et al., 2016	-Ontologie probabiliste.	- Logique floue -Réseaux bayésiens multi-entités.		
FuZzy-PR-OWL2	-Ontologie Probabiliste	-Logique floue -Réseaux bayésiens multi-entités flous.	-Connaissance Probabiliste -Connaissance floue	-Inférence probabiliste. -Inférence probabiliste floue

TABLE 5.1 – Comparaison de notre approche avec les approches existantes.

5.6 Conclusion

Nous avons proposé dans ce chapitre une nouvelle approche pour la gestion de la connaissance incertaine dans les ontologies, nommée FzZy-PR-OWL 2. Notre approche est une extension du langage probabiliste PR-OWL 2 pour permettre la modélisation de la connaissance floue du domaine. L'idée principale est d'exploiter les capacités de modélisation et de raisonnement offertes par notre FzMEBN pour traiter la connaissance floue et la connaissance probabiliste simultanément dans les ontologies du domaine.

Notre approche modélise et intègre l'incertitude attachée aux éléments de l'ontologie OWL en se basant sur FzMEBN qui fournit des mécanismes de modélisation très expressifs fondés sur la logique du premier ordre. De plus, nous proposons un processus d'inférence efficace pour déduire de nouvelles connaissances ontologiques à partir de celles représentées dans l'ontologie en appliquant une inférence probabiliste floue.

Dans le chapitre suivant, nous exposons une étude de cas dans le domaine médical pour montrer les capacités et les avantages de notre approche en matière de modélisation et du raisonnement avec la connaissance probabiliste et la connaissance floue. A cet effet, nous construisons une ontologie probabiliste floue pour le diabète Type 2 pour l'héritage de PimaIndian.

Chapitre 6

Tests et validation : étude de cas dans le domaine médical

“L’homme est sage, tant qu’il cherche la sagesse ; Mais dès qu’il croit l’avoir trouvée, il perd la tête.”

— Proverbe arabe

Sommaire

6.1	Introduction	119
6.2	Présentation du domaine d’application	120
6.3	Construction de l’ontologie probabiliste T2DRF	121
6.3.1	Étape 1 : Spécification des besoins	122
6.3.2	Étape 2 : Analyse et la conception	123
6.3.3	Étape 3 : Implémentation	126
6.4	Fuzzification de l’ontologie probabiliste	129
6.5	Tests	130
6.5.1	Vérification	131
6.5.2	Validation	133
6.6	Conclusion	135

6.1 Introduction

Les technologies du Web sémantique notamment les ontologies ont reçu une importance très remarquable durant les dernières années, elles sont capables de faciliter la gestion et le partage des connaissances et de promouvoir l’interopérabilité sémantique entre les systèmes d’information du domaine médical. Cependant, les décisions médicales sont difficiles à prendre car elles doivent être fondées souvent sur des informations incertaines d’une part, et d’autre part, les technologies du Web sémantique classiques ne permettent pas de manipuler des informations

incertaines. Dans notre étude nous allons montrer l'efficacité de notre approche dans le domaine médical afin de traiter les informations médicales incertaines.

L'organisation de ce chapitre est la suivante; nous présentons dans un premier lieu le domaine d'application ainsi que la maladie sur laquelle nous allons mettre l'accent pour montrer les performances de notre approche. Nous donnons également un aperçu sur la base de données utilisée pour le test. Nous enchaînons une deuxième partie (étude qualitative) dans laquelle nous présentons en détails comment exploiter notre solution en tant qu'approche permettant de modéliser et faire des raisonnements sur la connaissance probabiliste et la connaissance floue simultanément. Nous présentons par la suite une étude empirique (étude quantitative) afin d'étudier l'inférence dans notre approche en se basant sur certaines mesures d'évaluation.

6.2 Présentation du domaine d'application

Les connaissances dans le domaine médical relatives à une maladie et / ou un patient sont souvent incertaines et parfois très complexes à traiter. Pour cela les médecins sont obligés de prendre des décisions dans des situations très critiques. Plusieurs facteurs contribuent à l'apparence de l'incertitude dans les connaissances du domaine médical, notamment :

- (a) Le caractère naturel et incertain de ce domaine. Les relations entre la présence d'une maladie chez un patient et ses facteurs de risque sont souvent incertaines (**Aléatoire et Variabilité naturelle**).
- (b) La subjectivité du médecin. Les médecins prennent des décisions sur la base de leur savoir et leur savoir-faire (**Incertitude épistémique**).
- (c) Les informations d'un patient sont inexactes, floues et incomplètes (**Imprecision et Ambiguïté**).

Les médecins prennent des décisions très difficiles en présence de toutes ces imperfections. Il est donc intéressant de concevoir et de mettre en œuvre des systèmes automatisés dans le domaine de la santé qui facilitent la tâche de diagnostic médical. Les médecins sont alors face à des situations très critiques et doivent prendre des décisions qui ne sont pas évidentes. En effet, le but ultime est d'améliorer les pratiques médicales et le développement des systèmes d'aides à la décision efficaces. Ceci dépend fortement de la disponibilité des outils capables de partager et de représenter la sémantique des données médicales collectées en tenant compte de leur nature incertaine.

Le diabète c'est une conséquence d'un défaut d'insuline; une hormone produite par le pancréas. Ce défaut pourrait être l'absence ou la production insuffisante de résistance à l'insuline dans le corps, ce qui est comparable à la non-réponse des cellules humaines à l'insuline (défaut des récepteurs de l'insuline). Trois types de diabète sont distingués, Type 1 nommé insulino-dépendant, type 2 nommé non insulino-dépendant. Le troisième type est vu dans la période de grossesse appelé diabète gestationnel.

Dans notre étude, nous nous intéressons à la modélisation du diabète de type 2 pour Pima indien et ses facteurs de risque (**T2DRF** : Type 2 Diabetes Risk Factors ontology) en utilisant notre FuZzy-PR-OWL 2.

L'ensemble de données utilisé dans notre étude nommé Pima Indien [123], il sauvegarde des informations sur une population de femmes (768 femmes) d'ascendance indienne Pima. Les données ont été collectées par l'institut national américain du diabète, des maladies digestives et rénales (NIDDK : National Institute of Diabetes and Digestive and Kidney Diseases) en testant le diabète de chaque femme selon les critères de l'organisation mondiale de la santé. L'objectif de l'ensemble de données est de prédire le diagnostic en se basant sur l'historique des dossiers médicaux, si un patient est diabétique ou non en fonction de certaines mesures de diagnostic incluses dans l'ensemble de données.

6.3 Construction de l'ontologie probabiliste T2DRF

La construction de l'ontologie T2DRF passe par deux étapes. Dans la première nous construisons une ontologie probabiliste en adaptant la méthodologie UMP-ST. La deuxième étape consiste à faire la fuzzification sémantique de notre ontologie pour la rendre probabiliste floue.

La méthodologie UMP-ST est basée sur un processus itératif et incrémental, inespéré du processus unifié UP pour la conception d'une ontologie probabiliste. Cette méthodologie a été proposée dans un cadre général pour le développement des ontologies probabilistes. Dans notre étude nous allons suivre cette méthodologie afin de construire une ontologie probabiliste formalisée en PR-OWL 2. La méthodologie UMP-ST organise les tâches et les itérations en quatre étapes :

- La spécification des besoins,
- L'analyse et la conception,
- L'implémentation,
- Les tests.

Le cycle de la modélisation ontologique probabiliste ainsi que les différentes étapes qui constituent la méthodologie UMP-ST sont présentées dans la figure 6.1.

La répartition de nos efforts change avec le temps comme il est montré dans la figure 6.2. Les premières itérations ont tendance à mettre l'accent beaucoup plus sur les premières étapes du développement à savoir : **La spécification des besoins, l'analyse et la conception**. Par la suite, nos efforts seront portés sur les étapes **l'implémentation et les tests**.

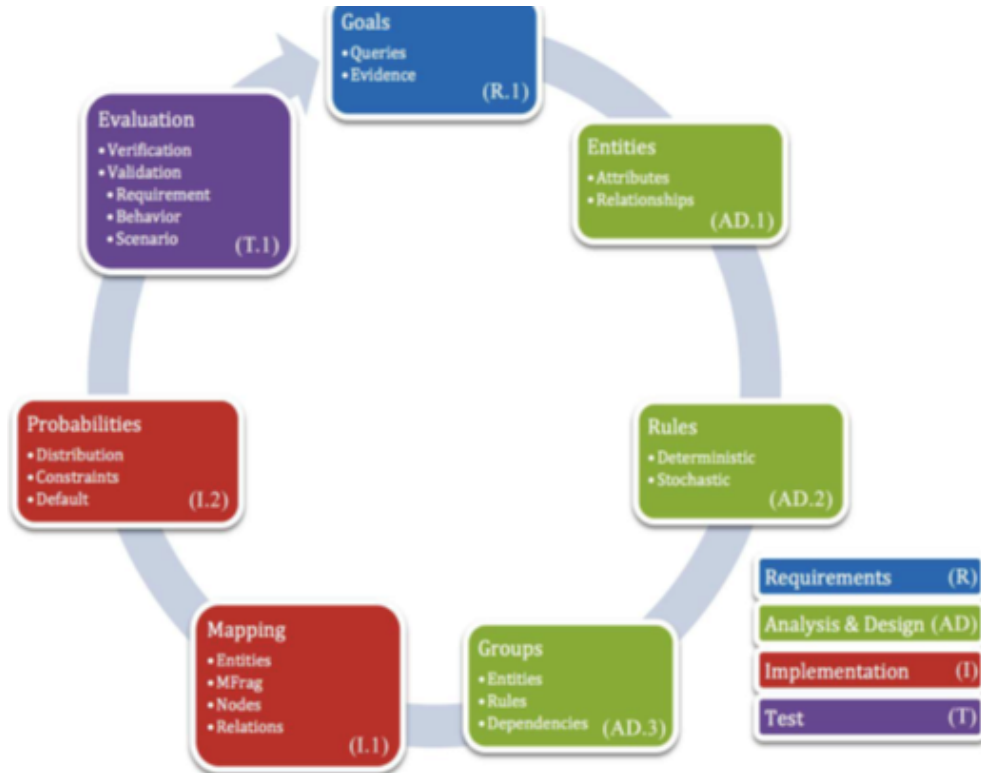


FIGURE 6.1 – Cycle de la modélisation ontologique probabiliste avec la méthodologie UMP-ST [102].

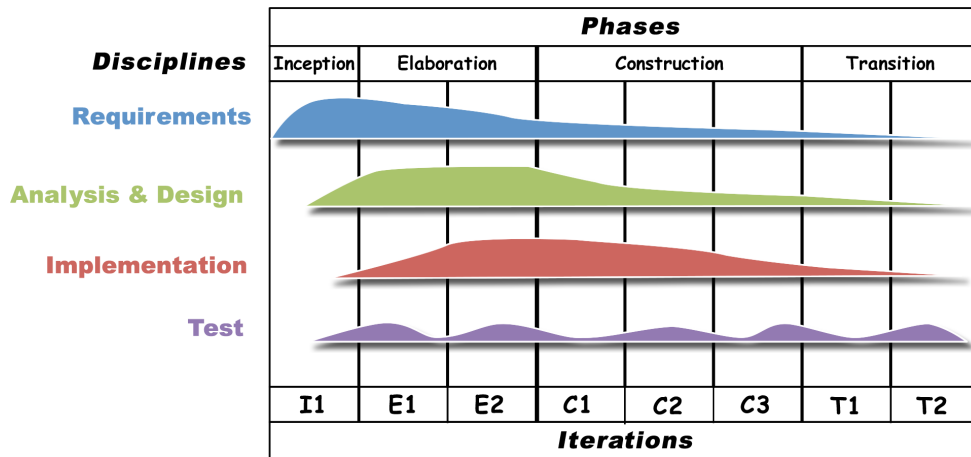


FIGURE 6.2 – Processus de la modélisation d’incertitude via la méthodologie UMP-ST [102].

6.3.1 Étape 1 : Spécification des besoins

La première étape consiste à spécifier les besoins ainsi que la portée de l’ontologie probabiliste. Le but de cette étape est la compréhension et l’expression des besoins et des exigences des utilisateurs. En effet, les besoins sont l’ensemble des requêtes pour lesquels l’utilisateur souhaite avoir des réponses à travers le raisonnement sémantique probabiliste. L’objectif principal de l’ontologie T2DRF est de répondre à la requête suivante :

Identifier si la patiente est touchée par le diabète ou pas ?

L'exécution de cette requête peut être considérée comme l'exécution de plusieurs sous requêtes, à savoir :

Sous requête 1 (Query) : La patiente a-t-elle fait un dépistage de diabète par le biais de tests ?

-Observations :

- Test de tension artérielle,
- Test de glucose,
- Test d'insuline.

Sous requête 2 (Query) : La patiente pratique-t-elle du sport ?

-Observations :

- Type de sport.

Sous requête 3 (Query) : Quelles sont les informations personnelles de la patiente qui sont utiles pour la prédiction ?

-Observations :

- Age de la patiente,
- L'épaisseur de la peau de la patiente (skin thickness), tension artérielle
- L'indice de masse corporelle (BMI : Body Mass Index),
- Le nombre de grossesses (Pregnancies number),
- La fonction de généalogie de diabète (DPF : Diabetes Pedigree Function). L'épaisseur de la peau de la patiente (skin thickness), tension artérielle

6.3.2 Étape 2 : Analyse et la conception

Cette étape a pour but de construire un modèle sémantique de conception, qui décrit la réalisation des exigences élaborées dans l'étape précédente. En effet, au cours de cette étape il faut tirer les classes d'entités, leurs attributs, les relations entre ces classes ainsi que les règles qui sont en relation avec dans le domaine en question et qu'on peut exploiter pour répondre à des requêtes.

6.3.2.1 Entités

On distingue trois entités, à savoir :

- Patiente.
- Dépistage du diabète (Diabetes screening).
- Activité (Activity).

Le tableau 6.1 présente les attributs de l'entité patiente :

Attributs	Domaine	Range
DPF	Patiente	Float
BMI	Patiente	Float
Age	Patiente	Int
skin thickness	Patiente	Float
Pregnancies number	Patiente	Int

TABLE 6.1 – Les attributs de l'entité patiente.

Le tableau 6.2 présente les attributs de l'entité Diabetes_screening :

Attributs	Domaine	Range
Glucose	Diabetes_screening	Float
Blood_Pressure	Diabetes_screening	Float
Insulin	Diabetes_screening	Int

TABLE 6.2 – Les attributs de l'entité Diabetes_screening.

Le tableau 6.3 présente les attributs de l'entité Activity :

Attributs	Domaine	Range
Name	Activity	String
Type	Activity	String

TABLE 6.3 – Les attributs de l'entité Activity.

Le tableau 6.4 présente les relations (Object properties) entre les entités :

Relations	Domaine	Range
has_practice_Activity	patiente	Activity
has_screening	patiente	Diabetes_screening

TABLE 6.4 – Les relations entre les entités.

Le diagramme de classes illustré dans la figure 6.3 présente une conception simplifiée de l'ontologie T2DRF illustrant les principaux attributs utilisés pour répondre à nos requêtes.

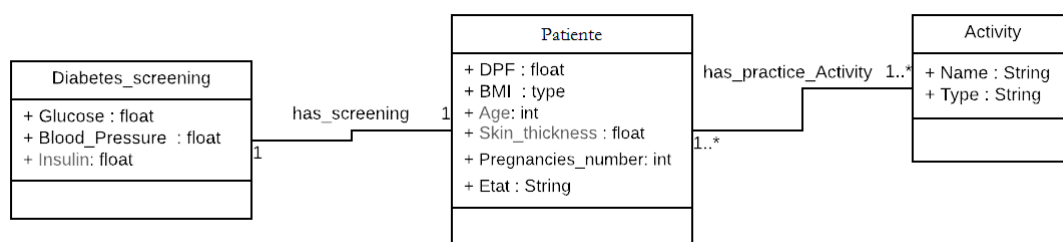


FIGURE 6.3 – Diagramme de classes pour l'ontologie T2DRF.

6.3.2.2 Règles

En plus de la définition des concepts, des propriétés et des règles déterministes, UMP-ST présente des techniques permettant d'identifier et de spécifier des règles non déterministes (probabilistes) pour répondre aux exigences définies dans l'étape 1 et d'introduire les relations de dépendance entre les propriétés de domaine d'application basées sur ces règles. Elle permet de quantifier les poids de ces relations comme des paramètres de distributions des probabilités. Parmi ces règles nous citons :

1. Si la patiente a exercée une activité faible alors la probabilité qu'elle soit physiquement inactive est de [true = 0.65, false = 0.35],
2. Si la patiente a exercée une activité vigoureuse alors la probabilité qu'elle soit physiquement inactive est de [true = 0.20, false = 0.80],
3. Si la patiente a un Glucose=low , Age =young , PhysicalInactivity= false , Insulin=low , BMI=low , Pregnancies=low , BloodPressure =low , DiabetesPedigreePedig=low , et BMI=low alors la probabilité qu'elle soit touchée par la maladie est de [false= 0.33, true= 0.66],
4. Si la patiente a un Glucose=medium, Age =old, PhysicalInactivity= true, Insulin=medium, BMI=medium, Pregnancies=medium ,BloodPressure =high , DiabetesPedigreePedig=medium et BMI=medium alors la probabilité qu'elle soit touchée par la maladie est de [false= 0.25, true= 0.75].

Il est à noter que la liste des règles présentées en haut n'est pas exhaustive. En effet, les probabilités des deux premières règles sont tirées auprès d'un expert médical. Cependant, les valeurs de probabilités pour les autres règles sont tirées en suivant le processus illustré dans la figure 6.4.

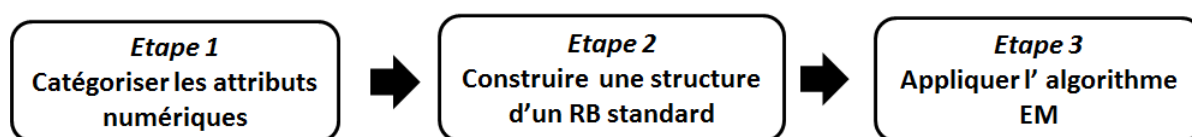


FIGURE 6.4 – Le processus d'extraction des probabilités.

- **Étape 1** : cette étape vise à catégoriser les attributs qui sont numériques pour pouvoir appliquer l’algorithme EM par la suite.
- **Étape 2** : cette étape a pour but de construire une structure d’un réseau bayésien qui contient les dépendances entre les attributs du data set et l’attribut qui représente l’état de la patiente, cette structure est alimentée dans l’étape suivante par des valeurs de probabilités.
- **Étape 3** : au cours de cette étape, l’algorithme EM est appliqué sur la structure élaborée dans l’étape précédente et le dataset (75% de la population a été utilisée pour l’apprentissage) afin d’extraire des valeurs de probabilités conditionnelles que nous avons utilisés pour définir les règles probabilistes combinées. L’avantage d’utiliser EM revient au fait que le dataset de notre étude contient des valeurs manquantes.

6.3.3 Étape 3 : Implémentation

Cette étape utilise le résultat de la conception pour formaliser l’ontologie dans un langage de modélisation probabiliste. Il s’agit de faire un mapping entre le modèle conceptuel élaboré dans l’étape de la conception et un langage spécifique capable de représenter sémantiquement l’incertitude. cette étape implique le codage des classes, des attributs, des relations et des règles dans un langage probabiliste. Dans notre cas nous allons implémenter notre modèle conceptuel dans le langage PR-OWL2 en utilisant le framwork UnBBayes.

UnBBayes [124] est un framework open source écrit en Java, permettant de manipuler les modèles graphiques probabilistes. Il possède à la fois une interface graphique GUI et une API pour fournir plusieurs services à savoir l’inférence, l’échantillonnage et l’apprentissage. Il prend en charge les réseaux bayésiens, les OOBNs, les MEBNs / PR-OWL2, etc.

La première étape d’implémentation de l’ontologie en PR-OWL 2 consiste à représenter les entités, les attributs et les relations conçus dans l’étape 6.3.2.1 sous forme de classes, propriétés de données et propriétés d’objects en OWL. La figure 6.6 présente les entités de l’ontologie T2DRF implémentées dans UnBBayes.

La définition des noeuds requiert la déclaration des variables ordinaires dont on aura besoin pour la définition des noeuds. Une variable ordinaire est définie via la relation IsA(Ressource, Class). Dans notre cas, nous avons deux variables ordinaires qui sont IsA (p, Patiente) IsA (a, Activity) tels que :

- **p** est une variable ordinaire et Patiente représente une entité.
- **a** est une variable ordinaire et Activity est une entité .

La définition des entités est suivie par le traitement des composants qui peuvent être incertains. Chaque attribut incertain d’une entité ou une relation incertaine entre entités sera représenté par une variable aléatoire dans MEBN. Par exemple, la variable aléatoire Glucose (p) correspond à l’attribut hasGlucose de l’entité Diabetes_screening.

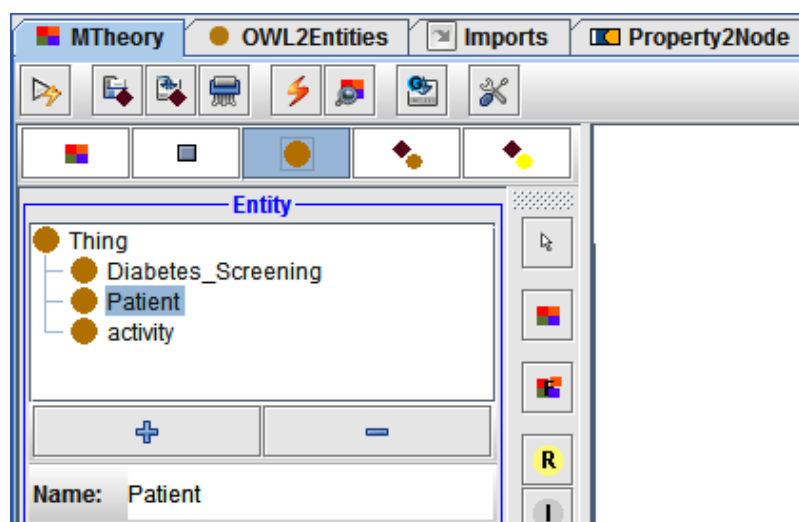


FIGURE 6.5 – Les entités de l'ontologie T2DRF.

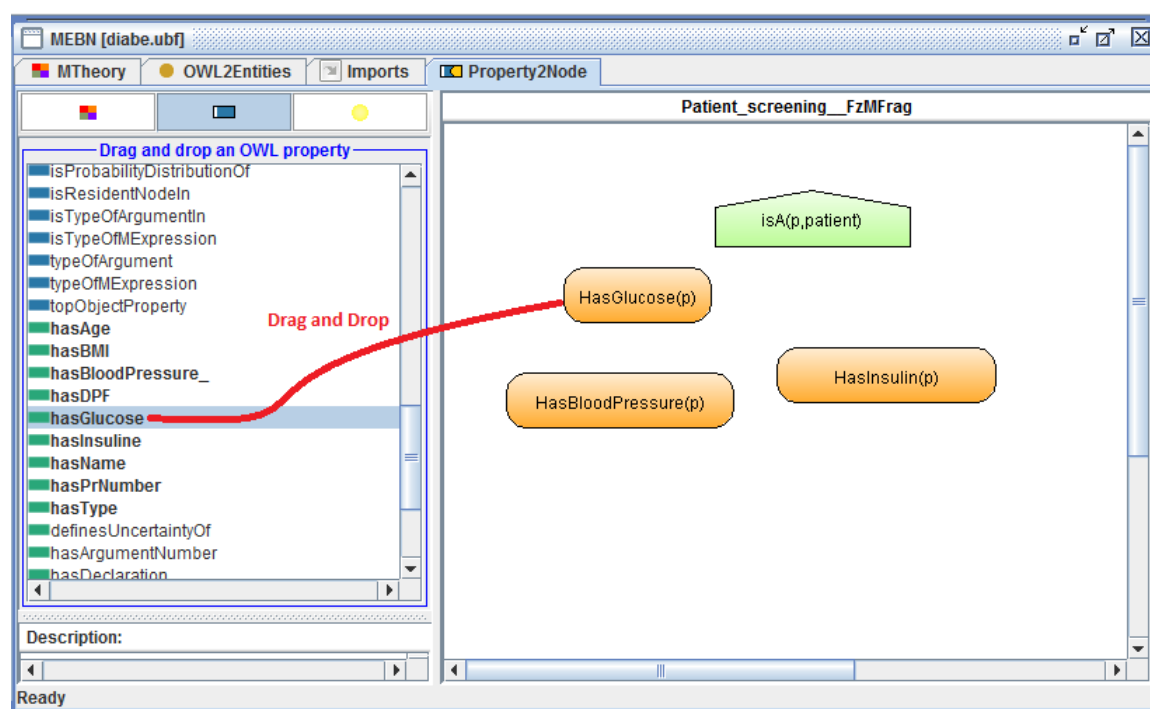


FIGURE 6.6 – Créer une variable aléatoire en PR-OWL 2 à partir d'une propriété OWL.

La MThéorie T2DRF comprend cinq FzMFrags à savoir PatientFzMFrag, ActivityFzMFrag, PhysicalInactivityFzMFrag, PatientScreeningFzMFrag et DiabetesT2FzMFrag.

- ActivityFzMFrag* a un nœud résident HasPractice, ce nœud est exporté dans FzMFrag PhysicalInactivityFzMFrag pour définir une condition,
- PatientFzMFrag* comporte cinq nœuds résidents à savoir HasPregnanciese, BMI, Age, DiabetesPedigreeFunction, HasSkinThickness.

- (c) *PatientScreeningFzMFrag* se compose de trois nœuds résidents qui sont : HasInsulin, HasGlucose, HasBloodPressure.
- (d) *PhysicalInactivityFzMFrag* se compose de deux nœuds résidents, PhysicalInactivity et TypeActivity.
- (e) *DiabetesT2FzMFrag* comprend un nœud résident appelé isDiabetic qui permet de prendre des décisions concernant le diagnostic du diabète. Il contient également un ensemble de nœuds d'entrée qui sont importés à partir d'autres FzMFrag, tels que : HasPregnancies, HasBMI, HasAge, DiabetesPedigree-Function, HasSkinThikness, HasInsulin, HasGlucose, HasBloodPressure et PhysicalInactivity.

La dernière étape de la construction d'une ontologie T2DRF dans UnBBayes consiste à définir les LPD pour tous les nœuds résidents. La LPD de chaque nœud a été définie sur la base de l'ensemble de règles établi dans l'étape 2 (voir section 6.3.2.2). Par exemple, la LPD du noeud hasGlucose est montrée dans la figure 6.9 et la LPD du noeud PhysicalInactivity est illustrée dans la figure 6.8.

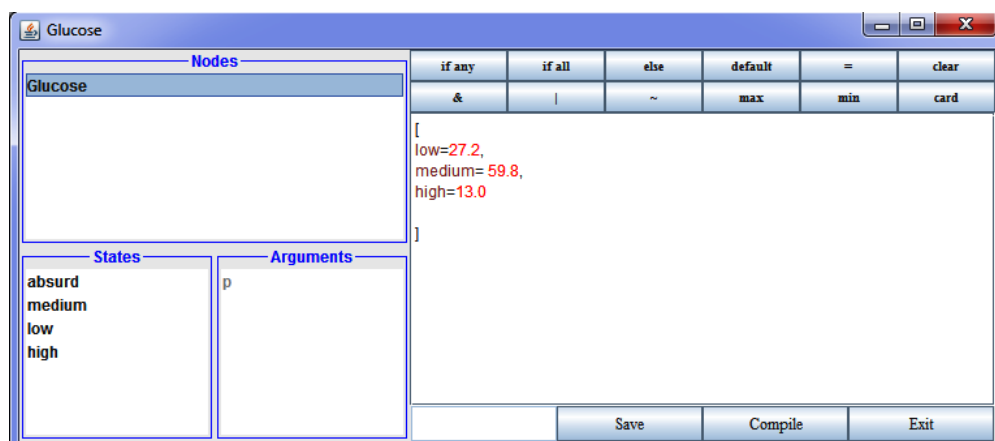


FIGURE 6.7 – LPD du noeud hasGlucose.

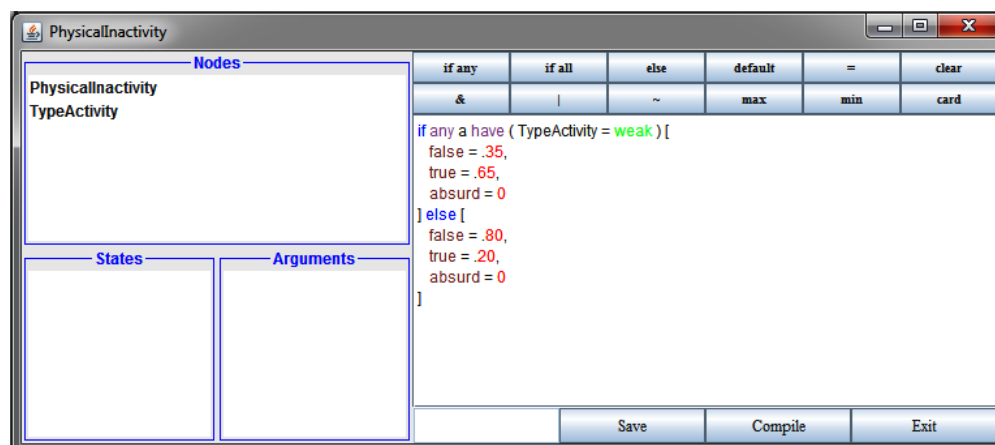


FIGURE 6.8 – LPD du noeud PhysicalInactivity.

6.4 Fuzzification de l'ontologie probabiliste

Cette étape permet de fuzzifier l'ontologie probabiliste afin de capturer la connaissance floue. Le tableau 6.5 montre les nœuds flous avec les fonctions d'appartenance nécessaires pour la fuzzification.

Noeud flou	Etats	Fonction d'appartenance	
		Type	Arguments [a, b, c, d]
Age	young	Trapezoid	21, 21, 27,35
	medium	Trapezoid	27,35, 47,52
	old	Trapezoid	47, 52, 81,81
glucoze	Low	Trapezoid	0, 0, 90,100
	Normal	Trapezoid	90, 100, 157,162
	High	Trapezoid	157,162, 200,200
Tension artérielle	Low	Trapezoid	0.1, 0.1, 78,85
	High	Trapezoid	78, 85, 123,123
L'épaisseur de la peau	Low	Trapezoid	0, 0, 20,26
	Normal	Trapezoid	20, 26, 47,53
	High	Trapezoid	47,53, 100,100
Insulin	Low	Trapezoid	0, 0, 53,58
	Normal	Trapezoid	53, 58, 95,103
	High	Trapezoid	95, 103, 846,846
MBI	Low	Trapezoid	0, 0, 16,20
	Normal	Trapezoid	16, 20, 27,33
	High	Trapezoid	27, 33, 71.5,71.5
La fonction de généalogie de diabète	Low	Trapezoid	0, 0, 0.30,0.35
	Normal	Trapezoid	0.3, 0.35, 1.45,1.50
	High	Trapezoid	1.45, 1.50, 2.5,2.5

TABLE 6.5 – La fuzzification des noeuds flous.

La figure 6.9 montre la MThéorie T2DRF (le graphe complet) qui contient tous les FuZzyM-Frags après la fuzzification.

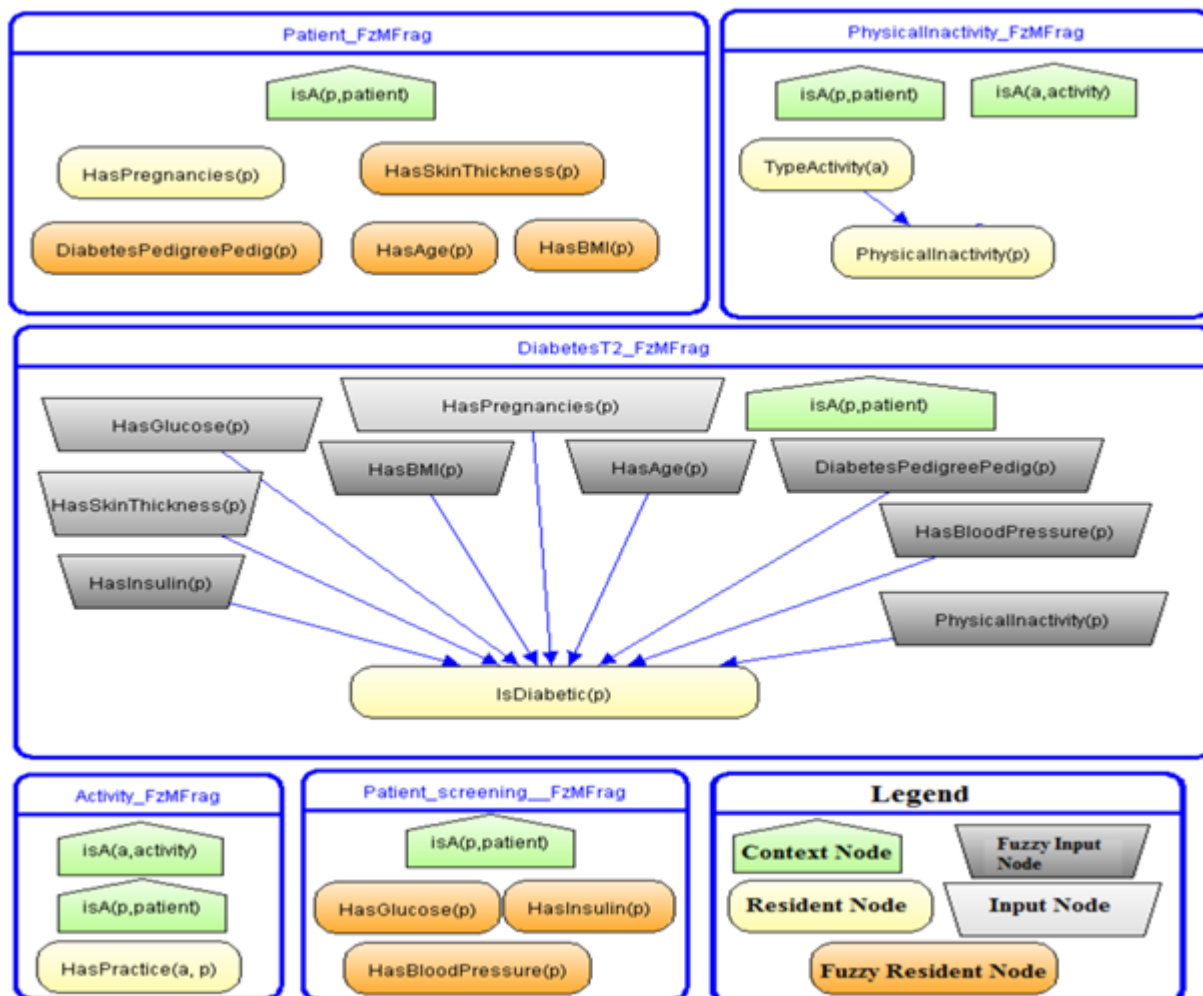


FIGURE 6.9 – La MThéorie T2DRF.

6.5 Tests

Le test est une étape primordiale qui vise à assurer la qualité de l'ontologie développée. Cette qualité comprend plusieurs dimensions qu'on doit aborder et prendre en considération au cours de cette étape. Selon [125], deux dimensions doivent être considérées, la vérification et la validation. En effet, l'objectif est de tester si notre système a été **bien** construit et **correctement**.

- La vérification est le processus qui vise à contrôler la conformité du système aux exigences spécifiées par les utilisateurs. Nous devons déterminer " si notre système a été construit **correctement** ?". Nous allons vérifier par la suite que notre ontologie développée permet de répondre à nos besoins à travers un scénario réel (section 6.5.1).
- La validation consiste à vérifier si un système donné est bien adapté à son objectif, fonctionne comme convenu et qu'il atteint les performances prévues dans son environnement opérationnel. Nous devons vérifier "si notre système a été **bien construit** ?". Nous allons présenter dans la section (6.5.2) une validation empirique en se basant sur des données historiques.

6.5.1 Vérification

Comme scénario de test unitaire, supposons que nous avons une patiente nommée «patiente_1», avec des observations incomplètes et floues comme illustrées dans le tableau 6.6. Nous voulons vérifier comment la requête `isDiabetic(patiente_1)` sera traitée en se basant sur les observations relatives au `patiente_1`.

- **Hypothèse à tester** : `isDiabetic(patiente_1)`,
- **Résultat attendu** : La patiente n'est pas diabétique.

Le SSBN flou généré pour la requête `isDiabetic(patiente_1)` est présenté dans la figure 6.10 .

Patiente_1	
Has_age = 29	Has_blood_pressure =112
Has_insulin=70	Has_Pregnancies= unknown
Has_DPF=0.32	Has_BMI =32
Has_skin_thickness= unknown	Has_practice_activity1 = sport. Activity_type= vigorous
Has_practice_activity2= working. Activity_type= vigorous	
Has_glucose = 113	

TABLE 6.6 – Les informations de la patiente_1

Nous pouvons voir dans le SSBN flou qu'il y a une chance de 45,34% que la `patiente_1` souffre de diabète 2 et qu'il y a une chance de 54,66% que la `patiente_1` ne souffre pas de diabète 2. Par conséquent, nous constatons que nous avons obtenu le résultat attendu par ce scénario.

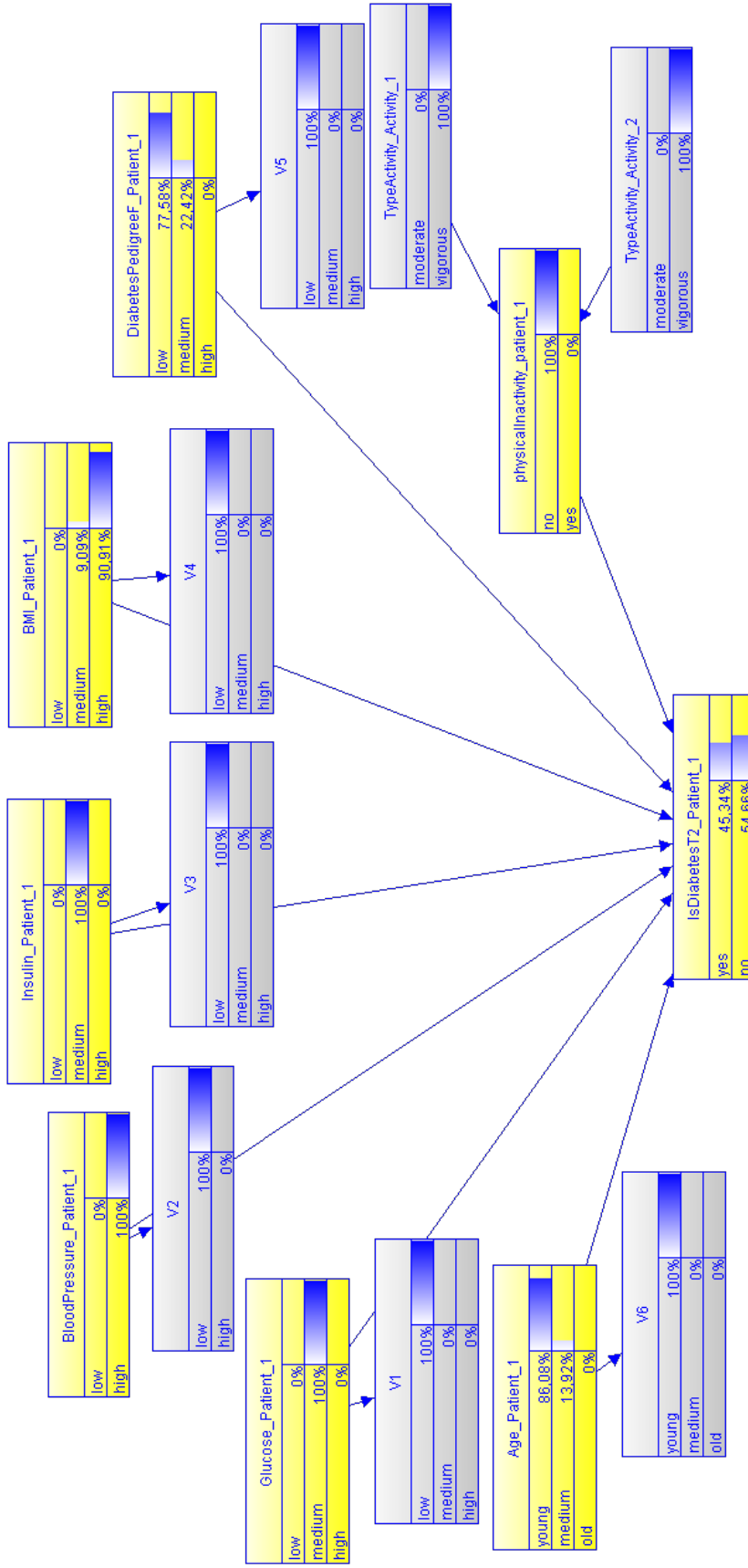


FIGURE 6.10 – Le SSBN flou généré pour la observations de la patiente_1.

Il convient de noter que ce résultat est obtenu en appliquant l'inférence probabiliste floue. Les nœuds virtuels $V1$, $V2$, $V3$, $V4$, $V5$ et $V6$ sont utilisés pour incorporer les observations floues. Par conséquent, après avoir exécuté un algorithme classique d'inférence, la croyance de chaque variable aléatoire est mise à jour et les observations floues stockées dans les nœuds virtuels sont également incorporées dans les probabilités pour répondre à la requête principale.

Dans cet exemple, nous pouvons voir que FuZzy-PR-OWL 2 permet de manipuler en même temps l'incertitude due à l'aléatoire, l'imprécision et l'incomplétude.

- **Aléatoire** : nous sommes confrontés à un problème pour lequel notre connaissance sur le résultat de la prédiction est limitée, nous ne pouvons pas être certains si une patiente est diabétique ou non. Cependant, en utilisant FuZzy-PR-OWL 2, nous pouvons prédire le résultat qui dépend d'un évènement incertain sur la base de la théorie des probabilités.
- **Incomplétude** : Effectivement, FuZzy-PR-OWL 2 est très flexible et permet de générer des SSBN flous selon les observations disponibles, même avec des informations incomplètes, c'est-à-dire que certaines informations manquent. Par exemple, les valeurs pour les *pregnancies* et *skin_thickness* sont manquantes (valeur inconnue) pour le patient_1.
- **L'imprécision** : FuZzy-PR-OWL 2 permet de gérer des informations floues et de faire face à des événements probabilistes associés à une connaissance floue. Par exemple, les croyances du noeud BMI_patiente_1 sont mises à jour à 90,91% pour l'état *high* et à 9,09% pour l'état *medium*. Cela signifie que l'observation accordée au noeud BMI_patiente_1 est floue où la valeur de BMI de «patiente_1» peut être classé comme élevé et moyen simultanément avec certains degrés d'appartenance.

6.5.2 Validation

Dans cette section, une évaluation quantitative est présentée. Nous allons évaluer le système de prédiction en utilisant l'inférence probabiliste floue dans FuZzy-PR-OWL 2 en comparant le résultat obtenu avec l'ensemble de données correspondant manuellement dans Pima. Pour ce faire, nous avons examiné les patients stockés dans l'ensemble de données en se basant sur une inférence probabiliste floue et une inférence probabiliste classique. Les paramètres d'évaluation utilisés sont les suivants :

- Vrai positif (TP), le nombre de patientes que nous avons correctement prédit qu'elles ont du diabète.
- Vrai négatif (TN), le nombre de patientes que nous avons correctement prédit qu'elles ne sont pas diabétiques.
- Faux positif (FP), le nombre de patientes que nous avons prédit par erreur qu'elles ont du diabète.
- Faux négatif (FN), le nombre de patientes que nous avons prédit par erreur qu'elles ne sont pas diabétiques.

Le tableau 6.7 (la matrice de confusion) donne une illustration sur ces paramètres :

		Classification manuelle	
		diabétique	non diabétique
classification du système	diabétique	TP	FP
	non diabétique	FN	TN

TABLE 6.7 – Illustration des paramètres d'évaluation.

Dans notre étude, l'inférence probabiliste floue et l'inférence probabiliste classique ont été mises en œuvre dans le langage java et la bibliothèque FuzzyLite [126] pour la manipulation de la logique floue. De plus, l'évaluation quantitative dans cette étude est basée sur le calcul du rappel, de la précision et de la mesure F des résultats obtenus par inférence probabiliste floue et par inférence probabiliste classique.

- *Le rappel dans notre cas répond à la question : de toutes les patientes qui sont vraiment diabétiques, combien avons-nous étiquetés ?*
- *La précision répond à la question : de toutes les patientes étiquetées comme diabétiques, combien sont réellement diabétiques ?*
- *La métrique de mesure F est calculée en fonction du rappel et de la précision.*

La figure 6.11 montre donc les résultats obtenus par l'inférence probabiliste floue et l'inférence probabiliste. Elle montre que la prédiction en se basant sur l'inférence probabiliste floue a la plus haute précision, rappel et la mesure F qui sont respectivement d'environ 81%, 55% et 66%. Contrairement aux résultats obtenus en appliquant l'inférence probabiliste classique sont moins efficaces et la précision, rappel et F-mesure sont respectivement d'environ 74%, 39% et 51%, ce qui signifie que la prédiction basée sur l'inférence probabiliste floue est meilleure que celle obtenue avec la prédiction basée sur l'inférence probabiliste classique.

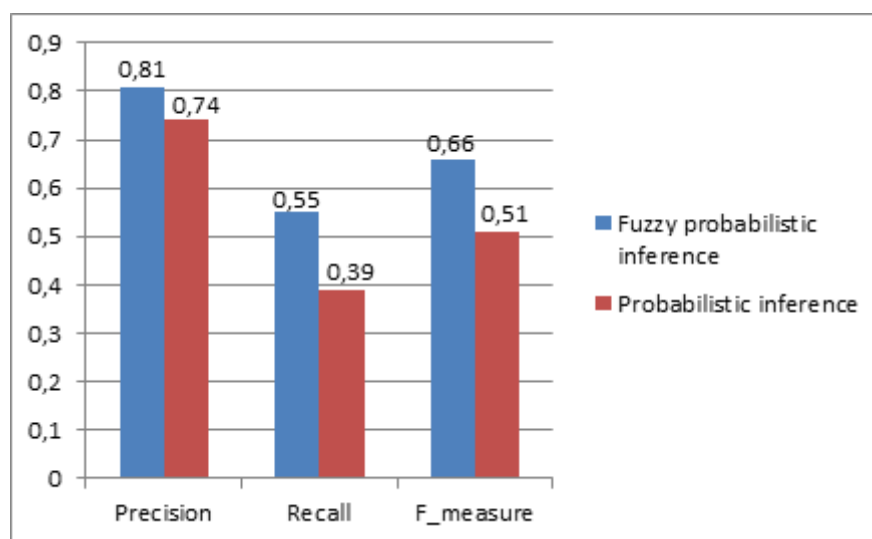


FIGURE 6.11 – Les résultats obtenus par une inférence probabiliste et une inférence probabiliste floue.

Nous pouvons constater une différence remarquable entre les mesures de performance utilisées. En effet, le raisonnement probabiliste flou dans notre solution améliore le raisonnement probabiliste classique en prenant en compte les connaissances vagues et imprécises impliquées dans les nœuds flous, où il intègre les degrés d'appartenance aux probabilités lors du calcul de l'inférence. Cependant, l'inférence probabiliste ignore ces nœuds et ne prend pas en compte la connaissance floue lors du calcul de l'inférence.

6.6 Conclusion

Dans ce chapitre, nous avons présenté une étude de cas permettant ainsi de confirmer l'intérêt de notre approche. Nous avons construit une ontologie probabiliste floue pour le diabète Type 2 pour l'héritage de PimaIndian. En effet, nous avons utilisé dans une première étape la méthodologie de construction d'ontologies probabilistes UMP-ST, dans une deuxième étape nous avons fuzzifié l'ontologie résultante pour tenir compte du flou. Nous avons également mis en avant une étude expérimentale pour tester les performances de l'inférence probabiliste floue par rapport à l'inférence probabiliste.

Conclusion Générale

Billan

Ce travail de recherche s'inscrit dans le cadre de la modélisation et le raisonnement sur des connaissances incertaines dans les ontologies. Une des lacunes des ontologies classiques c'est leur fondement théorique qui est basé sur des formalismes déterministes et booléens qui ne permettent pas de traiter l'incertitude.

Pour faire face à cette problématique, notre objectif consiste à traiter l'incertitude dans des ontologies riches en termes d'incertitude. Dans cette optique, nous avons proposé deux contributions majeures.

La première contribution nommée FzMEBN, vise à traiter le flou lié aux variables aléatoires dans les MEBNs en se basant sur une technique existante qui traite le flou dans les RBs classiques. Dans cette contribution nous avons focalisé sur deux volets à savoir la modélisation et le raisonnement :

- **La modélisation**, nous avons introduit à ce niveau la notion de fragments flous afin de permettre la distinction entre les nœuds flous et les nœuds précis.
- **L'inférence probabiliste floue**, il s'agit bien ici d'un processus d'inférence général qui permet d'incorporer des observations non certaines et floues avec des probabilités pour répondre à des requêtes spécifiques.

Quant à la deuxième contribution, nous avons introduit une nouvelle approche pour le traitement de l'incertitude dans les ontologies fondée sur notre première contribution. En effet, nous avons présenté une extension basée sur la logique floue du PR-OWL2 nommée FuZzy-PR-OWL2. Aussi, dans cette contribution nous avons mis l'accent sur la modélisation et le raisonnement :

- **La modélisation**, nous avons enrichi l'ontologie de haut niveau de PR-OWL 2 pour qu'elle représente la sémantique de notre FzMEBN. Par conséquent, elle permet concevoir des ontologies probabilistes floues.
- **L'inférence probabiliste floue**, les capacités du raisonnement offertes par les ontologies ont été étendues pour supporter une inférence ontologique probabiliste basée sur des observations floues.

Les contributions proposées dans notre thèse ont été évaluées d'une manière qualitative et d'une autre quantitative. Effectivement, nous avons présenté une étude de cas dans le domaine médical pour confirmer l'usage de nos contributions. En effet, nous avons réussi à exploiter notre approche pour modéliser la connaissance probabiliste et la connaissance floue ensemble. Nous avons aussi utilisé cette modélisation pour la classification des individus de l'héritage de Pimaïndian.

Ainsi, en faisant une analyse sur notre étude de cas, nous avons constaté que le système du raisonnement basé sur l'inférence probabiliste floue a abouti à une performance plus élevée que celui qui se base sur l'inférence probabiliste classique.

Nous pouvons donc conclure que le traitement de la connaissance probabiliste et la connaissance floue simultanément dans les ontologies est une voie de recherche très prometteuse.

Perspectives

Bien que les contributions proposées dans cette thèse traitent les connaissances probabilistes et les connaissances floues simultanément dans les ontologies ont montré leurs performances, notre travail reste limité et plusieurs dimensions doivent être abordées pour que nos approches atteignent leur plein potentiel. En effet, notre travail ouvre des perspectives scientifiques à court et à long terme pour améliorer et compléter le travail engagé dans cette thèse. Nous soulignons dans ce qui suit les recommandations futures qui nous semblent pertinentes pour l'amélioration de nos contributions :

Perspectives d'un point de vue théorique :

- Bien que l'applicabilité de notre approche FuZzy-PR-OWL 2 a été confirmée dans le présent document, les étapes de construction d'une ontologie probabiliste floue sur la base de FuZzy-PR-OWL 2 ne sont pas claires. Par conséquent, il semble très important de guider l'ontologiste pas à pas lors de la création d'une ontologie probabiliste floue. Dans cette optique, nous envisageons de proposer **une méthodologie complète** pour guider les ontologistes dans la construction d'ontologies probabilistes floues, allant de la spécification des besoins jusqu'à l'opérationnalisation de l'ontologie.
- Nous envisageons d'augmenter l'expressivité et les tâches de raisonnement de FuZzy-PR-OWL 2. En effet, nous nous intéressons à étudier comment construire une ontologie probabiliste floue formalisée en FuZzy-PR-OWL 2 en se basant sur les connaissances représentées dans une ontologie floue. Pour aboutir à cet objectif, nous avons fait le premier pas dans notre travail publié dans (Riali et al., 2019b) [127], qui vise à construire un RBF à partir d'une ontologie floue.
- Il est intéressant d'étendre nos approches pour qu'elles traitent d'autres types de connaissances d'une manière formelle et sémantique. Nous pensons à les mettre à jour pour qu'elles tiennent compte **des connaissances temporelles et contextuelles** : des connaissances qui changent en fonction de temps et du contexte.

Perspective d'un point de vue applicatif :

Gestion de l'incertain dans Le Web des objets (WdO) : plusieurs approches ont été proposées récemment pour gérer l'incertitude dans les WdO [128, 129]. Ces approches permettent la modélisation de services WdO incertains et l'évaluation de la composition incertaine en se basant sur des modèles bayésiens, sans tenir compte des représentations sémantiques des données incertaines.

Toutefois, une modélisation sémantique de ces services à l'aide d'une ontologie probabiliste floue nous semble souhaitable pour mieux représenter les données incertaines issues des services WdO et faciliter leur gestion en prenant en considération leur richesse en termes d'incertitude.

Annexe A

Annexe A : Langage PR-OWL2

Nous exposons dans cette partie les classes de l'ontologie de haut niveau de PROWL 2.

A.1 Variables aléatoires (RandomVariable)

Une variable aléatoire est une fonction mesurable qui associe les éléments d'un espace d'échantillon Ω à un espace mesurable E (des nombres réels). L'espace de l'échantillon représente tous les résultats possibles d'un événement ou d'une expérience aléatoire. Ainsi, la valeur de la variable aléatoire change d'un événement à un autre.

Dans PROWL 2, une variable aléatoire permet de définir l'incertitude liée à une propriété particulière définie dans OWL. Ainsi, la figure A.1 montre les principaux concepts pour définir des variables aléatoires. En effet, chaque variable aléatoire représente l'incertitude attachée à des formules du premier ordre (représentées par la classe `MExpression`) et pour la définir nous devons :

- Définir les arguments de la variable aléatoire et faire les correspondances entre ces arguments et leurs équivalences dans les propriétés définies en OWL, ceci est fait en utilisant une propriété d'objets nommée « *prowl2 :hasArgument* ».
- Donner les états possibles de la variable aléatoire en utilisant une propriété de données nommée « *prowl2 :hasPossibleValues* ».
- Représenter la distribution de probabilités via la classe `ProbabilityDistribution`, ensuite faire le lien entre cette distribution et la variable aléatoire via une propriété d'objets nommée « *prowl2 :hasProbabilityDistribution* ».
- Mettre en évidence de manière explicite la relation entre chaque variable aléatoire et la propriété de OWL qui présente l'incertitude, ce lien est défini en utilisant une propriété d'objets nommée « *prowl2 :definesUncertaintyOf* ».
- Définir une instance de la classe `MExpression` qui représente la formule du premier ordre qui correspond au variable aléatoire.

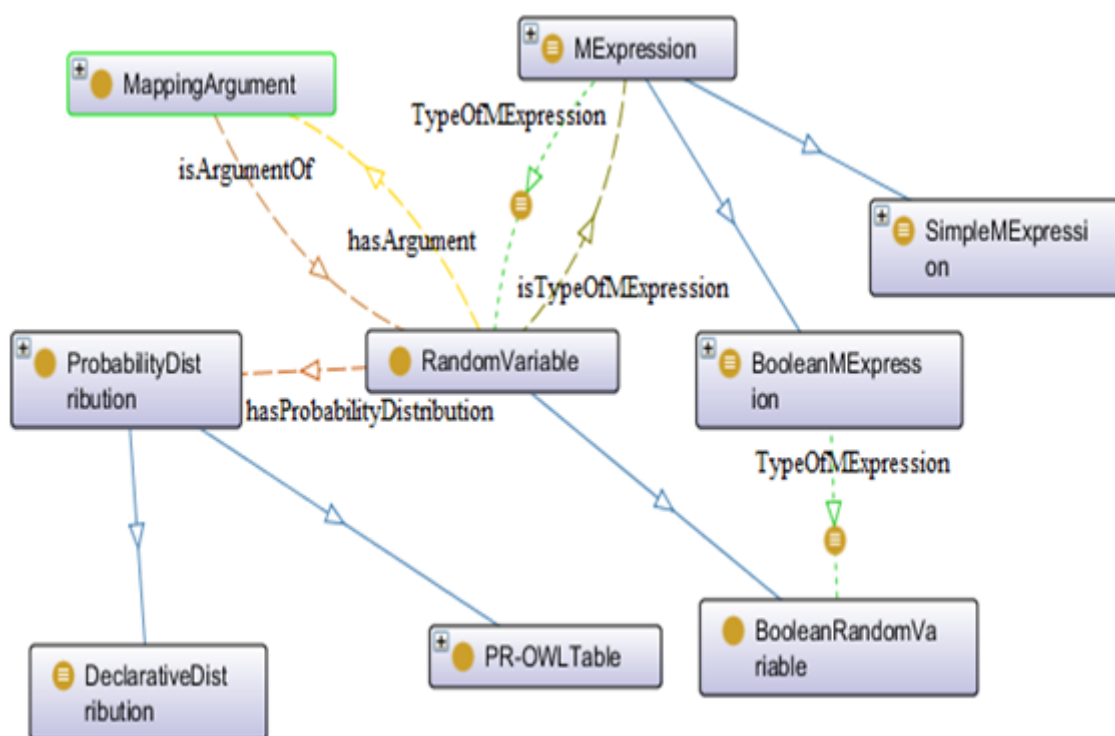


FIGURE A.1 – Les principaux éléments pour définir une variable aléatoire [130]

A.1.1 Variables aléatoires booléennes

Une variable aléatoire booléenne est un type particulier de variable aléatoire qui ne peut avoir que des valeurs booléennes comme valeurs possibles. Pour représenter ces valeurs, PROWL2 utilise une propriété de données nommée "**hasPossibleValues**" dont le rang est de type Booléen.

Afin de définir l'incertitude concernant cette propriété, nous devons créer une variable aléatoire booléenne "**RV.estAmie**" comme une instance de la classe **BooleanRandomVariable** qui définit l'incertitude de la propriété OWL "**estAmie**" et la mapper ainsi que ses arguments à cette propriété OWL. En effet, la variable "**RV.estAmie**" possède deux arguments, "**RV.estAmie.MA.personne1**" et "**RV.estAmie.MA.personne2**". Ainsi, sa distribution de probabilité est représentée par "**RV.estAmie.PT.dist1**"; son rang est booléen qui est hérité automatiquement de son type. La séquence présentée dans la figure A.2 montre les restrictions nécessaires pour définir la variable aléatoire "**RV.estAmie**" dans PROWL2.

```

1 Individu: RV.estAmie
2 Types:
3   pr-owl2:BooleanRandomVariable
4
5 Faits:
6   pr-owl2:hasArgument RV.estAmie.MA.personnel1,
7   pr-owl2:hasArgument RV.estAmie.MA.personne2,
8   pr-owl2:hasProbabilityDistribution RV.estAmie.PT.dist1 ,
9   pr-owl2:definesUncertaintyOf "&ex ; estAmie "^^xsd:anyURI

```

FIGURE A.2 – La définition de la variable RV.estAmie.

Par ailleurs, les opérateurs de la logique du premier ordre peuvent être exprimés en utilisant la classe `LogicalOperator` qui est un type spécial de la classe `BooleanRandomVariable` comme la montre la figure A.3.

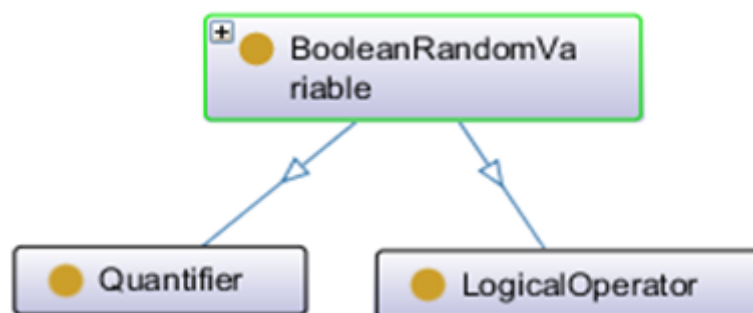


FIGURE A.3 – La classe BooleanRandomVariable [130]

Ces opérateurs logiques sont principalement utilisés pour exprimer des formules de la logique du premier ordre à l'aide d'expressions MEBN. Les opérateurs utilisés sont classés en deux types, les opérateurs binaires tel que : « *And* », « *Or* », « *EqualTo* » et « *Iff* » et des opérateurs unaires tel que « *Not* ».

- « **And** » représente l'opérateur « et » de la conjonction ;
- « **Ou** » représente l'opérateur « ou » de la disjonction ;
- « **EqualTo** » représente l'opérateur d'égalité « = » ;
- « **Implies** » représente l'opérateur d'implication « si alors » « \implies » ;
- « **Iff** » représente l'opérateur d'équivalence « si et seulement si » « \iff » ;
- « **Not** » représente l'opérateur "non" pour exprimer la négation.

Ainsi, les quantificateurs existentiel \exists et universel \forall sont exprimés en utilisant la classe **Quantifier**.

A.2 Arguments de mappage (MappingArgument)

Pour définir les deux arguments "**RV.estAmie.MA.personne1**" et "**RV.estAmie.MA.personne2**" de la variable "**RV.estAmie**", nous utilisons la classe **MappingArgument**. Cette classe a pour objectif de faire un mappage entre les arguments d'une variable aléatoire et les propriétés représentées dans OWL. Ainsi, afin de différencier entre l'argument qui fait référence au domaine et l'argument qui fait référence au rang d'une propriété en utilisant respectivement les relations **isSubjectIn** et **isObjectIn**.

En effet, le mappage entre l'argument de la variable et le domaine d'une propriété OWL est garanti en utilisant la propriété **isSubjectIn**. Cette propriété permet de référencier la propriété OWL où il appartient l'argument comme domaine.

Ainsi, le type d'argument de la variable aléatoire doit être compatible avec le type de domaine ou le rang de la propriété sur laquelle il pointe, ceci est garanti en utilisant la propriété **isSubstitutedBy**. Si la propriété est une propriété d'objets alors le rang de la propriété **isSubstitutedBy** sera une classe OWL, si la propriété est une propriété de données le rang de la propriété **isSubstitutedBy** sera un type de données (Double, String, etc). En outre, l'ordre d'apparaissant des arguments dans la variable aléatoire est représenté en se basant sur la propriété **hasArgumentNumber** dont le rang est un entier naturel. La figure A.4 montre les restrictions nécessaires pour la définition des arguments de la variable RV.estAmie.

```

1 Individual: RV.estAmie.MA.personne1
2   Types:
3     | pr-owl2:MappingArgument
4   Facts:
5     pr-owl2:isArgumentOf RV.estAmie,
6     pr-owl2:hasArgumentNumber 1 ,
7     pr-owl 2 : isSubstitutedBy "&ex ; Personne"^^xsd:anyURI ,
8     pr-owl2 : isSubjectIn "&ex ; estAmie "^^xsd:anyURI
9
10 Individual: RV.estAmie.MA.personne2
11   Types:
12     | pr-owl2:MappingArgument
13   Facts:
14     pr-owl2:isArgumentOf RV.estAmie,
15     pr-owl2:hasArgumentNumber 2 ,
16     pr-owl 2 : isSubstitutedBy "&ex ; Personne"^^xsd:anyURI ,
17     pr-owl 2 : isObjectIn "&ex ; estAmie "^^xsd:anyURI

```

FIGURE A.4 – la définition des arguments de la variable RV.estAmie

A.3 Distribution des probabilités

Chaque variable aléatoire possède une distribution de probabilités qui permet d'exprimer de manière formelle la connaissance probabiliste et l'incertitude liées à cet variable. En effet, la classe **ProbabilityDistribution** est utilisée pour définir les distributions locales pour certaines

variables aléatoires (les variables qui correspondent à des nœuds résidents). Une distribution de probabilité peut être exprimée à l'aide d'un pseudo code LPD ou via une table PR-OWL comme ils est illustré dans la figure A.5.

- Un **Distribution Declarative** est définie par un pseudo code qui suit une syntaxe spécifique et bien déterminée afin de quantifier les relations de dépendances entre les entités liées de façon beaucoup plus compact et flexible. En effet, Une distribution déclarative est une distribution qui est exprimée par un type de données `xsd:string` en se basant sur la propriété `hasDeclaration`.
- Les **Tables PR-OWL** expriment les distributions de probabilités de manière plus interoperable, mais ne sont pas assez souples pour représenter des distributions complexes notamment dans les cas où un nœud a plusieurs parents.

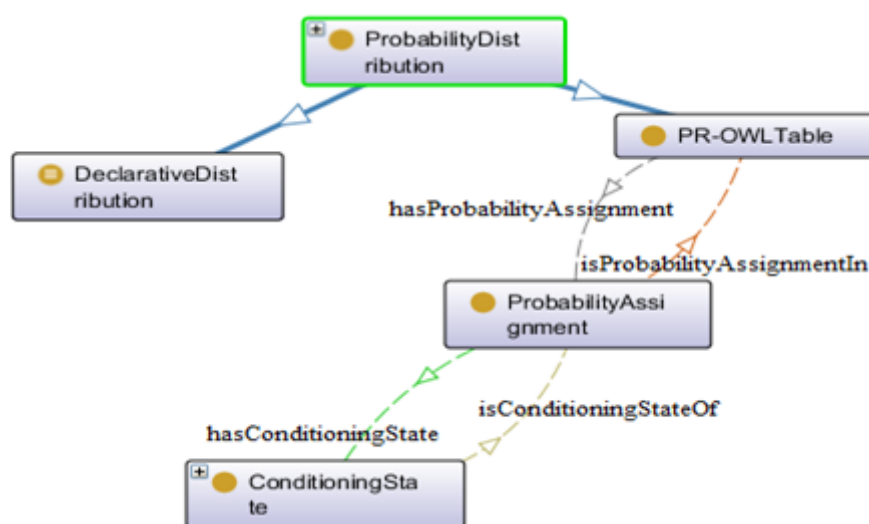


FIGURE A.5 – Les classes nécessaires pour la définition d'une distribution de probabilités[130]

A.4 Classe MEBN Expression

Les formules et les termes de la logique du premier ordre dans PROWL2 sont définis en utilisant la classe `MExpression`. En effet, pour chaque variable aléatoire, une instance de la classe **MExpression** sera créée pour définir sa sémantique. En outre, le nombre et les types d'arguments définis pour cette instance doivent être les mêmes que le nombre et les types d'arguments définis pour la variable qu'elle référence. Par ailleurs, deux types d'expressions MEBN sont distingués comme il est montré dans la figure A.6.

- **SimpleMExpression** représente une formule ou un terme atomique, également appelé atome. Elle contient que des arguments de type *OrdinaryVariableArgument* ou *constantArgument*.

- **BooleanMExpression** représente une formule booléenne qui vise à exprimer la sémantique d'une variable aléatoire booléenne (représentée par une instance de la classe *BooleanRandomVariable*).

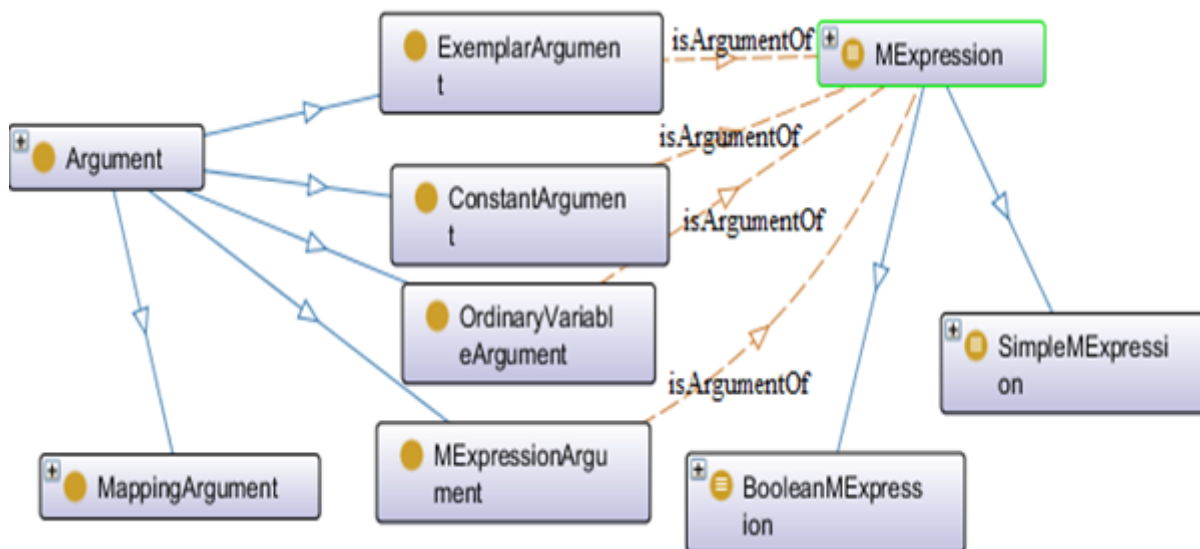


FIGURE A.6 – Les principaux éléments pour définir une MExpression [130]

A.5 Classe Exemplar

Les individus de cette classe sont des variables liées à des quantificateurs existentiels ou universels utilisées dans des expressions MEBN.

A.6 Classe OrdinaryVariable

Les éléments de cette classe sont des variables libres utilisées pour la construction des expressions MEBN.

A.7 Arguments

La classe argument contient plusieurs sous-classes d'arguments qui peuvent être classés en deux types, à savoir :

- La classe **ConstantArgument** : cette classe est utilisée pour représenter des formules ou des termes utilisant des constantes de données et / ou d'objets. A titre d'exemple : *égal (grade ("Mohammed"), «Professeurs»)*, où « *Mohammed* » et « *Professeurs* » sont des constantes représentant respectivement un enseignant et un grade universitaire.
- La classe **ExemplarArgument** : cette classe est utilisée pour représenter des variables liées dans une expression. Par exemple : l'expression ci-après déclare que certains régions sont dangereuses : $\exists R \text{ est-Dangereuse}(R)$, où R est un *Exemplar* de type région.

- La classe **OrdinaryVariableArgument** : cette classe est utilisée pour représenter des variables libres qui apparaissent dans une formule ou un terme. Par exemple, est-Dangereuse (R), où R est une variable libre de type région qui peut être remplacée par un individu de la classe *Région*.
- La classe **MExpressionArguemnt** : cette classe est utilisée construire des formules ou de termes complexes. Par exemple : égal (grade (Enseignant 1),grade (Enseignant2)) , où Enseignant 1 et Enseignant 2 sont des variables libres du type Enseignant.

Plus de détails sur ces classes peuvent être trouvés dans [\[130\]](#).

Bibliographie

- [1] Zhongli Ding, Yun Peng, and Rong Pan. Bayesowl : Uncertainty modeling in semantic web ontologies. In *Soft computing in ontologies and semantic web*, pages 3–29. Springer, 2006.
- [2] Yi Yang and Jacques Calmet. Ontobayes : An ontology-driven uncertainty model. In *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, volume 1, pages 457–463. IEEE, 2005.
- [3] Paulo Cesar G Da Costa, Kathryn B Laskey, and Kenneth J Laskey. Pr-owl : A bayesian ontology language for the semantic web. In *Uncertainty Reasoning for the Semantic Web I*, pages 88–107. Springer, 2006.
- [4] Rommel N Carvalho, Kathryn B Laskey, and Paulo CG Costa. Pr-owl—a language for defining probabilistic ontologies. *International Journal of Approximate Reasoning*, 91 :56–79, 2017.
- [5] Zhi-Yun Zheng, Zhuo-Yun Liu, Lun Li, Dun Li, and Zhen-Fei Wang. A fuzzy mebn ontology language based on owl2. In *FSDM*, pages 71–80, 2016.
- [6] Hlel Emna, Jamoussi Salma, Turki Mohamed, and Ben Hamadou Abdelmajid. Probabilistic ontology definition meta-model. In *International Conference on Intelligent Decision Technologies*, pages 243–254. Springer, 2016.
- [7] Hlel Emna, Jamoussi Salma, Turki Mohamed, and Ben Hamadou Abdelmajid. Building probabilistic ontologies based on meta-model podm. In *International Conference on Hybrid Intelligent Systems*, pages 288–297. Springer, 2016.
- [8] Abdul-Wahid Mohammed, Yang Xu, and Ming Liu. Knowledge-oriented semantics modelling towards uncertainty reasoning. *SpringerPlus*, 5(1) :706, 2016.
- [9] Silvia Calegari and Davide Ciucci. Fuzzy ontology, fuzzy description logics and fuzzy-owl. In *International Workshop on Fuzzy Logic and Applications*, pages 118–126. Springer, 2007.

- [10] Hanen Ghorbel, Afef Bahri, and Rafik Bouaziz. Fuzzy ontologies building method : Fuzzy ontomethodology. In *2010 Annual Meeting of the North American Fuzzy Information Processing Society*, pages 1–8. IEEE, 2010.
- [11] Fernando Bobillo and Umberto Straccia. Fuzzy ontology representation using owl 2. *International Journal of Approximate Reasoning*, 52(7) :1073–1094, 2011.
- [12] Kenneth J Laskey and Kathryn B Laskey. Uncertainty reasoning for the world wide web : Report on the urw3-xg incubator group. In *URSW*. Citeseer, 2008.
- [13] Peter Fisher, Alexis Comber, and Richard Wadsworth. Approaches to uncertainty in spatial data. *Fundamentals of spatial data quality*, pages 43–59, 2006.
- [14] Matteo Caglioni and Giovanni Fusco. Formal ontologies and uncertainty. in geographical knowledge. *Tema. Journal of Land Use, Mobility and Environment*, 2014.
- [15] Cyril De Runz. *Imperfection, temps et espace : modélisation, analyse et visualisation dans un SIG archéologique*. PhD thesis, Université de Reims-Champagne Ardenne, 2008.
- [16] Jim F Baldwin, Jonathan Lawry, and Trevor P Martin. A mass assignment theory of the probability of fuzzy events. *Fuzzy Sets and Systems*, 83(3) :353–367, 1996.
- [17] Ishak Riali, Messaouda Fareh, and Hafida Bouarfa. Fzmebn : Toward a general formalism of fuzzy multi-entity bayesian networks for representing and reasoning with uncertain knowledge. In *ICEIS (1)*, pages 520–528, 2017.
- [18] Ishak Riali, Messaouda Fareh, and Hafida Bouarfa. Fuzzy probabilistic ontology approach : A hybrid model for handling uncertain knowledge in ontologies. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 15(4) :1–20, 2019.
- [19] T Berners-Lee, James Hendler, and O Lassila. A new form of web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific american*, 2002.
- [20] Michael K Smith, Chris Welty, and Deborah L McGuinness. Owl web ontology language guide, w3c recommendation 10 february 2004. 2004. *WWW : http ://www. w3. org/TR/2004/REC-owl-guide-20040210/(Februar 2004)*, 23, 2014.
- [21] Thomas R Gruber. A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2) :199–220, 1993.
- [22] Pierdaniele Giaretta and Nicola Guarino. Ontologies and knowledge bases towards a terminological clarification. *Towards very large knowledge bases : knowledge building & knowledge sharing*, 25(32) :307–317, 1995.

- [23] WN Borst. Construction of engineering ontologies, centre for telematica and information technology, 1997.
- [24] Rudi Studer, V Richard Benjamins, and Dieter Fensel. Knowledge engineering : principles and methods. *Data & knowledge engineering*, 25(1-2) :161–197, 1998.
- [25] Asunción Gómez-Pérez. Ontological engineering : A state of the art. *Expert Update : Knowledge Based Systems and Applied Artificial Intelligence*, 2(3) :33–43, 1999.
- [26] Franz Baader, Diego Calvanese, Deborah McGuinness, Peter Patel-Schneider, and Daniele Nardi. *The description logic handbook : Theory, implementation and applications*. Cambridge university press, 2003.
- [27] Fernando Bobillo, Miguel Delgado, and Juan Gómez-Romero. Crisp representations and reasoning for fuzzy ontologies. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 17(04) :501–530, 2009.
- [28] Tim Berners lee. Architecture : semantic web. <https://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-1.html>. Accessed : 2019-06-30.
- [29] Tim Bray, Jean Paoli, C Michael Sperberg-McQueen, Eve Maler, and François Yergeau. Extensible markup language (xml). *World Wide Web Journal*, 2(4) :27–66, 1997.
- [30] Henry S Thompson, David Beech, Murray Maloney, and Noah Mendelsohn. Xml schema part 1 : structures second edition. *W3C recommendation*, 39, 2004.
- [31] Dan Brickley. Rdf vocabulary description language 1.0 : Rdf schema. <http://www.w3.org/TR/rdf-schema/>, 2004.
- [32] Dan Brickley, Ramanathan V Guha, and Brian McBride. Rdf schema 1.1. *W3C recommendation*, 25 :2004–2014, 2014.
- [33] Lee W Lacy. *OWL : Representing information using the web ontology language*. Trafford Publishing, 2005.
- [34] Robert Neches, Richard E Fikes, Tim Finin, Thomas Gruber, Ramesh Patil, Ted Senator, and William R Swartout. Enabling technology for knowledge sharing. *AI magazine*, 12(3) :36–36, 1991.
- [35] Lotfi Asker Zadeh. The concept of a linguistic variable and its application to approximate reasoning—i. *Information sciences*, 8(3) :199–249, 1975.
- [36] Michel Thiry. Combining value and project management into an effective programme management model. *International journal of project management*, 20(3) :221–227, 2002.

- [37] Raanan Lipshitz and Orna Strauss. Coping with uncertainty : A naturalistic decision-making analysis. *Organizational behavior and human decision processes*, 69(2) :149–163, 1997.
- [38] Frances J Milliken. Three types of perceived uncertainty about the environment : State, effect, and response uncertainty. *Academy of Management review*, 12(1) :133–143, 1987.
- [39] Didier Dubois and Henri Prade. Panorama des nouvelles méthodes de traitement de l’incertitude et de l’imprécision. *Institut de Recherche en Informatique de Toulouse (IRIT), CNRS et Université Paul Sabatier. Toulouse*, 2010.
- [40] Jean-Marc Tacnet, Valérie November, Didier Richard, and Mireille Batton-Hubert. Expertise, décision et incertitude : jusqu’où une approche interdisciplinaire est-elle possible dans le cadre de la gestion intégrée des risques naturels en montagne? In *Conférence Outils Pour Décider Ensemble 2010 (OPDE)*, pages 18–p, 2010.
- [41] Philippe Smets. Imperfect information : Imprecision and uncertainty. In *Uncertainty management in information systems*, pages 225–254. Springer, 1997.
- [42] Hlel. Emna. *Prob-Ont : Une nouvelle méthode de construction d’ontologies probabilistes en utilisant les réseaux bayésiens*. PhD thesis, Faculté des Sciences Économiques et de Gestion de Sfax, 2016.
- [43] Douglas L Allaire. *Uncertainty assessment of complex models with application to aviation environmental systems*. PhD thesis, Citeseer, 2009.
- [44] Douglas Allaire and Karen Willcox. Uncertainty assessment of complex models with application to aviation environmental policy-making. *Transport Policy*, 34 :109–113, 2014.
- [45] Lotfi Asker Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy sets and systems*, 1(1) :3–28, 1978.
- [46] Zdzisław Pawlak. Rough sets. *International journal of computer & information sciences*, 11(5) :341–356, 1982.
- [47] Zdzisław Pawlak. Rough set approach to knowledge-based decision support. *European journal of operational research*, 99(1) :48–57, 1997.
- [48] Arthur P Dempster. A generalization of bayesian inference. *Journal of the Royal Statistical Society : Series B (Methodological)*, 30(2) :205–232, 1968.
- [49] Glenn Shafer. *A mathematical theory of evidence*, volume 42. Princeton university press, 1976.

- [50] Bal-Bourai Safia and Mokhtari Aicha. Poss-owl 2 : Possibilistic extension of owl 2 for an uncertain geographic ontology. *Procedia Computer Science*, 35 :407–416, 2014.
- [51] Rihab Ben Lamine and Mohamed Nazih Omri. Information retrieval model based on possibilistic ontology. *International Journal of Computer Science*, 9 :3, 2012.
- [52] Amandine Bellenger and Sylvain Gatepaille. Uncertainty in ontologies : Dempster-shafer theory for data fusion applications. *arXiv preprint arXiv :1106.3876*, 2011.
- [53] Loukia Karanikola and Isambo Karali. Towards a dempster–shafer fuzzy description logic—handling imprecision in the semantic web. *IEEE Transactions on Fuzzy Systems*, 26(5) :3016–3026, 2018.
- [54] Franck Dernoncourt. Introduction à la logique floue. *Licence Commons-BY-SA*. <http://franckdernoncourt.developpez.com/tutoriels/algo/introduction-logique-oue/>, Paris, 2011.
- [55] George J Klir and Bo Yuan. Fuzzy sets and fuzzy logic : theory and applications. *Upper Saddle River*, page 563, 1995.
- [56] François Chevie and François Guély. La logique floue. *Cahier technique*, 191, 1998.
- [57] Richard O Duda, Peter E Hart, and Nils J Nilsson. Subjective bayesian methods for rule-based inference systems. In *Readings in artificial intelligence*, pages 192–199. Elsevier, 1981.
- [58] David Heckerman, Dan Geiger, and David M Chickering. Learning bayesian networks : The combination of knowledge and statistical data. *Machine learning*, 20(3) :197–243, 1995.
- [59] Stuart Russell and Peter Norvig. Artificial intelligence : A modern approach prentice-hall. *Englewood cliffs, NJ*, 26, 1995.
- [60] Nick Chater, Joshua B Tenenbaum, and Alan Yuille. Probabilistic models of cognition : Conceptual foundations, 2006.
- [61] Stuart J Russell and Peter Norvig. *Artificial intelligence : a modern approach*. Malaysia; Pearson Education Limited,, 2016.
- [62] Patrick Naïm, Pierre-Henri Wuillemin, Philippe Leray, Olivier Pourret, and Anna Becker. Réseaux bayésiens. *Eyrolles, Paris*, 3, 1999.
- [63] Veronique Delcroix, Karima Sedki, and FrançOis-Xavier Lepoutre. A bayesian network for recurrent multi-criteria and multi-attribute decision problems : Choosing a manual wheelchair. *Expert systems with applications*, 40(7) :2541–2551, 2013.

- [64] David Maxwell Chickering. Learning bayesian networks is np-complete. In *Learning from data*, pages 121–130. Springer, 1996.
- [65] Peter Spirtes, Clark N Glymour, Richard Scheines, David Heckerman, Christopher Meek, Gregory Cooper, and Thomas Richardson. *Causation, prediction, and search*. MIT press, 2000.
- [66] Gregory F Cooper and Edward Herskovits. A bayesian method for the induction of probabilistic networks from data. *Machine learning*, 9(4) :309–347, 1992.
- [67] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society : Series B (Methodological)*, 39(1) :1–22, 1977.
- [68] Judea Pearl. Bayesian networks : A model of self-activated memory for evidential reasoning. 1985.
- [69] Judea Pearl. Fusion, propagation, and structuring in belief networks. *Artificial intelligence*, 29(3) :241–288, 1986.
- [70] Steffen L Lauritzen and David J Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society : Series B (Methodological)*, 50(2) :157–194, 1988.
- [71] Finn V Jensen. Bayesian updating in causal probabilistic networks by local computations. *An introduction to Bayesian networks*, 1996.
- [72] Gregory F Cooper. The computational complexity of probabilistic inference using bayesian belief networks. *Artificial intelligence*, 42(2-3) :393–405, 1990.
- [73] Uffe Kjærulff. Approximation of bayesian networks through edge removals. In *AALBORG UNIVERSITY*. Citeseer, 1993.
- [74] Kathryn Blackmond Laskey. Mebn : A language for first-order bayesian knowledge bases. *Artificial intelligence*, 172(2-3) :140–178, 2008.
- [75] Paulo CG Costa, Marcelo Ladeira, Rommel N Carvalho, Kathryn B Laskey, Laécio L Santos, and Shou Matsumoto. A first-order bayesian tool for probabilistic ontologies. In *Proceedings of the Twenty-First International Florida Artificial Intelligence Research Society Conference*, pages 631–636, 2008.
- [76] Richard Haberlin. *Probabilistic ontology reference architecture and development methodology*. PhD thesis, 2013.

- [77] Keyvan Golestan, Fakhri Karray, and Mohamed S Kamel. High level information fusion through a fuzzy extension to multi-entity bayesian networks in vehicular ad-hoc networks. In *Proceedings of the 16th International Conference on Information Fusion*, pages 1180–1187. IEEE, 2013.
- [78] Keyvan Golestan, Fakhri Karray, and Mohamed S Kamel. An integrated approach for fuzzy multi-entity bayesian networks and semantic analysis for soft and hard data fusion. In *2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–8. IEEE, 2015.
- [79] Gabriel Moura and Mauro Roisenberg. Probabilistic fuzzy bayesian network. In *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pages 476–482. IEEE, 2015.
- [80] Peng-cheng Li, Guo-hua Chen, Li-cao Dai, and Li Zhang. A fuzzy bayesian network approach to improve the quantification of organizational influences in hra frameworks. *Safety science*, 50(7) :1569–1583, 2012.
- [81] Golam Kabir, Rehan Sadiq, and Solomon Tesfamariam. A fuzzy bayesian belief network for safety assessment of oil and gas pipelines. *Structure and Infrastructure Engineering*, 12(8) :874–889, 2016.
- [82] Limao Zhang, Xianguo Wu, Yawei Qin, Mirosław J Skibniewski, and Wenli Liu. Towards a fuzzy bayesian network based approach for safety risk analysis of tunnel-induced pipeline damage. *Risk Analysis*, 36(2) :278–301, 2016.
- [83] Muhammad Saiful Islam and Madhav Nepal. A fuzzy-bayesian model for risk assessment in power plant projects. *Procedia Computer Science*, 100 :963–970, 2016.
- [84] Han-Saem Park, Ji-Oh Yoo, and Sung-Bae Cho. A context-aware music recommendation system using fuzzy bayesian networks with utility theory. In *International conference on Fuzzy systems and knowledge discovery*, pages 970–979. Springer, 2006.
- [85] Ali Ben Mrad, Mohamed Amine Maalej, Veronique Delcroix, Sylvain Piechowiak, and Mohamed Abid. Fuzzy evidence in bayesian network. In *2011 International Conference of Soft Computing and Pattern Recognition (SoCPaR)*, pages 486–491. IEEE, 2011.
- [86] Hao Tang and Shi Liu. Basic theory of fuzzy bayesian networks and its application in machinery fault diagnosis. In *Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007)*, volume 4, pages 132–137. IEEE, 2007.
- [87] Christopher Fogelberg, Vasile Palade, and Phil Assheton. Belief propagation in fuzzy bayesian networks. In *1st International Workshop on Combinations of Intelligent Methods and Applications (CIMA) at ECAI'08*, pages 19–24, 2008.

- [88] Nicholas Ryhajlo, Liessman Sturlaugson, and John W Sheppard. Diagnostic bayesian networks with fuzzy evidence. In *2013 IEEE AUTOTESTCON*, pages 1–8. IEEE, 2013.
- [89] Henry E Kyburg. Probabilistic reasoning in intelligent systems : networks of plausible inference by judea pearl. *The Journal of Philosophy*, 88(8) :434–437, 1991.
- [90] Xiao Li. On the use of virtual evidence in conditional random fields. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing : Volume 3-Volume 3*, pages 1289–1297. Association for Computational Linguistics, 2009.
- [91] Yun Peng, Shenyong Zhang, and Rong Pan. Bayesian network reasoning with uncertain evidences. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 18(05) :539–564, 2010.
- [92] Heping Pan and Lin Liu. Fuzzy bayesian networks—a general formalism for representation, inference and learning with hybrid bayesian networks. *International Journal of Pattern Recognition and Artificial Intelligence*, 14(07) :941–962, 2000.
- [93] CG Costa Paulo. Bayesian semantics for the semantic web. *PhD, George Mason University*, 2005.
- [94] Daphne Koller, Alon Levy, and Avi Pfeffer. P-classic : A tractable probabilistic description logic. *AAAI/IAAI*, 1997(390-397) :14, 1997.
- [95] Rodrigo B Polastro, Fabio G Cozman, Felipe I Takiyama, and Kate C Revoredo. Computing inferences for credal alc terminologies. In *Proceedings of the 8th International Conference on Uncertainty Reasoning for the Semantic Web-Volume 900*, pages 94–97. CEUR-WS. org, 2012.
- [96] Rosalba Giugno and Thomas Lukasiewicz. P-shoq (d) : A probabilistic extension of shoq (d) for probabilistic ontologies in the semantic web. In *JELIA*, volume 2, pages 86–97. Springer, 2002.
- [97] Ismail Ilkan Ceylan and Rafael Penaloza. The bayesian description logic *bel*. In *International Joint Conference on Automated Reasoning*, pages 480–494. Springer, 2014.
- [98] I Ceylan, Julian Mendez, and Rafael Peñaloza. The bayesian ontology reasoner is born ! CEUR, 2015.
- [99] The bayesian ontology language author=Ceylan, Ismail Ilkan and Penaloza, Rafael, journal=Journal of Automated Reasoning, volume=58, number=1, pages=67–95, year=2017, publisher=Springer.

- [100] Yoshio Fukushige. Representing probabilistic relations in rdf. In *Proceedings of the 2005 International Conference on Uncertainty Reasoning for the Semantic Web-Volume 173*, pages 106–107. Citeseer, 2005.
- [101] Octavian Udrea, VS Subrahmanian, and Zoran Majkic. Probabilistic rdf. In *2006 IEEE International Conference on Information Reuse & Integration*, pages 172–177. IEEE, 2006.
- [102] Rommel N Carvalho, Kathryn B Laskey, and Paulo CG Da Costa. Uncertainty modeling process for semantic technology. *PeerJ Computer Science*, 2 :e77, 2016.
- [103] Mouna Ben Ishak, Philippe Leray, and Nahla Ben Amor. Ontology-based generation of object oriented bayesian networks. In *Bayesian Modeling Applications Workshop (BMAW-11)*. Citeseer, 2011.
- [104] Daphne Koller and Avi Pfeffer. Object-oriented bayesian networks. In *Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence*, pages 302–313. Morgan Kaufmann Publishers Inc., 1997.
- [105] Stefan Fenz. An ontology-based approach for constructing bayesian networks. *Data & Knowledge Engineering*, 73 :73–88, 2012.
- [106] Anouar Tadlaoui Mouenis, Khaldi Mohamed, and Aammou Souhaib. Towards probabilistic ontology based on bayesian networks. 2014.
- [107] Boris Motik, Peter F Patel-Schneider, Bijan Parsia, Conrad Bock, Achille Fokoue, Peter Haase, Rinke Hoekstra, Ian Horrocks, Alan Ruttenberg, Uli Sattler, et al. Owl 2 web ontology language : Structural specification and functional-style syntax. *W3C recommendation*, 27(65) :159, 2009.
- [108] Emna Hlel, Salma Jamoussi, and Abdelmajid Ben Hamadou. A new method for building probabilistic ontology (prob-ont). In *Information Retrieval and Management : Concepts, Methodologies, Tools, and Applications*, pages 1409–1434. IGI Global, 2018.
- [109] Fernando Bobillo and Umberto Straccia. Representing fuzzy ontologies in owl 2. In *International Conference on Fuzzy Systems*, pages 1–6. IEEE, 2010.
- [110] Mauro Mazzieri, Aldo Franco Dragoni, and UPD Marche. A fuzzy semantics for semantic web languages. In *ISWC-URSW*, pages 12–22, 2005.
- [111] Fernando Bobillo, Miguel Delgado, and Juan Gómez-Romero. Delorean : A reasoner for fuzzy owl 2. *Expert Systems with Applications*, 39(1) :258–272, 2012.

- [112] Fernando Bobillo, Miguel Delgado, and Juan Gómez-Romero. Delorean : a reasoner for fuzzy owl 1.1. In *Proceedings of the Fourth International Conference on Uncertainty Reasoning for the Semantic Web-Volume 423*, pages 13–24. CEUR-WS. org, 2008.
- [113] Fernando Bobillo and Umberto Straccia. The fuzzy ontology reasoner fuzzydl. *Knowledge-Based Systems*, 95 :12–34, 2016.
- [114] Obed Jules, Abdelhakim Hafid, and Mohamed Adel Serhani. Bayesian network, and probabilistic ontology driven trust model for sla management of cloud services. In *2014 IEEE 3rd International Conference on Cloud Networking (CloudNet)*, pages 77–83. IEEE, 2014.
- [115] Obed Jules. Modèle de confiance et ontologie probabiliste pilotés par réseaux bayésiens pour la gestion des accords de services dans l’environnement de services infonuagiques. 2015.
- [116] Richard Haberlin, Paulo C da Costa, and Kathryn B Laskey. Probabilistic ontology architecture for a terrorist identification decision support system. Technical report, EMSOLUTIONS INC ARLINGTON VA, 2014.
- [117] Juan Antonio Morente-Molinera, Robin Wikström, Enrique Herrera-Viedma, and Christer Carlsson. A linguistic mobile decision support system based on fuzzy ontology to facilitate knowledge mobilization. *Decision Support Systems*, 81 :66–75, 2016.
- [118] Christer Carlsson, Matteo Brunelli, and József Mezei. Decision making with a fuzzy ontology. *Soft Computing*, 16(7) :1143–1152, 2012.
- [119] Shaker El-Sappagh and Mohammed Elmogy. A fuzzy ontology modeling for case base knowledge in diabetes mellitus domain. *Engineering science and technology, an international journal*, 20(3) :1025–1040, 2017.
- [120] Ali Ben Mrad. *Observations probabilistes dans les réseaux bayésiens*. PhD thesis, 2015.
- [121] David Poole, Clinton Smyth, and Rita Sharma. Semantic science : Ontologies, data and probabilistic theories. In *Uncertainty Reasoning for the Semantic Web I*, pages 26–40. Springer, 2006.
- [122] Paulo CG Costa. *Bayesian semantics for the Semantic Web*. Citeseer, 2005.
- [123] Pima indians diabetes. <https://www.kaggle.com/kumargh/pimaindiansdiabetescsv>. Accessed : 2017-01-15.
- [124] Unbbayes overview. <http://unbbayes.sourceforge.net/>. Accessed : 2019-01-01.

- [125] Leonard Adelman. *Evaluating decision support and expert systems*. Wiley New York, 1992.
- [126] Juan Rada-Vilela. The fuzzylite libraries for fuzzy logic control. <http://www.fuzzylite.com/>. Accessed : 2017-01-23.
- [127] Ishak Riali, Messaouda Fareh, and Hafida Bouarfa. A semantic approach for handling probabilistic knowledge of fuzzy ontologies. 2019.
- [128] Samir Awad, Abdelhamid Malki, Mimoun Malki, Mahmoud Barhamgi, and Djamal Benslimane. Composing wot services with uncertain data. *Future Generation Computer Systems*, 101 :940–950, 2019.
- [129] Abdelhamid Malki, Mahmoud Barhamgi, Sidi-Mohamed Benslimane, Djamal Benslimane, and Mimoun Malki. Composing data services with uncertain semantics. *IEEE Transactions on Knowledge and Data Engineering*, 27(4) :936–949, 2014.
- [130] Rommel Novaes Carvalho. *Probabilistic ontology : representation and modeling methodology*. George Mason University, 2011.