

## CHAPITRE 2

# LA LOGIQUE FLOUE ET LES ONTOLOGIES

### 1 Introduction

Ce chapitre s'intéresse à la présentation du formalisme de représentation de connaissances le plus adapté dans le Web sémantique qui est les logiques de descriptions, Nous arborons quelques limitations de ce formalisme pour en fin introduire une de ses extensions qui est à logique de descriptions floue.

### 2 Ontologies et modèles de connaissance

La modélisation des connaissances consiste à représenter un ensemble de données ou connaissances sous une forme adaptée pour qu'un opérateur humain et/ou machine, puisse les interpréter et les manipuler. Une représentation est définie selon un modèle qui fournit les règles syntaxiques de modélisation, appelées la syntaxe [25]. Le modèle peut être muni d'une sémantique, logique par exemple, pour définir clairement le sens de ce qui est modélisé. Le modèle est dit formel si les modélisations basées sur ce modèle peuvent être interprétées syntaxiquement et sémantiquement sans ambiguïté.

Les modèles de représentation de connaissance utilisés en ingénierie ontologique peuvent être regroupés selon les paradigmes conceptuels qu'ils réifient. Sont ainsi distingués :

- les modèles à base de Frames ;
- le modèle des Réseaux Sémantiques;
- les modèles des Logiques de Description.

Chacun de ces modèles de représentation est implémenté dans un ou plusieurs langages implémentant une partie ou la totalité du modèle, en particulier des langages adaptés au Web et utilisant la syntaxe XML .Les langages implémentant ces modèles sont souvent opérationnels, c'est-à-dire qu'ils offrent des mécanismes de raisonnement, et servent alors à représenter des ontologies déjà opérationnelles. D'autres ne font que permettre la spécification déclarative de connaissances [26,27]. Le modèle des logiques de descriptions étant celui que nous utilisons dans nos travaux, nous en détaillerons dans ce qui suit.

### 3 La représentation de connaissances à l'aide des logiques de descriptions

Il n'existe pas de méthode universelle pour concevoir des systèmes à base de connaissances, mais un courant de recherche très actif s'est développé autour de ces idées. Ce courant de recherche, qui s'est nourri des études effectuées sur la logique des prédicats, les réseaux sémantiques et les langages de frames, a donné naissance à une famille de langages de représentation appelés logiques de descriptions (LDs), ou encore logiques terminologiques.

La qualité principale de cette famille de langages de représentation est le pouvoir de représenter la connaissance d'un domaine d'application par un moyen clair, formel et structuré. Le nom de logique de descriptions se rapporte, d'une part à la description de concepts utilisés pour décrire un domaine et d'autre part à la sémantique basée sur la logique qui peut être donnée par une transcription en logique des prédicats du premier ordre [26].

La logique de descriptions (LDs) a été développée comme une extension des frames et des réseaux sémantiques, qui ne possédaient pas de sémantique formelle basée sur la logique. Les LDs décrivent les concepts d'un domaine en utilisant des concepts atomiques, correspondant à des prédicats unaires, et des rôles atomiques, correspondant à des prédicats binaires décrivant les relations entre les objets/concepts du domaine. Les rôles sont spécifiés à l'aide de constructeurs fournis par le langage formel des LDs [27].

La plupart des logiques de descriptions divisent la connaissance en deux niveaux :

- **les informations terminologiques:** définition des notions basiques ou dérivées et de comment elles sont reliées entre elles. Ces informations sont "génériques" ou "globales", vraies dans tous les modèles et pour tous les individus.
- **les informations sur les individus:** ces informations sont "spécifiques" ou "locales", vraies pour certains individus particuliers.

Toutes les connaissances sont alors prises en compte selon deux niveaux : la représentation et la manipulation des concepts et des rôles relèvent du niveau terminologique (appelé aussi T-Box); la description et la manipulation des individus relèvent du niveau factuel ou niveau assertionnel (appelé aussi A-Box).

### 3.1 Famille des langages LD

Ils existent de nombreuses logiques de descriptions qui se distinguent par la richesse des constructeurs qu'elles proposent à la représentation des connaissances souhaitées. La logique nommée AL (Attributive Language) qui a été introduite par Schmib et Smolka en 1991 est l'une parmi elles. Cette logique est minimale, dans le sens où elle est la moins expressive (c.-à-d. propose le moins de constructeurs). Nous décrivons par la suite la syntaxe et la sémantique de la logique AL [27].

#### 3.1.1 La syntaxe d'AL

La grammaire d'AL est donnée par le tableau suivant :

$C, D \rightarrow A$ (concept atomique)
$\top$ (Le concept universel)
$\perp$ (Le concept le plus spécifique)
$\neg A$ (la négation atomique)
$C \cap D$ (l'intersection)
$\exists R.T$ (Restriction existentielle limitée)
$\forall R.C$ (restriction universelle complète)

**Table 2.1** Syntaxe de la logique AL

A est concept atomique, C, D sont des concepts composés et R est un rôle atomique (AL ne permet pas la spécification de rôles à l'aide de constructeurs : rôles composés) [28].

- L'interprétation de  $\top$  est le domaine  $\Delta$  tout entier tandis que celle de  $\perp$  se réduit à l'ensemble vide.
- Le constructeur  $\neg A$  : est utilisé pour évoquer la négation qui ne peut être appliquée qu'à un concept atomique, c'est-à-dire les individus pour une interprétation qui n'appartiennent pas au concept atomique A.
- Le constructeur  $C \cap D$  : permet de faire la conjonction de deux concepts composés, ce qui représente l'ensemble des individus appartenant à la fois au concept C et au concept D pour une interprétation.
- Le quantificateur existentiel  $\exists R.T$  : désigne l'ensemble des individus, membres du domaine du rôle R pour une interprétation donnée.

- Le quantificateur universel  $\forall R.C$  : désigne l'ensemble des individus du domaine du rôle  $R$  qui sont en relation, par le biais de  $R$ , qu'avec les individus du concept  $C$ , pour une interprétation donnée.

Les concepts et les rôles (c'est-à-dire des relations entre concepts) atomiques constituent les entités élémentaires d'une T-box. Ces derniers peuvent être combinés au moyen de constructeurs pour former des concepts et des rôles composés. Par exemple, le concept composé  $\text{Male} \cap \text{Femelle}$  est le résultat de l'utilisation du constructeur  $\cap$  sur les concepts atomiques  $\text{Mâle}$  et  $\text{Femelle}$ . L'interprétation du concept ainsi composé est « l'ensemble des individus qui appartiennent à la fois au concept  $\text{Male}$  et au concept  $\text{Femelle}$  ».

### 3.1.2 La sémantique formelle d'AL

La sémantique d'une LD est donnée au moyen d'une interprétation  $I$  qui est un couple  $(\Delta^I, I)$  où :

$\Delta^I$  : est le domaine d'interprétation. C'est un ensemble non vide d'individus.

$I$  : est la fonction d'interprétation qui fait correspondre à un concept atomique  $A$  un ensemble  $A^I$  tel que  $A^I \subseteq \Delta^I$  et à chaque rôle atomique  $R$  une relation binaire  $R^I \subseteq \Delta^I \times \Delta^I$ .

La sémantique de AL défini plus haut est donnée par la table suivante [29]:

$T^I = \Delta^I$
$\perp^I = \Phi$
$(\neg A)^I = \Delta^I \setminus A^I$
$(C \cap D)^I = C^I \cap D^I$
$(\forall R.C)^I = \{a \in \Delta^I \mid \forall b, \text{ if } (a,b) \in R^I \text{ then } b \in C^I\}$
$(\exists R.T)^I = \{a \in \Delta^I \mid \exists b. (a,b) \in R^I\}$

**Table 2.2** Sémantique de la logique AL

La logique de description AL peut être enrichie par les constructeurs suivants:

- ◆  $O$  : qui permet la description de concepts par l'énumération d'individus nommés.
- ◆  $U$  : désigne l'union de concepts.
- ◆  $E$  : la quantification existentielle complète.
- ◆  $C$  : la négation complète.
- ◆  $I$  : les rôles inverses.

- ♦  $H$  : l'inclusion entre rôles.
- ♦ Les constructeurs  $F$ ,  $Q$  et  $N$  sont trois variantes de la contrainte de cardinalité sur le rôle.

Tableau 2.3 donne une vue d'ensemble de constructeurs enrichissant AL avec leur syntaxe et leur sémantique.

Constructeur	Syntaxe du Constructeur	Sémantique du constructeur
[O]	$\{a_1, a_2, \dots, a_n\}$	$\{a_1^I, a_2^I, \dots, a_n^I\}$
[U]	$C \sqcup D$	$C^I \cup D^I$
[E]	$\exists R.C$	$\{a \in \Delta^I \mid \{\exists b. (a, b) \in R^I\} \wedge b \in C^I\}$
[C]	$\neg C$	$\Delta^I \setminus C^I$
[I]	$R_1^{-1}$	$\{(y, x) \mid (x, y) \in R_1^I\}$
[Q]	$\geq n R.C$	$\{a, b \in \Delta^I \mid \{(a, b) \in R^I\} \wedge b \in C^I \mid \geq n\}$
	$\leq n R.C$	$\{a, b \in \Delta^I \mid \{(a, b) \in R^I\} \wedge b \in C^I \mid \leq n\}$
	$=n R.C$	$\{a, b \in \Delta^I \mid \{(a, b) \in R^I\} \wedge b \in C^I \mid =n\}$
[R]	$R_1 \sqcap R_2$	$R_1^I \cap R_2^I$
.....	.....	.....

**Table 2.3** Exemple de constructeurs de rôles et concepts pour étendre AL

La première colonne contient la lettre qui désigne le constructeur, la deuxième sa syntaxe d'utilisation et la dernière sa sémantique [29].

Il est important de noter qu'il existe une autre façon d'étendre une LD. La spécification d'un ensemble de rôles transitifs  $NR^+$ , constitue une extension par ajout de contraintes sur l'interprétation des rôles (désignée par la lettre  $R^+$ ), qui permet l'expression de rôles transitifs. Une dernière extension, symbolisée par la lettre ( $D$ ), ajoute le support des types primitifs.

La nomenclature des LD dicte que pour chaque constructeur ajouté, il faut agglutiner la lettre correspondante au nom de la logique originale. Par conséquent, La différence entre AL et ALC, vient du constructeur de négation complète (C). Ainsi, la logique AL, enrichie de l'union (U) et de la quantification existentielle complète (E), se nomme ALUE. Il faut noter que l'appellation ALC équivaut à ALUE (l'union et la quantification existentielle complète s'expriment par la négation complète et inversement car :

$C \cup D \Leftrightarrow \neg (\neg C \cap \neg D)$  et  $\exists R.C \Leftrightarrow \neg \forall R. \neg C$ .

La lettre  $S$  désigne la logique  $ALC$  additionnée de  $R^+$ .

### 3.2 La T-box et la A-box

La modélisation des connaissances d'un domaine à l'aide des LDs comporte deux niveaux, la Tbox et la A-box.

#### 3.2.1 Le niveau terminologique T-box

Le niveau terminologique T-box : décrit les connaissances générales d'un domaine et contient les déclarations des primitives conceptuelles organisées en concepts et relations.

Ces déclarations décrivent les propriétés des concepts et des relations et constituent donc une définition intentionnelle des connaissances [30].

#### 3.2.2 Le niveau assertionnel A-box

Décrit les connaissances factuelles d'un domaine et représente une configuration précise. Il contient les déclarations d'individus, instances des concepts qui ont été définis dans la T-box. Plusieurs A-box peuvent être associées à une même T-box ; chacune représente une configuration constituée d'individus, et utilise les concepts et rôles de la T-box pour l'exprimer [30].

#### 3.2.3 Un exemple d'une base de connaissances

Un exemple d'une base de connaissances est fourni au Tableau 2.4. Le côté gauche présente un exemple de T-box dans laquelle les noms commençant par une lettre majuscule, comme Humain,

Animal, Femelle ou Male, désignent des concepts et ceux commençant par une minuscule, comme relationParentEnfant, désignent des rôles [31].

<b>T-box</b>	<b>A-box</b>
$Femelle \subseteq T \cap \neg M\grave{a}le$ $M\grave{a}le \subseteq T \cap \neg Femelle$ $Animal \equiv M\grave{a}le \cup Femelle$ $Humain \subseteq Animal$ $Femme \equiv Humain \cap Femelle$ $Homme \equiv Humain \cap \neg Femelle$ $M\grave{e}re \equiv Femme \cap \exists relationParentEnfant$ $P\grave{e}re \equiv Homme \cap \exists relationParentEnfant$ $M\grave{e}reSansFille \equiv M\grave{e}re \cap$ $\forall relationParentEnfant. \neg Femme$ $RelationParentEnfant \subseteq TR$	$Humain (Ahmed)$ $Femelle (Fatima)$ $Femme (Aïcha)$ $Humain (Mohamed)$ $\neg Femelle (Mohamed)$ $Homme (Ali)$ $relationParentEnfant (Aïcha, Ahmed)$ $relationParentEnfant (Mohamed, Ali)$

**Table 2.4** Base de connaissances composée d'une T-box et d'une A-box

### 3.3 L'inférence dans les logiques de description

Elle s'effectue au niveau terminologique ou assertionnel. Quatre principaux problèmes se présentent pour chacun de ces deux niveaux que le moteur d'inférence essaye de résoudre.

➤ **L'inférence au niveau terminologique** : quatre principaux problèmes d'inférence se présentent au niveau terminologique :

- Satisfiabilité : Trouver tous les concepts insatisfiables.
- Subsomption : Calculer la hiérarchie de subsomption ou taxonomie des concepts.
- Trouver les concepts équivalents.
- Trouver les concepts disjoints.

➤ **L'inférence au niveau assertionnel** : le niveau assertionnel comprend quatre principaux problèmes d'inférence :

- Cohérence de la ABox : les assertions définies restent cohérentes avec la TBox.
- Vérification d'instance : vérifier que chaque instance respecte la définition de son concept.
- Vérification de rôle : vérifier si les rôles de l'instance sont correctement utilisés.

Dans Doyle [31] sont également discutées les limitations des logiques de descriptions dans le cadre de la conception de systèmes d'intelligence artificielle réalistes.

Ces limitations sont de plusieurs sortes, parmi lesquelles figurent :

- la difficulté de représenter des relations entre rôles.
- la représentation de quantifications sur les relations.
- les traitements numériques en général.
- les définitions récursives.
- les relations n-aires.

La structure des logiques de descriptions classique qui ne peut exprimer des faits qu'avec "vrai" ou "faux" limite leur champs d'action dans des techniques et des applications de l'Intelligence Artificielle comme l'Aide à la décision, le Web sémantique, etc. qui veulent imiter le raisonnement et l'esprit humain. C'est à dire des techniques qui s'appuient sur l'incertitude pour leur bon fonctionnement.

De ce faite, on peut affirmer que les logiques de descriptions classiques ont une expressivité considérable, mais elles sont très faibles lors ce qu'on on veut modéliser un domaine dont les connaissances et les informations sont vagues et imprécises. Pour cette raison il y avait beaucoup de propositions pour étendre les logiques de descriptions par des théories mathématiques qui traitent l'incertain et l'imprécis, et comme résultat, c'est la naissance des logiques de descriptions floues.

## **4 La logique floue**

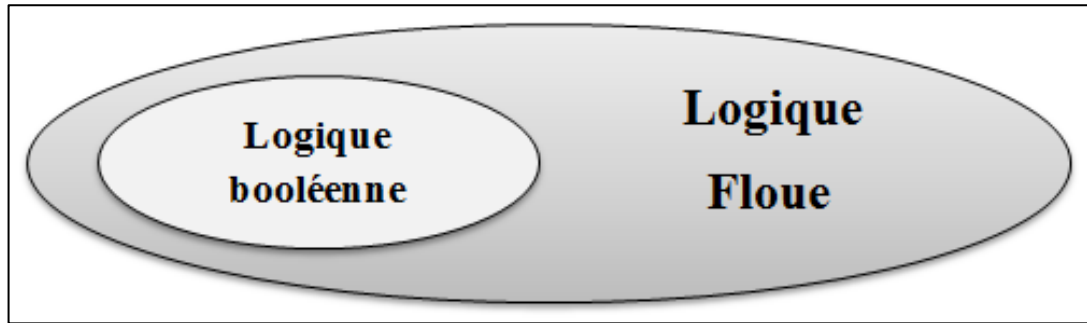
### **4.1 Définition**

La logique floue a été introduite par L. Zadeh comme extension de la logique booléenne. Elle permet à une proposition d'être dans un autre état que vrai ou faux. La logique floue s'appuie sur la théorie mathématique des ensembles flous, introduite aussi par L. Zadeh.

Ces ensembles permettent de modéliser l'incertitude et l'imprécision qui caractérisent souvent les représentations humaines des connaissances [33]. Chaque ensemble flou est défini par sa variable linguistique et sa fonction d'appartenance. Aujourd'hui, la logique floue est arrivée à maturité et elle est utilisée dans différents domaines d'application. Notre intérêt porte sur son utilisation pour la représentation des données imprécises et incertaines dans



le cadre du web sémantique, et plus précisément son intégration dans la définition des ontologies, devenant ainsi floues.



**Figure 2.1** Logique floue vs logique booléenne

#### **4.2 Les logiques de descriptions floues**

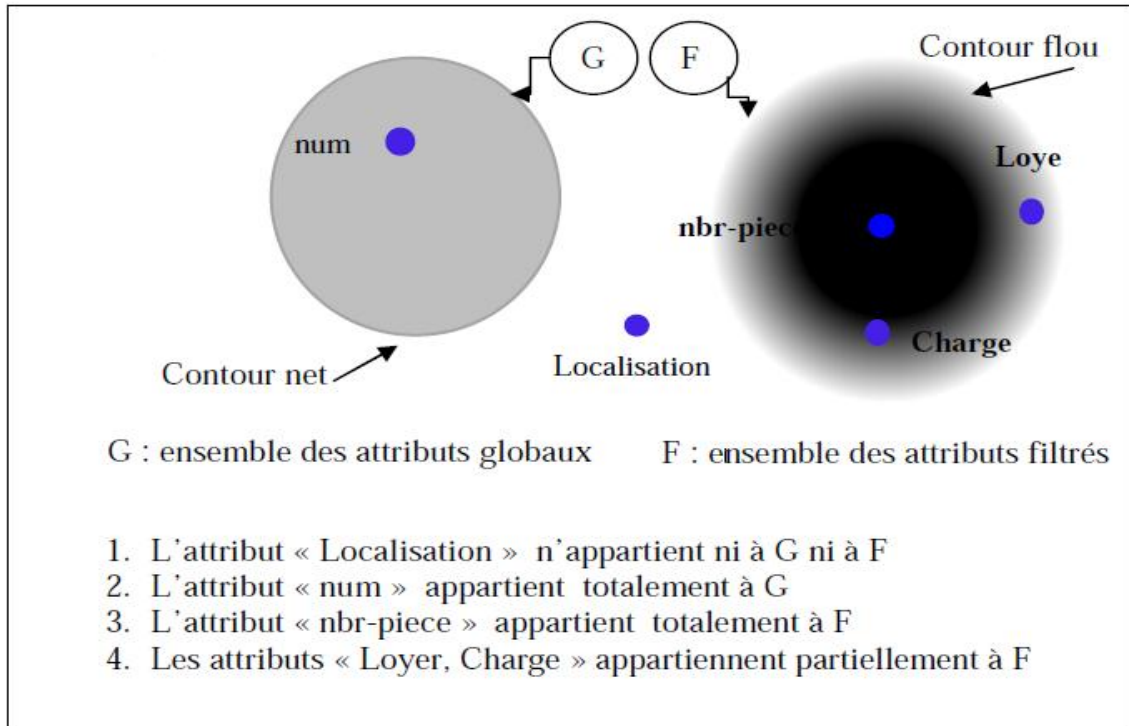
Les Logiques de descriptions floues (Fuzzy DLs) sont des extensions des logiques de descriptions classiques, elles ont été proposées en tant que langages pouvant représenter et raisonner sur des connaissances vagues ou imprécises.

Depuis le travail de Yen [32], de nombreuses propositions ont été faites pour introduire des fonctionnalités floue dans les logiques de descriptions et dans les langages du Web sémantique, puisque la théorie des sous-ensembles flous a pour but de fournir une représentation des classes et des relations avec une appartenance progressive, il serait plus approprié de les utilisés comme modèle pour représenter des concepts ayant une définition un peu vague et imprécise.

#### **4.3 La sémantique des LDs floues**

Les LDs classiques sont interprétées grâce à des concepts ensemblistes classiques : ensemble, relation binaire, appartenance, etc. les extensions floues des LDs ont une sémantique exprimée grâce à la théorie des sous-ensembles flous, un élément appartient à l'ensemble avec un certain degré. Plus formellement, soit  $X$  un ensemble d'éléments, un sous-ensemble flou  $A$  de  $X$ , est défini par une fonction d'appartenance  $\mu_A(x)$ , ou simplement  $A(x)$ .

Cette fonction affecte tout  $x \in X$  à une valeur comprise entre 0 et 1 ( $[0, 1]$ ) qui représente le degré dont lequel cet élément appartient à  $X$  [33].



**Figure 2.2** Comparaison d'un ensemble classique et d'un ensemble flou

Dans ce nouveau cadre, la théorie des ensembles classiques et les opérations logiques sont appliquées par des fonctions mathématiques spéciales. Plus précisément le complément flou qui serait une fonction unaire de la forme  $C: [0, 1] \rightarrow [0, 1]$ , l'intersection et l'union floue sont assurées par deux fonctions binaires de la forme  $T: [0, 1] \times [0, 1] \rightarrow [0, 1]$  et  $U: [0, 1] \times [0, 1] \rightarrow [0, 1]$ , appelées respectivement les opérations t-norme et t-conorm, et l'implication floue assurée par une fonction binaire de la forme  $J: [0, 1] \times [0, 1] \rightarrow [0, 1]$ . Afin de produire les compléments, les conjonctions, disjonctions flous et leurs implications, ces fonctions doivent satisfaire certaines propriétés mathématiques, Par exemple, les opérateurs doivent satisfaire les propriétés suivantes [34,35] :

$$C(0) = 1, C(1) = 0, T(1, a) = a \text{ et } U(0, a) = a.$$

Une relation floue  $R$  de  $X \times X$  est transitive si  $\forall a, b, c \in X, R(a, c) \geq \{R(a, b), R(b, c)\}$ ,  $R$  est réflexive si  $\forall a \in X, R(a, a) = 1$ , tandis qu'elle est dite i réflexive si  $\forall a \in X, R(a, a) = 0$ . L'inverse d'une relation floue  $R: X \times Y \rightarrow [0, 1]$  est une relation floue de la forme  $R^-: Y \times X \rightarrow [0, 1]$  défini comme  $R^-(b, a) = R(a, b)$ .

Les extensions floues des logiques de descriptions ont une sémantique exprimée grâce à la théorie des sous-ensembles flous : pour une interprétation  $I = (\Delta^I, .^I)$ , un concept  $C$  s'interprète comme un sous-ensemble flou  $C^I$  de  $\Delta^I$ , un rôle  $R$  comme une relation binaire floue

$R^I$  de  $\Delta^I \times \Delta^I$ , l'appartenance d'une instance  $a$  à un concept  $C$  qui est  $a^I \in C^I$  dans le cas classique- devient le degré d'appartenance de  $a^I$  à  $C^I$  :  $\mu_{C^I}(a^I)$ .

La fonction l'interprétation  $I$  doit satisfaire les équations suivantes [36]:

$$\begin{aligned}
 &\forall x \in \Delta^I \\
 &\mu_{\top^I}(x) = 1 \\
 &\mu_{\perp^I}(x) = 0 \\
 &\mu_{(C \cap D)^I}(x) = \min \{ \mu_{C^I}(x), \mu_{D^I}(x) \} \\
 &\mu_{(C \cup D)^I}(x) = \max \{ \mu_{C^I}(x), \mu_{D^I}(x) \} \\
 &\mu_{\neg^I}(x) = 1 - \mu_{C^I}(x) \\
 &\mu_{(\forall R.C)^I}(x) = \inf_{y \in \Delta^I} \{ \max \{ 1 - \mu_{R^I}(x, y), \mu_{C^I}(y) \} \} \\
 &\mu_{(\exists R.C)^I}(x) = \sup_{y \in \Delta^I} \{ \max \{ 1 - \mu_{R^I}(x, y), \mu_{C^I}(y) \} \}
 \end{aligned}$$

**Table 2.5** équations de l'interprétation standard

Ces équations sont l'interprétation standard de la conjonction, disjonction, la négation et la quantification, respectivement.

Notant que la sémantique de  $(\exists R.C)^I(x)$  peut être vue comme une formule ouverte du 1er ordre de la forme :  $\forall y F_R(x, y) \wedge F_C(y)$ , quant au quantificateur existentiel  $\exists$  est considérée comme une disjonction sur les éléments du domaine [37].

De même,  $\mu_{(\forall R.C)^I}(x) = \inf_{y \in \Delta^I} \{ \max \{ 1 - \mu_{R^I}(x, y), \mu_{C^I}(y) \} \}$  est liée à la formule ouverte de premier ordre  $\forall y \neg F_R(x, y) \vee F_C(y)$ , où le quantificateur universel  $\forall$  est considérée comme une conjonction sur les éléments du domaine. La sémantique des axiomes et des assertions classiques ( $C \sqsubseteq D$ ,  $C \sqsupseteq D$ ,  $a : C \sqcap R(a, b)$ ) est inchangée, sauf que, l'inclusion et l'égalité sont des opérations entre sous-ensembles flous et que l'appartenance devient un degré d'appartenance.

Concernant les inférences des LDs flous, nous nous sommes concentrés sur la « *subsumption floue* ». Ce terme désigne le faite, étant donnés deux concepts flous  $C$  et  $D$  de tester si (oui ou non)  $C \sqsubseteq D$ . on aura  $C \sqsubseteq D$  si, pour tout modèle  $I$  de la base de connaissance,  $C^I \sqsubseteq D^I$ , au sens de l'inclusion entre ensembles flous,  $\mu_{C^I}(x) \leq \mu_{D^I}(x) \forall x \in \Delta^I$ .

certain travaux considèrent un « *degré de subsumption* » entre deux concepts, étant donné deux concepts  $C$  et  $D$ , le degré de subsumption de  $C$  dans  $D$  est la valeur  $F_{\sqsubseteq}(C, D) \in [0,1]$  défini par :  $F_{\sqsubseteq}(C, D) = \inf \{ F_{\sqsubseteq}(C^I, D^I) \}$ .

$I$  : modèle de la base de connaissance.

$F_{\sqsubseteq}$  : est une mesure du degré d'inclusion entre ensemble flou.

#### 4.4 Des logiques de descriptions floues

Les LDs floues diffèrent entre elles principalement par le moyen par lequel elles introduisent le flou c'est-à-dire par les éléments syntaxiques (constructeurs, axiomes, assertions) pour lesquels l'interprétation classique s'avère insuffisante [35]. L'auteur dans a distingué trois de ces moyens :

- ✓ Utilisation de prédicats flous dans les domaines concrets.
- ✓ Utilisation de modificateurs sur les concepts.
- ✓ Association aux axiomes ou aux assertions de la base de connaissances d'informations concernant leurs degrés de vérité.

### 5 Ontologies floues

#### 5.1 Définition

Ontologie floue se base sur l'intégration de la logique floue à la définition des ontologies précises pour représenter les incertitudes et les imprécisions dans le cadre du web sémantique. Nous pouvons définir l'ontologie floue comme une ontologie qui possède, en plus des composants de base (concepts, relations, axiomes et instances), de nouveaux composants conformes à la logique floue : les concepts flous et les relations floues. Ces derniers assurent la représentation des parties floues d'un univers de discours [35].

#### 5.2 Les composants d'ontologies floues

Une ontologie floue comprend 6 composants : les concepts précis, les concepts flous, les relations précises, les relations floues, les axiomes et les instances. Les concepts et les relations précis gardent les mêmes définitions et jouent les mêmes rôles que dans une ontologie classique.

Les concepts et les relations flous, définis selon la logique floue, servent à la représentation des éléments flous du domaine modélisé. Ils ont des répercussions sur l'expression des axiomes et des instances [38].

### 5.2.1 Les concepts flous

Un concept flou est un concept défini sur la base d'une valeur particulière d'une variable linguistique (ex. Classification-Âge) relative à l'univers de discours, avec l'incertitude et l'imperfection touchant une propriété fuzzifiée, , rendue floue.

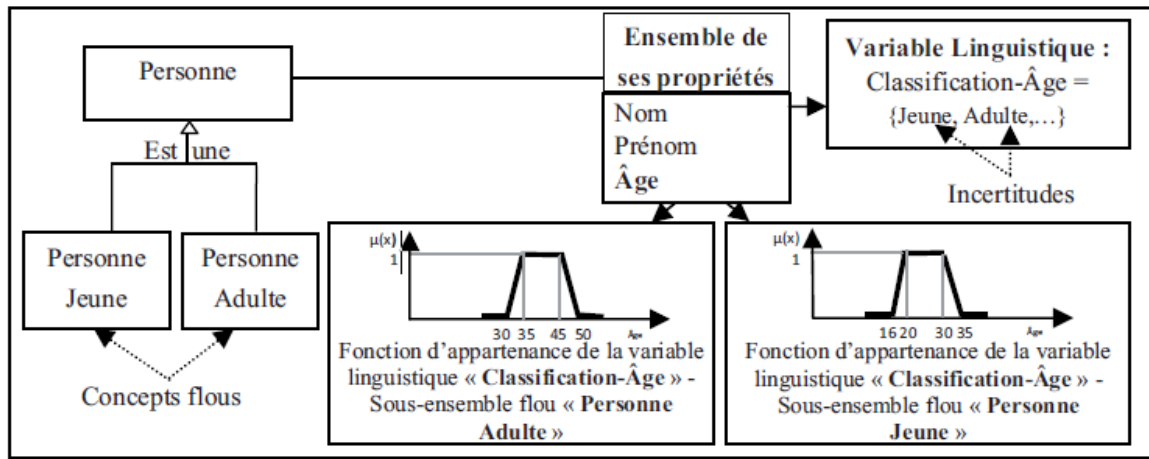


Figure 2.3 Exemple de concepts flous [35]

La Figure 2.3 montre deux concepts flous : Personne Jeune et Personne Adulte.

### 5.2.2 Les relations floues

Une relation floue est une relation binaire (i.e., entre deux concepts d'un domaine) dont la sémantique est décrite par une valeur particulière d'une variable linguistique.

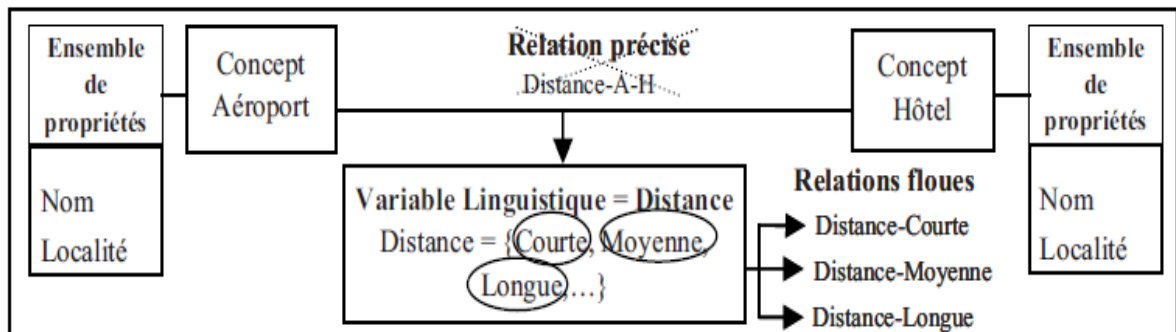


Figure 2.4 Exemple de relations floues [35]

La Figure 2.4 montre trois relations floues : Distance-courte, Distance-Moyenne et Distance-Longue.

### 5.2.3 Les instances

Dans les ontologies floues, nous trouvons deux types d'instances : les instances des concepts et des relations précis et les instances des concepts et des relations flous. L'appartenance d'une instance a un composant flou n'est pas sure.

Elle est déterminée par une variable probabiliste prenant ses valeurs entre 0 et 1, et déterminée par une fonction d'appartenance [37].

### 5.2.4 Les axiomes

Dans le cas d'ontologies floues, les axiomes servent aussi à exprimer les formules de calcul des fonctions d'appartenance.

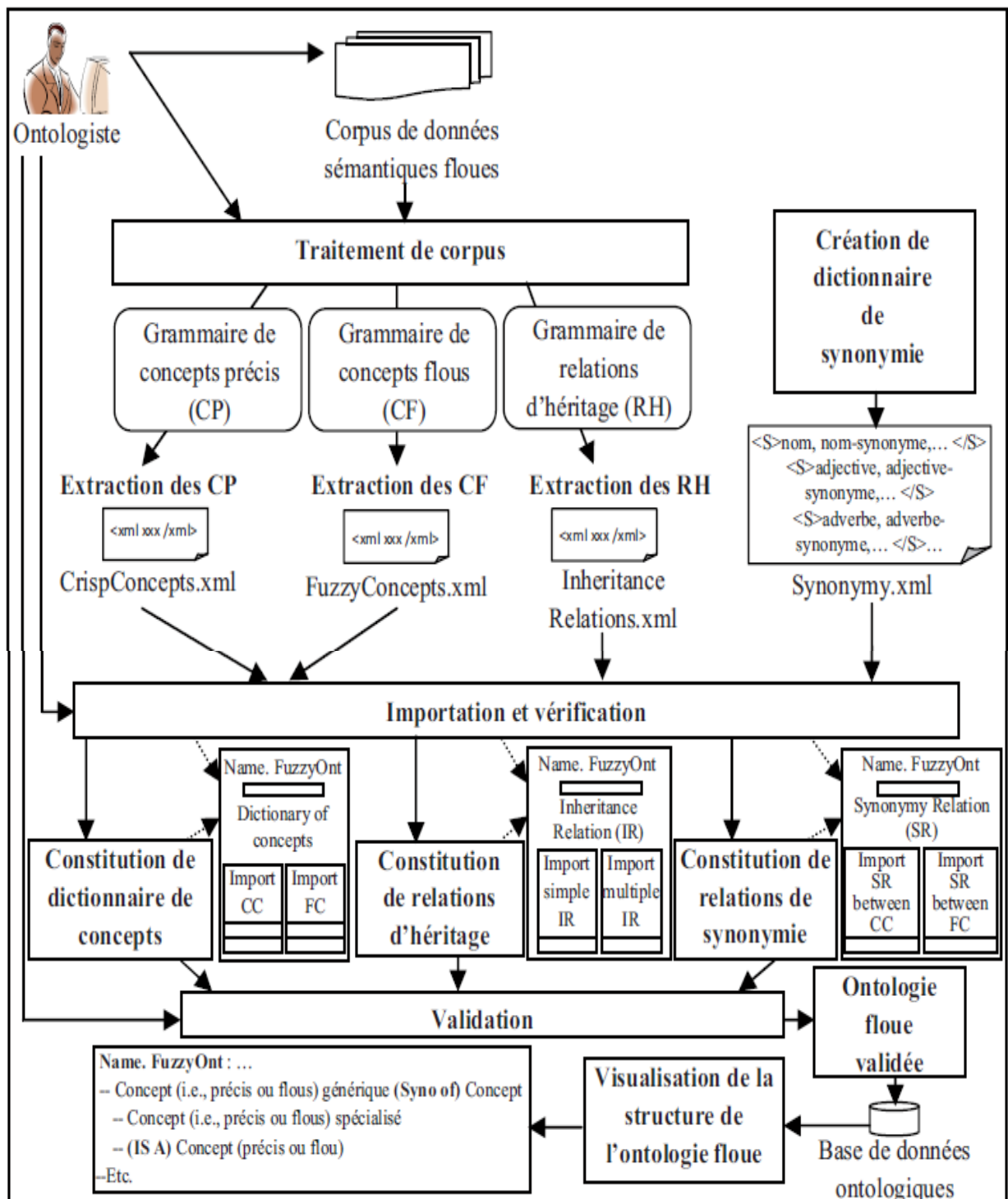
Exemple : Ahmed a comme Age 31 ans. On peut alors le considérer comme une « Personne Jeune » avec un degré d'appartenance de 0.8, calculé selon les formules de calcul suivant de la fonction d'appartenance trapézoïdale [35].

- $\text{Degré\_Appartenance} = (\text{Age} - 16) / (20 - 16)$  si  $\text{Age} \in [16 ; 20[$ .
- $\text{Degré\_Appartenance} = 1$  si  $\text{Age} \in [20 ; 30[$ .
- $\text{Degré\_Appartenance} = (35 - \text{Age}) / (35 - 30)$  si  $\text{Age} \in [30 ; 35]$ .
- $\text{Degré\_Appartenance} = 0$  ailleurs.

## 5.3 Approche de la construction des composants d'ontologies floues

Notre approche concerne la définition d'un dictionnaire des concepts précis et flous d'une ontologie floue à partir d'un corpus de données sémantiques floues. Elle concerne aussi la définition des relations d'héritage et de synonymie du fait que nous voulons définir une hiérarchie et une organisation entre les concepts. Elle nécessite l'utilisation d'outils pour assister l'ontologiste tout au long de la phase d'évaluation des besoins et de conceptualisation. Pour atteindre ces objectifs on doit suivre les cinq étapes suivantes [38] :

- 1- Constitution de corpus de données sémantiques floues.
- 2- Traitement automatique de corpus de données sémantiques floues.
- 3- Constitution de dictionnaires des concepts pour les ontologies floues.
- 4- Construction des relations d'héritage et de synonymie pour les ontologies floues.
- 5- Visualisation de la structure de l'ontologie floue.



**Figure 2.5** Approche de construction des composants d'ontologies floues [35]

## 6 Systèmes de raisonnement sur les ontologies

Les systèmes de raisonnement permettent la classification des nouveaux concepts intégrés dans l'ontologie. Ainsi de vérifier et corriger la classification dans une ontologie. Il existe plusieurs systèmes parmi eux nous citons [39] :

**RACER** (Renamed Abox and Concept Expression Résonner) : est un système implémentant une logique de description. Prise en compte des représentations au format *DAML+OIL*. RACER permet le test de **satisfiabilité** d'un concept (vérifier qu'un concept C admet des instances) le test de **subsumption** de concepts (vérifier qu'un concept C est subsumé par un concept D), et le test d'**instanciation** (vérifier qu'un individu a est instance d'un concept C, si seulement si à  $\hat{I} C$ ).

**JENA** : est une bibliothèque de classes Java qui facilite le développement d'applications pour le web sémantique. Elle permet de : manipulation de déclarations RDF, lecture et écriture RDF/XML, Notation, Stockage en mémoire ou sur disque de connaissances RDF, Langage d'interrogation d'une base RDF, Gestion d'ontologies : RDF-Sch.

### 6.1 Le raisonnement en logique floue

En logique classique, les raisonnements sont de la forme [39]:

$$\left\{ \begin{array}{l} \text{Si } p \text{ alors } q \\ P \text{ vrai alors } q \text{ vrai} \end{array} \right.$$

En logique floue, le raisonnement flou, également appelé raisonnement approximatif, se base sur des **règles floues** qui sont exprimées en langage naturel en utilisant les variables linguistiques dont nous avons donné la définition précédemment. Une règle floue aura cette forme: Si [prémisses] alors [conclusion].

En logique floue, les règles auront des formes telles que [37] :

$$\left\{ \begin{array}{l} \text{Si } x \in A \text{ et } y \in B \text{ alors } z \in C \\ \text{Si } x \in A \text{ alors } z \in C \\ \text{Si } x \in A \text{ ou } y \in B \text{ alors } z \in C \end{array} \right.$$

Avec A, B et C des ensembles flous.

**Exemple :** Si (la qualité de la nourriture est délicieuse) alors (le pourboire sera élevé).

Ce qui correspond à la forme  $\text{Si } x \in A \text{ alors } z \in C$



## 7 Moteurs d'inférences

La plupart des moteurs d'inférences existants sont conçus pour raisonner sur les logiques de descriptions, mais acceptent en entrée des fichiers OWL. Une fois l'ontologie chargée, ces moteurs effectuent les inférences sur la TBox et la ABox [28].

Moteur	RACER	Pellet	FaCT	FaCT++
DL	<i>SHIQ</i>	<i>SHIN(D)</i> <i>SHON</i>	<i>SHIQ</i> <i>SHF</i>	<i>SHIF</i>
Implantation	C++	Java	Common Lisp	C++
Inférence	TBox/ABox	TBox/ABox	TBox	TBox
API Java	Oui	Natif	Oui	Oui
OWL	OWL DL	OWL DL	OWL DL	OWL Lite
Décidabilité	Oui (OWL Lite)	Oui (OWL Lite)	Oui	Oui
Moteur	Surnia	Hoolet	F-OWL	
DL	Logique predicats	Logique predicats	<i>SHIQ</i>	
Implantation	Python	Java	Java	
Inférence	TBox/ABox	TBox/ABox	TBox/ABox	
API Java	Non	Oui	Oui	
OWL	OWL Full	OWL DL	OWL Full	
Décidabilité	Non	Non	Non	

**Table 2.6** Une comparaison des principaux moteurs d'inférence pour les LDs.

Pellet et Racer sont à l'heure actuelle les deux seuls moteurs d'inférence, permettant le raisonnement sur la ABox et la TBox et exploitent des ontologies possédant un niveau d'expressivité en logique de description et acceptent en entrée des fichiers OWL [25].

Le tableau 2.6, présenté ci-dessus, représente une comparaison des principaux moteurs d'inférences pour les LD.

## **8 Conclusion**

Les ontologies conçues comme réponse aux problèmes posés par l'intégration de connaissances au sein des systèmes informatiques, les ontologies apparaissent désormais comme une clé pour la manipulation automatique de l'information au niveau sémantique. Dans certains domaines d'applications, les connaissances terminologiques à représenter peuvent être parfois de nature vague et imprécise. De ce fait, les Logiques de descriptions floues ont été proposées en tant que langues de représentation et de raisonnement sur les connaissances imprécises et incertaines.

Dans le chapitre suivant nous allons entamer la partie de développement et de construction de l'ontologie floue.

## Table des matiere

1 Introduction.....	26
2 Ontologies et modèles de connaissance .....	26
3 La représentation de connaissances à l'aide des logiques de descriptions .....	27
3.1 Famille des langages LD .....	28
3.1.1 La syntaxe d'AL.....	28
<b>3.1.2 La sémantique formelle d'AL.....</b>	<b>29</b>
3.2 La T-box et la A-box.....	31
<b>3.2.1 Le niveau terminologique T-box.....</b>	<b>31</b>
<b>3.2.2 Le niveau assertionnel A-box .....</b>	<b>31</b>
<b>3.2.3 Un exemple d'une base de connaissances.....</b>	<b>31</b>
3.3 L'inférence dans les logiques de description .....	32
4 La logique floue .....	33
4.1 Définition .....	33
4.2 Les logiques de descriptions floues.....	34
4.3 La sémantique des LDs floues .....	34
4.4 Des logiques de descriptions floues .....	37
5 Ontologies floues .....	37
5.1 Définition .....	37
5.2 Les composants d'ontologies floues.....	37
<b>5.2.1 Les concepts flous.....</b>	<b>38</b>
<b>5.2.2 Les relations floues .....</b>	<b>38</b>
<b>5.2.3 Les instances .....</b>	<b>39</b>
<b>5.2.4 Les axiomes .....</b>	<b>39</b>
5.3 Approche de la construction des composants d'ontologies floues.....	39
6 Systèmes de raisonnement sur les ontologies.....	41
6.1 Le raisonnement en logique floue .....	41
7 Moteurs d'inférences.....	42
8 Conclusion.....	43

## Les Figure

Figure 2.1 Logique floue vs logique booléenne .....	34
Figure 2.2 Comparaison d'un ensemble classique et d'un ensemble flou. ....	35
Figure 2.3 Exemple de concepts flous .....	38
Figure 2.4 Exemple de relations floues .....	38
Figure 2.5 Approche de construction des composants d'ontologies floues. ....	40

## Les Tableaux

Table 2.1 Syntaxe de la logique AL.....	28
Table 2.2 Sémantique de la logique AL.....	29
Table 2.3 Exemple de constructeurs de rôles et concepts pour étendre AL.....	30
Table 2.4 Base de connaissances composée d'une T-box et d'une A-box.....	32
Table 2.5 équations de l'interprétation standard.....	36
Table 2.6 Une comparaison des principaux moteurs d'inférence pour les LDs.....	42