

### Partie IV : Formalisation et opérationnalisation

#### 1. Introduction

Les méthodes formelles jouent un rôle crucial dans le développement des technologies du Web sémantique et ont pour but d'assurer leur fiabilité et leur sécurité. Les techniques de modélisation et de vérification peuvent être utiles dans les différents niveaux de la conception et du déploiement des ontologies (Chaâbani & al, 2009). Parmi ces méthodes les logiques de description en sont un outil très puissant par la variété de ses différents langages.

Dans une ontologie formalisée, nous pouvons vérifier la consistance, et calculer la hiérarchie des classes, elle permet aussi de compléter et valider le modèle construit.

A notre connaissance, c'est la première fois qu'une ontologie en arabe est formalisée avec la logique de description pour que, une fois terminée elle puisse être intégrée dans n'importe quelle application et qu'on puisse raisonner dessus. La formalisation peut bien être faite avec les frames<sup>53</sup>, les graphes conceptuels<sup>54</sup>, le formalisme-Z<sup>55</sup>, LIFE<sup>56</sup> ou une logique de description<sup>57</sup>.

#### 2. Formalisation des concepts

##### 2.1. Les logiques de description

Les logiques de descriptions sont une famille de formalisme pour la représentation des connaissances dans différents domaines notamment dans les ontologies. Dans une base de connaissance en logique descriptive, on distingue la TBox (niveau Terminologique) et la ABox (niveau Assertionnel). La première contient tous les axiomes définissant les concepts du domaine, comme la définition de « رسول » qui est un « نبي » qui a en plus « *a un message de Dieu à transmettre à un peuple* » par exemple. La ABox contient les assertions sur les individus en spécifiant leurs classes et leurs attributs. C'est dans la ABox qu'on trouvera que « عيسى » est un « نبي » qui en plus a un message à un peuple « بنو اسرائيل ». Dans la TBox on est

<sup>53</sup> <http://www.learningwebdesign.com/pdf/frames.pdf>

<sup>54</sup> [http://www.jfsowa.com/cg/cg\\_hbook.pdf](http://www.jfsowa.com/cg/cg_hbook.pdf)

<sup>55</sup> <http://www.ppig.org/papers/14th-triffitt.pdf>

<sup>56</sup> <http://www.cs.uiowa.edu/~fleck/lifeIntro.pdf>

<sup>57</sup> <http://www.cs.man.ac.uk/~horrocks/Publications>

intéressé à savoir si tous les concepts sont consistants, par exemple que si deux classes « مؤمن » et « كافر » sont disjointes, on ne doit trouver une sous-classe communes aux deux. C'est aussi dans la TBox qu'on exprime la relation de subsomption, par exemple si on a que « نبي » *est\_un* « إنسان » et qu'on a que « رسول » *est\_un* « نبي » alors on déduit automatiquement que « رسول » est un « إنسان ». Toutes fois il existe pour notre ontologie des inconsistances avec lesquelles on doit travailler comme tout « إنسان » descend de « ذكر » et « أنثى » et que « عيسى » est un « إنسان » mais ne descend pas de « ذكر » et « أنثى ». Outre cela, nous avons été appelés à compléter la hiérarchie avec des termes générique quand cela s'est avéré nécessaire.

### 2.2. AL : La base des logiques de description

Les logiques de description varient, de La base qui est le langage AL (attributive language), jusqu'à celles avec une complexité exponentielle comme c'est le cas pour SHIF or SHIQ (Papini, 2002).

Le degré d'expressivité du langage AL est limité, mais il peut convenir à une utilisation qui ne nécessite pas un haut degré d'expressivité. Les descriptions possibles dans le langage AL sont les suivantes (notons que les concepts ou rôles atomiques ou primitifs, constituent les entités élémentaires d'une TBox tels que « نبي » et « رسول », alors que les concepts et les rôles composés ou définis sont ceux combinés au moyen de constructeurs tels que « رسول  $\cap$  نبي » (Napoli, 1997)).

#### 2.2.1. Syntaxe du langage AL

En supposant que A est un concept atomique et que C et D peuvent être atomiques ou complexes (Gagnon, 2004) nous avons:

A	Concept atomique
T	Concept universel
$\perp$	Concept impossible
$\neg A$	Négation atomique

$C \cap D$	Intersection de concepts
$\forall R.C$	Restriction de valeur
$\exists R. T$	Quantification existentielle limitée

**Tableau 15:** La syntaxe du langage AL

Le concept le plus générique est la racine désignée par T.  $\perp$  est le concept le plus spécifique, le constructeur *et*  $\cap$  définit une conjonction et le symbole  $\neg$  exprime une négation. Le quantificateur universel *tous* ( $\forall r.C$ ) donne le co-domaine du rôle r, alors que le quantificateur existentiel *quelque* ( $\exists r$ ) exprime le fait qu'il y a au moins un couple d'individus reliés par la relation (ou rôle) r.

### 2.2.2. Sémantique du langage AL

La sémantique du langage AL fait appel à la théorie des ensembles. A chaque concept est associé un ensemble d'individus. Une interprétation suppose l'existence d'un ensemble non vide  $\Delta$  qui représente des entités du monde décrit. Soit une fonction d'interprétation  $I$ , qui associe à chaque description un sous-ensemble de  $\Delta$ . On suppose que pour chaque concept atomique A, la fonction  $I(A)$  associe un sous-ensemble  $A^I \subseteq \Delta$ , et pour chaque relation atomique R, une relation binaire  $R^I \subseteq \Delta \times \Delta$ . La fonction d'interprétation est définie ainsi :

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

**Tableau 16:** Sémantique du langage AL

De plus on définit le concept d'axiome terminologique qui est en fait toute formule de la forme suivante :

$$C \subseteq D \text{ ou } C \equiv D$$

La première forme déclare que toute entité de la classe C appartient aussi à la classe D, alors que la seconde indique que les concepts C et D sont équivalents, c'est-à-dire que si un individu b appartient à la classe C, il appartient nécessairement à la classe D et vice versa.

Leur sémantique est :

$$I(C \subseteq D) = \text{vrai si } I(C) \subseteq I(D)$$

$$I(C \equiv D) = \text{vrai si } I(C) = I(D)$$

Une définition est un axiome de la forme  $C \equiv D$  où C est un concept atomique. Elle sert à associer un nom à un concept complexe.

Nous pouvons bien sûr définir d'autres constructeurs pour obtenir d'autres langages tels que:

ALU =  $AL \cup (C \cup D)$ : disjonction (Union).

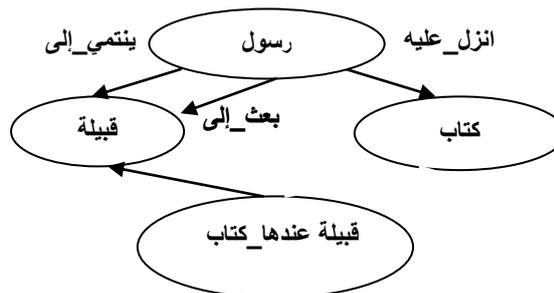
ALC =  $AL \cup \{\neg c\}$  c est un concept défini ou combiné. ALC (AL avec le Complément) est le langage le plus important.

### 2.2.3. Les deux niveaux de description

Dans ce qui suit nous allons présenter la formalisation des concepts et relations Avec le langage AL. Comme nous l'avons mentionné plus haut, en logique des descriptions il existe deux niveaux: le terminologique et l'assertionnel.

#### 2.2.3.1. Le niveau terminologique ou TBox

La TBox décrit la connaissance générale d'un domaine particulier, elle inclut les définitions des concepts et des rôles et contient le modèle du monde en termes de concepts leurs propriétés et les relations entre les concepts.



**Figure 40:** Représentation des concepts et relations (Zaidi & al., 2012b)

Nous pouvons exprimer ces concepts et ces relations ainsi:

- قبيلة . ينتمي-إلى رسول  $\subseteq \forall$  (un messager appartient seulement à une tribu)
- قبيلة  $\subseteq$  قبيلة\_عندها\_كتاب (une tribu\_avec\_un\_livre est une tribu )
- قبيلة . بعث\_إلى  $\subseteq \exists$  نبي (un prophete est envoyé à au moins une tribu )

Nous avons notés le type de relations suivantes:

- L'identité: (كتاب، كتاب)
- La synonymie : (مكة، بكة), (إسرائيل، يعقوب), (جبريل، روح\_القدس), (العرش، الكرسي)
- La classification (الرسول، النبي)
- L'antonymie : (ذكر، أنثى), (مؤمن، كافر)
- L'équivalence : (نعمة، فضل\_من\_الله)

Et quelques propriétés comme la réflexivité dans :

- قوم، قوم) لا يسخر من
- مؤمن، مؤمن) أخ

### 2.2.3.2. Le niveau assertionnel (factuel) ou ABox

Il décrit les individus en les nommant et en spécifiant les assertions, en termes de concepts et de rôles. Plusieurs ABox peuvent être associées à une même TBox. Chacune d'elles montre une configuration constituée par des individus et utilise les concepts et les rôles de la TBox pour l'exprimer.

Considérons l'exemple suivant, tiré de notre corpus:

Concept1	Relation	Concept2
رسول	أنزل_عليه	كتاب

## Chapitre 4 : Le système proposé

Avec « عيسى » comme instance du concept « رسول » et « الإنجيل » comme instance du concept « كتاب » et la relation « انزل\_عليه ». La TBox et la ABox correspondantes sont:

<p>T-BOX</p> <p>string . اسمه <math>\exists \cap</math> انسان <math>\doteq</math> رسول</p> <p>string . قبيلة <math>\exists \cap</math></p> <p>كتاب . انزل عليه <math>\exists \cap</math></p> <p>قبيلة . بعث_إلى <math>\exists \subseteq</math></p>
<p>ABOX</p> <p>عيسى : رسول</p> <p>الإنجيل : كتاب</p> <p>(بنو_إسرائيل ، عيسى) بعث_إلى</p> <p>(الإنجيل ، عيسى) انزل_عليه</p>

**Figure 41:** Exemple de TBox et de ABox associée

Soit l'interprétation suivante:

$\Delta = \{ \text{محمد، عيسى، موسى، إبراهيم، داوود، شعيب، صالح، القرآن، الإنجيل، التوراة، الزابور، الصحف،} \\ \text{بنو_إسرائيل، مدين، قريش، ثمود،العراق،}$

$\mathcal{I}^{\text{نبي}} = \{ \text{محمد، عيسى، موسى، إبراهيم، داوود، شعيب، صالح} \}$

$\mathcal{I}^{\text{رسول}} = \{ \text{محمد، عيسى، موسى، إبراهيم، داوود} \}$

$\mathcal{I}^{\text{قبيلة}} = \{ \text{بنو_إسرائيل، مدين، قريش، ثمود،العراق} \}$

$\mathcal{I}^{\text{انزل_عليه}} = \{ \text{إبراهيم، الصحف،} \text{داوود، الزابور،} \text{موسى، التوراة،} \text{عيسى، الإنجيل،} \text{محمد، القرآن} \}$

$\mathcal{I}^{\text{ينتمي_إلى}} = \{ \text{بنو_إسرائيل، داوود،} \text{بنو_إسرائيل، موسى،} \text{بنو_إسرائيل، عيسى،} \text{قريش، محمد} \}$

$\mathcal{I}^{\text{بعث_إلى}} = \{ \text{ثمود، صالح} \}$  (العراق، إبراهيم)، (مدين، شعيب

Alors :

$\overline{\mathcal{I}^{\text{رسول}} \cap \mathcal{I}^{\text{نبي}}} = \{ \text{شعيب، صالح} \}$

$\overline{\mathcal{I}^{\text{نبي}}} = \{ \text{القرآن، الإنجيل، التوراة، الزابور، الصحف، بنو_إسرائيل، مدين، قريش، ثمود،العراق} \}$

$\overline{\mathcal{I}^{\text{قبيلة.بعث_إلى}}} = \{ \text{قريش، محمد} \}$

(مدین),(بنو اسرائیل ،داوود) ,( بنو اسرائیل ،موسی )،(بنو اسرائیل ،عیسی ) = { قبيلة .بعث\_إلى(∀

{ (ثمود ،صالح) ( العراق ،إبراهيم) ،(شعيب

Pour formaliser les concepts, un éditeur a été spécialement développé parce que tous les éditeurs existants ne pouvaient supporter les caractères arabes avec les symboles mathématiques requis par la logique de descriptions.

Le nouvel éditeur a une syntaxe spécifique qui respecte la grammaire de la LD. Il contient les deux parties Terminologique (TBox) et factuelle (ABox) et il permet d'importer un fichier LD pour le réutiliser (Zitouni, 2010).

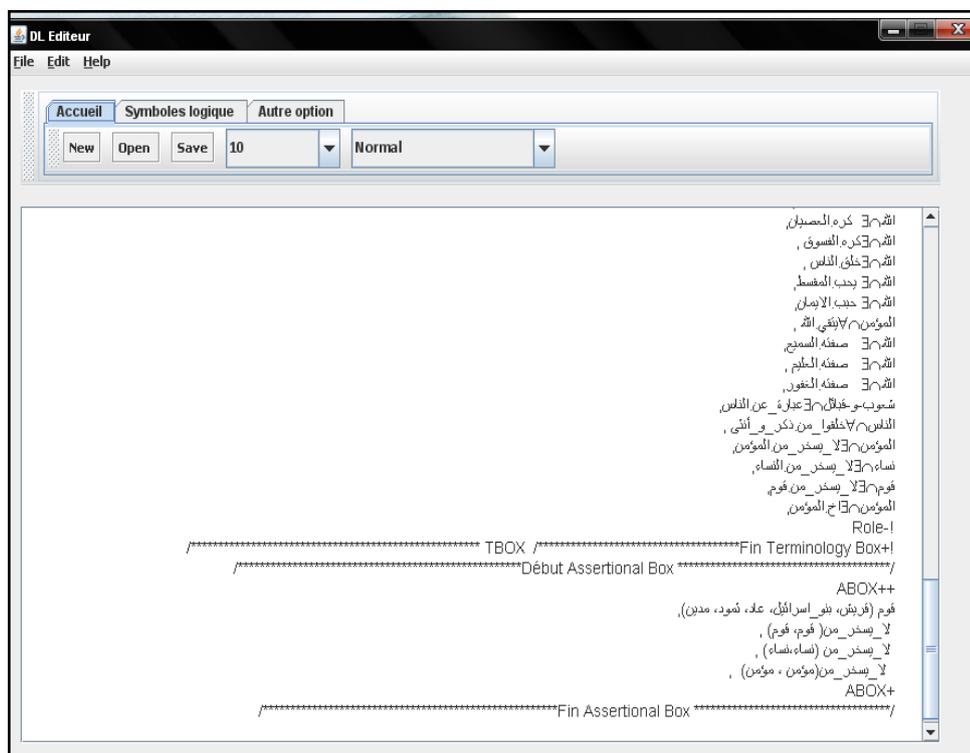
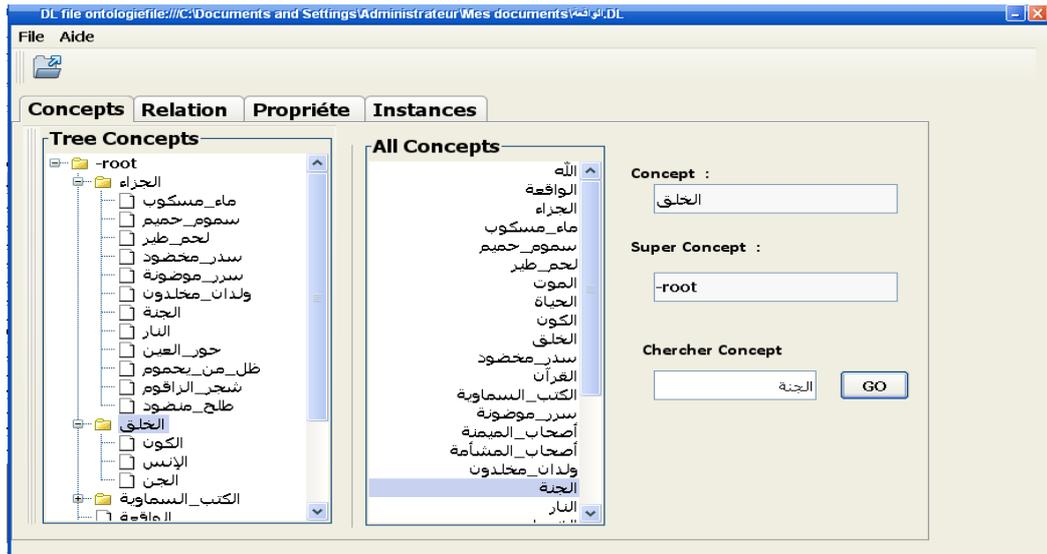


Figure 42: Editeur pour la manipulation des caractères arabes et des symboles mathématiques

### 3. Opérationnalisation de l'ontologie

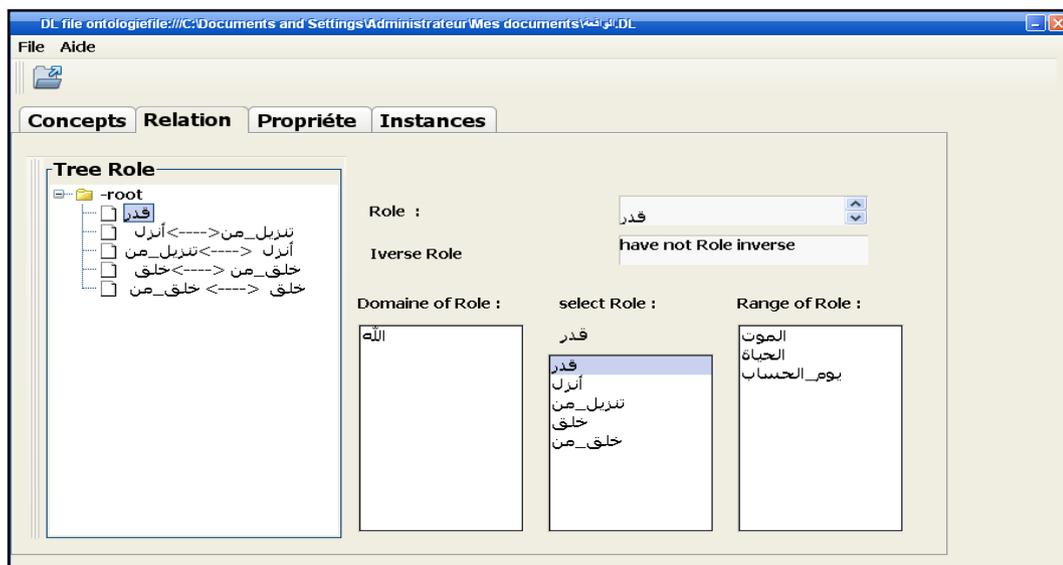
Pour la phase d'opérationnalisation et afin de tester notre fichier LD obtenu, nous avons essayé de l'utiliser dans une application (Djabourabi, 2011), pour montrer que notre ontologie, une fois complétée et formalisée, peut être intégrée dans n'importe quel système dans le but d'exploiter le fichier LD. Pour cela une application simple a été créée dans le but d'opérationnaliser l'ontologie cette application prend en entrée un fichier LD qui doit représenter l'ontologie

formalisée et produit en sortie la visualisation de la hiérarchie des concepts, les relations, les propriétés et les instances et qui permet de rechercher un concept dans l'arbre conceptuel ou mettre à jour la structure ontologique par l'ajout de nouveaux concepts, la modification ou la suppression et ce soit directement de l'interface de l'application ou en mettant à jour le fichier LD à l'aide de l'éditeur conçu à cette fin.



**Figure 43:** Interface utilisateur du navigateur de l'ontologie

Cette application n'est qu'un essai, l'ontologie peut être intégrée dans d'autres systèmes comme l'expansion d'une requête en rajoutant des hyperonymes afin



**Figure 44:** Visualisation des relations

d'améliorer le rappel ou des hyponymes pour améliorer la précision. Elle serait également d'une aide considérable pour des applications de brainstorming dans le sens où elle fournit à l'utilisateur des termes en relation avec un éventuel projet.

### **4. Conclusion**

Nous avons présenté dans ce chapitre l'essentiel de notre travail qui se résume en quatre parties, la première nous l'avons dédiée à l'extraction des termes simples à l'aide d'une approche statistique basée sur tf-idf, notons qu'on aurait pu utiliser GATE pour retrouver des noms en tant que termes et nous pouvons filtrer les résultats avec tf-idf. La seconde partie concerne la recherche des collocations avec une méthode hybride d'abord linguistique puis les résultats obtenus sont filtrés avec une méthode statistique utilisant l'information mutuelle comme mesure de cohésion entre les termes. La troisième traite de l'extraction de relations sémantiques entre termes simples ou collocations préalablement retrouvés, elle repose aussi sur la succession des deux mêmes approches mentionnées. Dans la dernière partie nous avons présenté la formalisation des concepts avec l'un des formalismes de représentation de connaissances les plus puissants, qui est la logique de description. Enfin nous avons contribué à la conception d'une application pour expérimenter l'utilisation du fichier LD créé avec un éditeur développé à cette fin, puisque les éditeurs existants ne permettaient pas la manipulation des caractères arabes avec les symboles mathématiques utilisés en logique de description. L'objectif étant de rendre l'ontologie construite opérationnelle, cette application permet la recherche, la visualisation et la mise à jour des concepts.