

République Algérienne Démocratique Et Populaire  
Ministère De L'enseignement Supérieur Et De Recherche Scientifique  
Université Abderrahmane Mira - Béjaïa -



Faculté Des Sciences Exactes  
Département D'informatique

*Mémoire de :*

**MASTER PROFESSIONNEL**

*Option :*

**GÉNIE LOGICIEL**

- *Thème* -

---

# Construction d'ontologies à l'aide de significations unifiées et liées pour le Web sémantique

---

*Présenté par :*

**BAZIZI Rabah**

*Le 20/09/2020 devant le jury composé de :*

|                    |                           |         |
|--------------------|---------------------------|---------|
| <b>Examineur 1</b> | Dr. AIT ABDELOUHAB Karima | - MCB - |
| <b>Examineur 2</b> | Dr. ZIDANI Ferroudja      | - MCB - |
| <b>Encadreur</b>   | Dr. EL BOUHISSI Houda     | - MCA - |

*- Promotion -*

**2019/2020**

---

# *Remerciements*

.  
. *Nous*  
. *Voudrions*  
. *Remercier*  
. *Tous*  
. *Ceux*  
. *Qui*  
. *Ont*  
. *Contribué*  
. *Pour*  
. *Pouvoir*  
. *Mener*  
. *Ce*  
. *Travail*  
. *A*  
. *Terme*  
. *Les parents*  
. *Les proches*  
. *Les amis*  
. *Les enseignants*  
. *Et les collègues.*



# Table Des Matières

|   |            |
|---|------------|
| <b>Table des matières</b>   | <b>i</b>   |
| <b>Liste des figures</b>  | <b>iv</b>  |
| <b>Liste des tableaux</b>   | <b>vi</b>  |
| <b>Liste des abréviations</b>   | <b>vii</b> |
| <b>Introduction générale</b>  | <b>1</b>   |
| <b>1 Fondement du Web sémantique</b>                                      | <b>4</b>   |
| 1.1 Introduction . . . . .  | 4          |
| 1.2 Web sémantique. . . . .   | 6          |
| 1.2.1 Définition . . . . .  | 7          |
| 1.3 Concept d'ontologie. . . . .  | 8          |
| 1.3.1 Définition d'ontologie . . . . .                                    | 9          |
| 1.3.2 Historique des ontologies en informatique . . . . .                 | 11         |
| 1.3.3 Conception d'ontologie. . . . .                                     | 12         |
| 1.3.3.1 Les composants d'ontologie . . . . .                              | 12         |
| 1.3.3.2 cycle de vie d'une ontologie . . . . .                            | 14         |
| 1.3.3.3 Classification d'ontologies . . . . .                             | 17         |
| 1.3.3.4 Critères de conception d'une ontologie . . . . .                  | 18         |
| 1.3.4 Les outils de construction d'ontologie . . . . .                    | 19         |
| 1.3.4.1 Les outils dépendants de formalisme de représentation . . . . .   | 20         |
| 1.3.4.2 Les outils indépendants de formalisme de représentation . . . . . | 20         |
| 1.3.5 Apports d'ontologie . . . . .                                       | 21         |

|          |  |           |
|----------|--|-----------|
| 1.4      | Architecture du web sémantique . . . . .                         | 22        |
| 1.5      | Technologies utilisées en Web sémantique . . . . .               | 23        |
| 1.5.1    | Les identificateurs de ressources uniformes . . . . .            | 23        |
| 1.5.2    | Les métadonnées . . . . .  | 24        |
| 1.5.3    | Extensible Markup Language . . . . .                             | 25        |
| 1.5.4    | Resource Description Framework . . . . .                         | 25        |
| 1.5.5    | Resource Description Framework Schema . . . . .                  | 27        |
| 1.5.6    | Web Ontology Language . . . . .                                  | 28        |
| 1.5.6.1  | Web Ontology Language LITE . . . . .                             | 28        |
| 1.5.6.2  | Web Ontology Language LD . . . . .                               | 29        |
| 1.5.6.3  | Web Ontology Language FULL . . . . .                             | 29        |
| 1.5.7    | Rule Interchange Format . . . . .                                | 29        |
| 1.5.8    | SPARQL Protocol And Resource Query Language . . . . .            | 30        |
| 1.5.9    | Apache Jena . . . . .  | 30        |
| 1.6      | Apports du Web sémantique . . . . .                              | 30        |
| 1.7      | Conclusion . . . . .   | 31        |
| <b>2</b> | <b>Alignement des ontologies</b>                                 | <b>32</b> |
| 2.1      | Introduction . . . . .   | 32        |
| 2.2      | Les sources d'hétérogénéités dans les ontologies . . . . .       | 33        |
| 2.2.1    | Le niveau syntaxique . . . . .                                   | 33        |
| 2.2.2    | Le niveau conceptuel . . . . .                                   | 33        |
| 2.2.3    | Le niveau terminologique . . . . .                               | 35        |
| 2.2.4    | Le niveau sémantique ou pragmatique . . . . .                    | 35        |
| 2.3      | Définition de l'alignement d'ontologies . . . . .                | 35        |
| 2.4      | Les méthodes d'alignement . . . . .                              | 36        |
| 2.4.1    | Les méthodes locales . . . . .                                   | 37        |
| 2.4.1.1  | Méthodes à base de la comparaison lexicale . . . . .             | 37        |
| 2.4.1.2  | Méthodes à base de traitement des langages naturels . . . . .    | 37        |
| 2.4.1.3  | Méthode à base de comparaison structurelle des entités . . . . . | 37        |
| 2.4.1.4  | Méthodes extensionnelles . . . . .                               | 38        |
| 2.4.2    | Les méthodes globales . . . . .                                  | 38        |
| 2.4.3    | WordNet . . . . .  | 39        |
| 2.5      | Opérations sur les ontologies . . . . .                          | 40        |
| 2.6      | Conclusion . . . . .   | 40        |

|  |               |
|--|---------------|
| <b>3 Conception et implémentation</b>                | <b>41</b>     |
| 3.1 Introduction . . . . .                           | 41            |
| 3.2 Approche proposée . . . . .                      | 42            |
| 3.3 Conception de l'application . . . . .            | 44            |
| 3.3.1 Conception d'un dictionnaire unifié . . . . .  | 45            |
| 3.3.2 Création des ontologies . . . . .              | 47            |
| 3.3.3 Fonctionnalités de l'application . . . . .     | 49            |
| 3.3.4 Architecture de l'application . . . . .        | 50            |
| 3.4 Implémentation et réalisation . . . . .          | 51            |
| 3.4.1 Les outils utilisés . . . . .                  | 51            |
| 3.4.1.1 Prototégé 5.5.0. et le langage OWL . . . . . | 51            |
| 3.4.1.2 Langage OWL . . . . .                        | 52            |
| 3.4.1.3 Netbeans 8.0.2 . . . . .                     | 52            |
| 3.4.1.4 Langage JAVA . . . . .                       | 52            |
| 3.4.1.5 Langage SPARQL . . . . .                     | 52            |
| 3.4.1.6 Langage de règles d'inférence Jena . . . . . | 52            |
| 3.4.2 Interfaces et fonctionnalités . . . . .        | 53            |
| 3.5 Conclusion . . . . .                             | 60            |
| <br><b>Conclusion Générale</b>                       | <br><b>61</b> |
| <br><b>Références</b>                                | <br><b>63</b> |

# Liste Des Figures

|    |   |    |
|----|---|----|
| 1  | Partie d'une ontologie du monde des "Animaux". . . . .                                | 10 |
| 2  | Cycle de vie d'ontologie [6]. . . . .   | 13 |
| 3  | L'architecture « Layer cake » proposée par Tim Berners Lee [12]. . . . .              | 22 |
| 4  | Données échangées sans utilisation de métadonnées. [16] . . . . .                     | 23 |
| 5  | Données échangées avec utilisation de métadonnées. [16] . . . . .                     | 24 |
| 6  | Une simple déclaration représentée par un RDF graphique [18]. . . . .                 | 25 |
| 7  | Un graphique RDF décrivant la personne « Eric Miller » [18]. . . . .                  | 25 |
| 8  | RDF / XML décrivant « Eric Miller » [18]. . . . .                                     | 26 |
| 9  | Un exemple schématique de RDFS [20]. . . . .  | 26 |
| 10 | Les trois dimensions de l'hétérogénéité au niveau conceptuel [29]. . . . .            | 32 |
| 11 | Exemple d'alignement sémantique [31]. . . . .   | 34 |
| 12 | Schémat montrant comment choisir un mot spécifique à un sens unifié. . . . .          | 42 |
| 13 | Hierarchie des concepts d'ontologie "FamilleFrançaise.owl". . . . .                   | 45 |
| 14 | Hierarchie des concepts d'ontologie "FamilleAllemande.owl". . . . .                   | 46 |
| 15 | Fonctionnalités de l'application. . . . .   | 47 |
| 16 | L'architecture de l'application. . . . .  | 48 |
| 17 | Règles d'inférences programmées pour l'application. . . . .                           | 50 |
| 18 | Interface "Affichage". . . . .  | 51 |
| 19 | Une requête SPARQL et son résultat obtenu en allemand. . . . .                        | 52 |
| 20 | Une requête SPARQL et son résultat obtenu en anglais. . . . .                         | 52 |
| 21 | Une requête SPARQL et son résultat obtenu en français. . . . .                        | 52 |
| 22 | Règles d'inférence ajoutées pour représenter l'entité "Enfant". . . . .               | 53 |
| 23 | La requête SPARQL pour afficher les enfants ainsi que son résultat. . . . .           | 53 |
| 24 | Interface "Opérations". . . . .   | 54 |
| 25 | Hierarchie des concepts des deux ontologies obtenue avec l'opération "Union". . . . . | 55 |
| 26 | Les requêtes SPARQL pour afficher les hommes et les masculins. . . . .                | 56 |

# Liste Des Tableaux

|   |  |    |
|---|--|----|
| 1 | L'ensemble des entités en langues différentes avec leurs identifiants. . . . . | 44 |
| 2 | Les instances de l'ontologie "FamilleFrançaise.owl". . . . .                   | 45 |
| 3 | Les instances de l'ontologie "FamilleAllemande.owl". . . . .                   | 46 |

# Liste Des Abréviations

|                |  |
|----------------|--|
| <i>DCMI</i>    | Dublin Core Metadata Initiative,                                     |
| <i>HTML</i>    | Hypertext Markup Language,   |
| <i>IAS</i>     | Intelligence Artificielle Symbolique,                                |
| <i>IFLA</i>    | International Federation of Library Associations and Institutions,   |
| <i>INRIA</i>   | Institut National de Recherche en Informatique et en Automatique,    |
| <i>ISO</i>     | International Organization for Standardization,                      |
| <i>OWL</i>     | Web Ontology Language,   |
| <i>PC</i>      | Personel Compuetr,   |
| <i>RIF</i>     | Rule Interchange Format,   |
| <i>RDF</i>     | Resource Description Framework,                                      |
| <i>RDFS</i>    | Resource Description Framework Shemat,                               |
| <i>SGML</i>    | Standard Generalized Markup Language,                                |
| <i>SKOS</i>    | Simple Knowledge Organization System,                                |
| <i>SPARQL</i>  | SPARQL Protocol and RDF Query Language,                              |
| <i>URC</i>     | Unified Resource Classification,                                     |
| <i>URI</i>     | Uniform Resource Identifier,   |
| <i>URL</i>     | Unified Resource Locator,  |
| <i>URN</i>     | Unified Resource Name,   |
| <i>W3C</i>     | World Wide Web Consortium,   |
| <i>WIMMICS</i> | Web-Instrumented Man-Machine Interactions, Communities and Semantics |

# Introduction générale

Les progrès considérables des technologies informatiques ont conduit à l'émergence de nombreux programmes et applications dans divers domaines (la médecine, l'économie, les médias, les réseaux sociaux, ...) faisant que l'Internet accumule une quantité énorme de ressources (pages Web, données, applications, connaissances, programmes, ...) qui doivent être interrogées, traitées et partagées quotidiennement. Généralement, le but est de tirer les informations pertinentes dans cette gigantesque quantité d'information dont les utilisateurs ont besoin.

Plusieurs actions ont été prises à cet égard pour fournir un cadre commun afin de partager et réutiliser les données, malgré les considérations de différences de compatibilité, d'interopérabilité et de performances qui existent entre les applications, les entreprises et les communautés. Ces efforts ont encouragé le développement du Web actuel pour le transformer en une toile de données appelé le "Web sémantique". Il s'agit d'un domaine issu de la collaboration étroite des dirigeants du W3C et de la participation d'un grand nombre de chercheurs et de partenaires de l'industrie, sur la description des ressources (RDF).

Ces travaux ont mené à la réalisation d'un réseau de données, où les informations prennent la description nécessaire qui donne aux machines la possibilité de mieux exploiter et comprendre ces ressources. Cela est possible grâce au concept d'ontologie comme moyen de définition des accords de contenu pour partager et réutiliser les connaissances entre les entités logicielles.

Cette nouvelle approche d'utilisation d'ontologies s'est avérée fructueuse, cependant l'existence de multitude ontologies qui sont, pour les mêmes connaissances, hétérogènes et distribuées dans le Web, est en soi une source d'ambiguïté et un problème qui freine l'interopérabilité complète entre machines. D'où la nécessité de développer des méthodes de réconciliation entre les ontologies, techniquement appelé alignement d'ontologies.

**- Problématique :**

Les procédés algorithmiques d'alignement semblent la solution idéale afin de remédier à la multiplicité des ontologies, et malgré le progrès bien connu dans le développement de ces méthodes effectives cependant divers facteurs importants, qui sont parfois conflictuels, justifient l'hétérogénéité d'utilisation universelle d'ontologies. En plus de la multiplicité, l'existence de plusieurs formalismes de représentations des ontologies et l'utilisation de langages de description différents (KIF, RDF, SHOE, OWL, etc.) sont également l'un de ces facteurs qui touchent à l'interopérabilité.

Mais encore, le multilinguisme (les données sont décrites dans des vocabulaires de plusieurs langages naturels) est une source d'hétérogénéité majeure et cela soulève des problèmes inhérents au traitement du langage naturel (tels que les synonymes, les acronymes, les homonymes, les hyponymes, etc.) nécessitant des procédures de traduction très fines ; vu que le choix d'un langage naturel unique sort du cadre technique et suscite des questions plutôt éthiques.

Ainsi, pour obtenir une interprétation similaire d'une ontologie et parvenir à la même sémantique de ses données par les nombreux utilisateurs et machines différentes, il faut réduire considérablement ces sources d'hétérogénéité.

**- Contributions :**

Dans ce mémoire, nous avons proposé une méthode pour réduire grandement et à tous les niveaux ces sources d'hétérogénéité au niveau des concepts, pour cela nous avons utilisé un mécanisme standardisé pour remplir les ontologies avec les données dès le début de leur conception. Ceci, en concevant un dictionnaire unifié des significations de tous les concepts existants, auquel il est fait référence lorsque l'on commence à remplir les ontologies avec les données. Après, développer un intermédiaire qui nous permette le traitement de ces ontologies selon ce dictionnaire unifié.

# Fondement du Web sémantique

## 1.1) Introduction

## 1.2) Web sémantique

## 1.3) Concept d'ontologie

## 1.4) Architecture du web sémantique

## 1.5) Technologies utilisées en Web sémantique

## 1.6) Apports du Web sémantique

## 1.7) Conclusion

### 1.1) Introduction

Ce chapitre constitue les généralités du Web sémantique, alors nous avons défini ce qui est un Web sémantique. Puis dans une prochaine section, nous avons introduit la notion d'ontologie où nous avons éclairé sa définition et comment concevoir une ontologie, qu'elles sont ses composants, aussi nous avons montré qu'elle existe en différents types par classes, ainsi les technologies de création et de représentation d'ontologies, avant de parler sur leur apport et de ce qu'elles nous permettent de réaliser. Ensuite, nous avons parlé de l'architecture sur laquelle le Web sémantique est basé, suivi d'une présentation des différents outils permettant de réaliser des applications conçues pour cette architecture. En fin nous avons terminé le chapitre par une conclusion.

## 1.2) Web sémantique

De plus en plus, les machines (les moteurs de recherche, les agents, les PC, les applications) ont besoin davantage de détails sur les données qu'elles reçoivent, désigner qu'un service affiché à l'écran est le plus approprié dans sa catégorie, ou bien un agent logiciel traitant une page Web textuelle, par exemple publiée sur un article économique, peut extraire des statistiques comme informations utiles<sup>1</sup> sont des exemples et bien d'autres qui nous poussent à réfléchir sur le formalisme des connaissances stockées et échangées sur Web. La grande masse des informations contenues sur Internet et leur représentation insuffisante font que la recherche des bonnes choses constitue un véritable travail pour les êtres humains. Le défaut majeur de HTML est qu'il convient bien à la consommation humaine, mais pas à la capacité de traitement par machine[1].

L'évolution des technologies d'analyse du langage, telles que la lemmatisation<sup>2</sup>, l'extraction d'entités nommées<sup>3</sup>, la classification et le clustering dans les corpus de pages web permettent d'améliorer grandement la recherche d'information et aident au perfectionnement de ce processus dans les machines, comme exemple les moteurs de recherche. Ces technologies permettent d'augmenter la qualité des résultats et d'engager l'internaute dans un processus de recherche plus performant, c'est le cas avec la recherche à facettes<sup>4</sup>. Cependant, ces méthodes essentiellement syntaxiques possèdent des limites quand il s'agit des quantités massives et hétérogènes d'informations, notamment les connaissances textuelles formatées qu'en HTML.

Le manque de fonctionnalités d'analyse sémantique<sup>5</sup>, que ce soit par le biais de ces technologies d'analyse ou dans l'infrastructure des machines (Matériels et logiciels) rend l'intégration et l'interprétation d'informations de plus en

---

1. Information pertinente directement exploitable par son destinataire.

2. Traitement lexical apporté à un mot afin d'identifier son lemme.

3. Consiste à catégoriser les mots dans des classes telles que noms de personnes, noms de lieux, quantités, distances, valeurs, dates, etc.

4. Technique en recherche d'information par filtrage de collection de données en choisissant un ou plusieurs critères.

5. Analyse qui en établit la signification en utilisant le sens d'une ressource.

plus difficiles voire impossibles, car la plupart des informations sont conçues uniquement pour les humains qui peuvent lire et comprendre contrairement aux machines, celles-ci nous informent seulement sur les pages Web en ce qui concerne leurs popularités sur un sujet et les mots qu'elles portent dans leurs contenu, mais pas à comprendre ce dernier à cause de l'absence des moyens qui leur permettent de le faire.

En outre, la taille gigantesque et la forme textuelle des connaissances sur Web ont peu à peu été enfermées et rendues moins accessibles du fait qu'elles sont formatées dans une et seule couche présentationnelle, encore le sens du contenu du Web n'est pas traitable par machines. C'est un gros obstacle, toutefois il y a eu une idée vers la solution ça fait plusieurs années, celle du web sémantique, ce que nous allons voire prochainement.

### 1.2.1) Définition

L'idée a été proposée par Tim berne-se-lee<sup>1</sup> en 1998 visant à créer un Web qui permet une recherche plus précise et pertinente dans l'immense quantité de données qui s'y retrouve[2]. Il confirme que cette solution est : « une extension du Web actuel, dans laquelle l'information reçoit une signification bien définie, améliorant les possibilités de travail collaboratif entre les ordinateurs et les personnes » [3]. En 1999, Tim berne-se-lee a ajouté : “The first step is putting data on the Web in a form that machines can naturally understand, or converting it to that form. This creates what I call a Semantic Web—a web of data that can be processed directly or indirectly by machines” [4].

Techniquement, l'appellation "Web de données" est la plus juste, « Le terme sémantique prête un peu à confusion car la sémantique s'intéresse au sens du langage pour en déduire des constructions logiques ... En fait, nous aurions dû l'appeler dès le départ "Web de données". Mais il est trop tard pour changer de nom. »[5]. Appelé aussi "Web de données liées", faisant des liens automatiques pour relier les données qui sont stockées dans les différents fichiers et bases de

---

1. Principal inventeur du World Wide Web et HTML, fondateur de l'organisme World Wide Web Consortium (W3C) et son président depuis 1994.

données de nos ordinateurs, de la même façon que HTML permet de lire toutes sortes de documents et de les relier [5]. Il s'agit d'un réseau sémantique, une infrastructure sur laquelle les ressources seront cataloguées, la plus part de ces ressources sont virtuelles, codées avec des langages différents et réparties dans le monde.

Ainsi améliorer la capacité d'interprétation et d'interopérabilité des machines, après une bonne simulation du contexte et de sens de ces ressources. Ceci ce fait par le biais de nouvelles concepts et de technologies que nous détaillerons ci-après. Le Web sémantique diffère de celui actuel avec la possibilité de capturer et de traiter presque la totalité du contenu des pages Web, ce qui fait que les ressources sont accessibles et utilisables en ce qui concerne la recherche, le filtrage, l'intégration et l'affichage d'informations utiles. Des progrès considérables sont marqués permettant ainsi la possibilité de raisonner et favoriser une meilleure coopération entre Homme-machine et machine-machine, voire automatiser des tâches complexes et gérer les données plus intelligemment afin de répondre efficacement aux besoins des utilisateurs.

Cependant, rendre ces données énormes du Web accessibles et traitables conduit inévitablement à réfléchir sur d'autres formalismes plus souples et plus explicites quant à la représentation des connaissances, de ce fait un moyen approprié pour le faire est avec le concept d'ontologie.

### 1.3) Concept d'ontologie

En effet, chaque domaine d'activité particulière est la représentation d'un ensemble de connaissances sur son sujet.

Afin d'établir une certaine forme d'organisation des connaissances de chaque domaine, des regroupements et des catégorisations de concepts se font. Cela consiste à préciser la signification et la classification des concepts pouvant être utilisés, à caractériser les relations possibles entre eux et définir les contraintes possibles de leurs utilisation.

Plusieurs problèmes techniques empêchent l'utilisation d'applications fondés sur la connaissance, les besoins de techniques permettant leur intégration, réutilisation et analyse se multiplient à cause de l'explosion du nombre de ressources et de sources d'informations accessibles via le Web. Afin qu'une communication efficace soit possible entre les systèmes et les utilisateurs, une certaine modélisation des connaissances doit être effectuée.

#### 1.3.1) Définition d'ontologie

En sens large, beaucoup d'ontologues et conceptualistes, voire aussi des ingénieurs en informatique, disaient que le terme « Ontologie » est construit à partir des racines grecques « Ontos » (ce que existe, l'existant) et « Logos » (le discours, l'étude). Fabien Gandon<sup>1</sup> a dit que c'est « Un concept qui s'intéresse à l'existence et ses catégories fondamentales, c'est l'étude des propriétés générales de l'être, qui est, à son tour, les objets, concepts et autres entités supposées exister dans un domaine d'intérêt et les relations qui les unissent » [6]. Aussi, Thomas Gruber<sup>2</sup> définit l'ontologie comme « une spécification explicite d'une conceptualisation » [7].

Dans le contexte de l'informatique et des sciences de l'information, Gruber (1991) a dit : « Pour établir des accords sur la connaissance, tels que

---

1. Responsable de l'équipe-projet INRIA WIMMICS

2. Chercheur, concepteur spécialisé dans les technologies permettant d'accroître l'intelligence humaine. Il a été cofondateur, directeur de la technologie et vice-président de la conception chez Siri et inventeur de HyperMail.

des hypothèses partagées et des modèles du monde, les ontologies peuvent jouer un rôle de spécification logicielle » [7]. une ontologie est un moyen de modélisation des connaissances, conçu pour formuler, échanger et raisonner avec la connaissance d'un domaine d'intérêt. Les résultats des processus de modélisation sont appelés ontologies [8].

En systèmes de base de données, elles peut être considérées comme un niveau d'abstraction de modèles de données, analogue aux modèles hiérarchiques et relationnels, mais destinées à modéliser les connaissances sur les individus, leurs attributs et leurs relations avec d'autres individus. Pour cette raison, elles sont dites au niveau "sémantique", alors que les schémas de base de données sont des modèles de données au niveau "logique" ou "physique" [9].

L'ontologie est un outil puissant et définit un ensemble de primitives de représentation permettant de modéliser un domaine de connaissance. Ces primitives de représentation sont généralement des classes (ou des ensembles), des attributs (ou des propriétés) et des relations (ou des relations entre les membres de la classe) [9]. Elle fait partie de la pile de normes W3C pour le Web sémantique, et plusieurs ontologies sont conçues dans plusieurs domaines : ontologie médicale, de la génétique, la géométrie, etc.

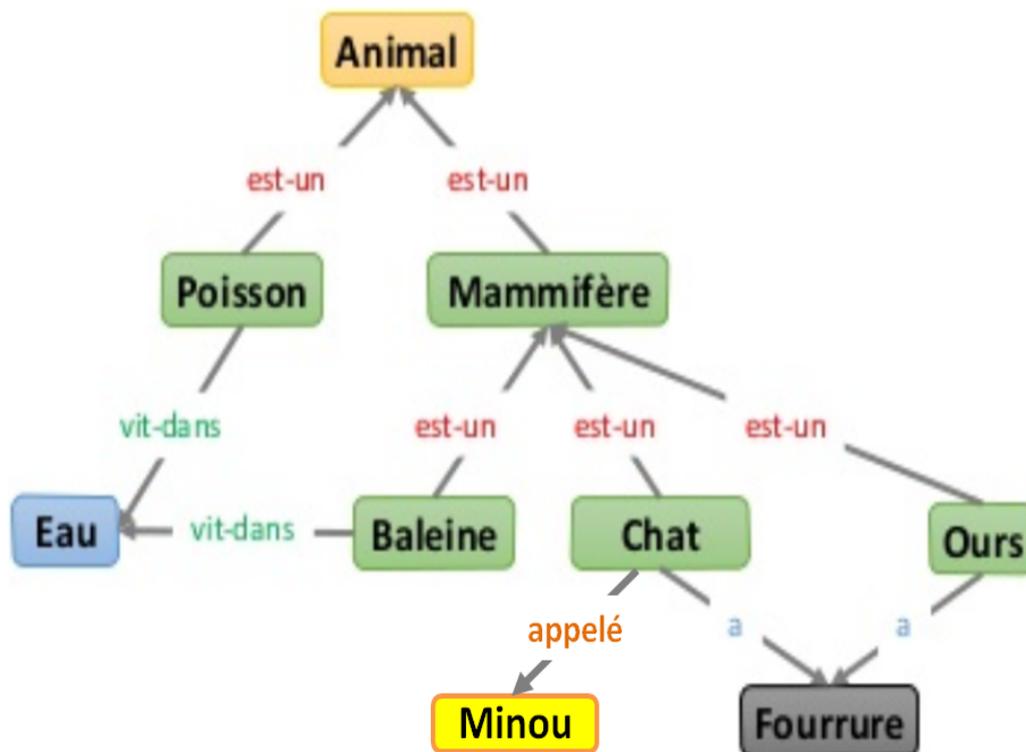


Figure 1 – Partie d’une ontologie du monde des "Animaux".

### 1.3.2) Historique des ontologies en informatique

La notion d’ontologie existait, sans être nommée, dès les années 1970 où on spécifie les types de mots qui existent dans notre représentation et leurs caractéristiques, en plus les hiérarchies de multi-héritage qui sont fini par schématiser les « Frames » et les classes des langages de modélisation par objets, d’ailleurs la programmation orientée objets ressemble à la notion d’ontologie en un ancêtre commun : les systèmes de l’intelligence artificielle symbolique (IAS) [6]. Les chercheurs d’IAS sont les premiers qui ont adopté le concept d’ontologie, ils avaient reconnu l’applicabilité des travaux à la logique mathématique et avaient soutenu qu’ils pouvaient créer de nouvelles ontologies sous forme de modèles informatiques, permettant certains types de raisonnement automatisé.

En 1980, ces chercheurs finissent par utiliser le terme « ontologie » pour désigner à la fois une théorie d’un monde modélisé et une composante des systèmes de connaissances. C’est jusqu’aux débuts des années 1990, en vue de créer des

normes d'interopérabilité, qu'on a identifié une pile technologique appelant la couche ontologie comme composant standard des systèmes de connaissance, et la notion se retrouve maintenant dans une large famille de systèmes d'information [9].

Actuellement, les ontologies est devenu un sujet de recherche populaire dans diverses communautés à savoir le traitement du langage naturel, la recherche d'information, l'ingénierie des connaissances, notamment, en l'étude des systèmes experts, l'intégration intelligente d'information et la gestion des connaissances, etc.

### 1.3.3) Conception d'ontologie

Dans ce qui suit, nous allons voir quelques titres qui survolent sur la conception des ontologies, à savoir leurs composants, critères de conception et une classification selon le type d'ontologie.

#### 1.3.3.1) Les composants d'ontologie

Comme nous l'avons abordé, une ontologie est constituée d'un ensemble de termes structurés permettant de représenter des connaissances. Cet ensemble tiré du monde réel, n'est qu'une conceptualisation organisée de connaissances liées par des relations sémantiques et logiques. La connaissance dans les ontologies est principalement formalisée en utilisant les cinq types de composants tels que : concepts (ou classes), relations (ou propriétés), fonctions, axiomes (ou règles) et instances (ou individus) :

##### 1) Les concepts

C'est est une notion abstraite qui peut représenter une classe, un ensemble de classe ou un type d'objet du monde réel. Il peut être divisé en trois parties sont : un terme, une intention et un ensemble d'objets appelés extension :

- **Le terme** : qui exprime le concept en langue naturelle par un élément lexical admet des synonymes.

- **L'intention du concept** : C'est une définition du concept, elle est exprimée en termes de propriétés et attributs, et de contraintes.
- **L'extension** : Elle regroupe les objets manipulés à travers le concept, ces objets sont appelés instances du concept.

Dans l'exemple de la figure (1), le terme «Chat» renvoie, à la fois, à la notion de table comme objet de type «Mammifère» possédant une «Fourrure», et à l'ensemble des chats de ce type.

## 2) Les instances

Les instances ou les individus constituent la définition extensionnelle de l'ontologie en représentant les objets réel manipulés, et véhiculent les connaissances (statiques, factuelles) à propos d'un domaine d'intérêt. Par exemple le terme « Minou » est une instance du concept « Chat ».

## 3) Les relations

Les relations désignent les interactions entre les éléments de l'ontologie et permettent d'apercevoir son hiérarchie. Si on se réfère à la figure (1), «est-un» est une relation. Les relations incluent des associations telles que :

- Etre sous classe d'une autre (généralisation/spécialisation).
- Un concept est une partie d'un autre (agrégation/composition).
- Un concept est associé à un autre.
- Un objet est une instance d'un concept.

## 4) Les axiomes

Les axiomes participent dans la description des concepts, des relations ou dans l'inférence de nouvelles informations, etc ; et considérés des assertions vraies à propos des abstractions du domaine concernant l'ontologie.

## 5) Les fonctions

Elles constituent des cas particuliers de relations, où un élément de la relation est défini en fonction des éléments précédents.

### 1.3.3.2) Cycle de vie d'une ontologie

Afin d'être en accord avec les standards de développement génie logiciel, notamment favoriser le repérage et la réparation des erreurs, ainsi éviter leur propagation au fur et à mesure du processus de développement, les ontologies informatiques sont considérées comme des composants logiciels dans les systèmes et répondant à des objectifs opérationnels différents et évolutifs, ainsi dotent d'un cycle de vie comporte sept activités posent des problèmes de recherche.

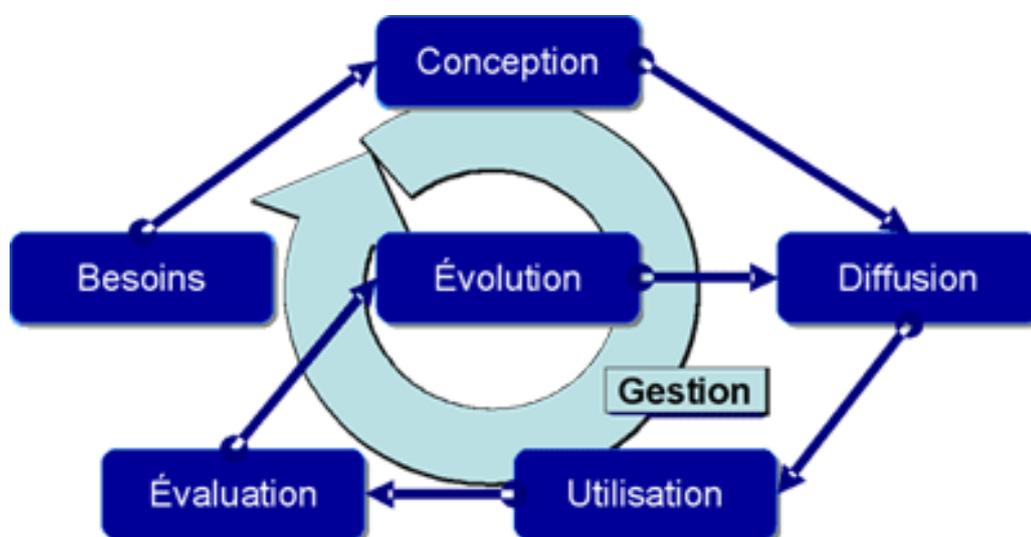


Figure 2 – Cycle de vie d'ontologie [6].

#### 1) Besoins et évaluation

Cette première étape consiste à définir les objectifs et les limitations de l'ontologie à développer, en précisant ces trois aspects [10] :

##### - L'objectif opérationnel :

Dans quel but utiliser cette ontologie? et à quel types de questions cette ontologie devra répondre?

**- Le domaine de connaissance :**

Quel domaine ? et quelles connaissances ?

**- Les utilisateurs :**

Qui va utiliser l'ontologie ? et qui va maintenir l'ontologie ?

**2) La conception et évolution**

Un problème se pose dans cette étape qui est l'obtention d'un consensus sur les choix de représentation et de conceptualisation d'ontologies en plus le maintenir pendant le cycle de vie des logiciels. Pour cela, on fait appel aux collecticiels et à des outils de gestion des points de vue, des terminologies et des jargons différents. On peut simuler cette étape par les deux grandes phases suivantes [10] :

**A. Première phase :**

Dans la première phases, deux choses à faire qui sont :

**A.1 La collecte des données :**

La première étape de conceptualisation dans la construction d'ontologie. la collecte de données se base sur des entretiens avec les experts du domaine d'intérêt pour obtenir une embauche de spécification et la liste détaillée des connaissances spécifiques.

**A.2 L'étude linguistique :**

En première fois, elle se fait de manière informelle, le but est d'extraire les principaux concepts, puis une second fois, de façon formelle, afin de retirer les différents concepts sous forme de triplet (terme, intention, extension), en outre identifier les relations associées en utilisant des outils de traitement de langue naturelle (approche syntaxique, approche statistique, analyse distributionnelle, etc.) qui permet de définir avec précision la nature des termes. Il s'agit d'un processus d'analyse terminologique.

## B. Deuxième phase :

Ainsi une étude sémantique et une formalisation des connaissances à réaliser dans la deuxième phase.

### B.1 L'étude sémantique :

Une normalisation sémantique s'effectue avec les experts du domaine, l'objectif est de fournir une terminologie fiable, rigoureuse et commune à tous les acteurs.

- **Identification des clés de concept** : Le but est de définir des fonctions qui définissent les concepts, en appliquant par exemple le paradigme de différentiel, il s'agit d'indiquer en quoi un concept est différent de ses frères et en quoi il est similaire à ses concepts parents.

- **Définition des intentions** : Définir les intentions de chaque concept en assurant leur complétude et consistance avec les termes existants, aussi indiquer les relations associées avec eux.

### B.2 Formalisation des connaissances :

Cette étape consiste à traduire les connaissances en notions compréhensibles par les machines en codant les différents notions de l'ontologie avec des langages de représentations formels tels que OWL ou RDF.

## 3) La diffusion

La diffusion d'une ontologie est accomplie par son déploiement et son mise en place où des problèmes bien contraints surgissent. Pour la diffuser, des architectures pair à pair ou autres architectures distribuées peuvent être utilisées dans le contexte de partage de fichiers, par contre des technologies adéquates sont exploitées en application web, et des architectures de services web dans l'intégration d'applications. La distribution des ressources (données, modèles, applications et utilisateurs) et leur hétérogénéité (syntaxes,

sémantiques, protocoles, contextes, etc.) sont causes de plusieurs problèmes tels que l'interopérabilité entre machines, l'alignement d'ontologies, l'optimisation d'inférences, la composition de services, etc. [6]

#### **4) L'utilisation**

L'étape d'utilisation regroupe beaucoup d'activités telles que le traitement de la langue, la rétro-ingénierie de base de données, la déduction de connaissances, aussi l'aide à la décision (moteurs d'inférence à base de règles), en outre la navigation assistée, l'identification et composition de services et l'analyse de gros volumes de connaissances (Clustering). De nouveaux problèmes sont posés dans cette phase comme la complexification des modèles de données qui engendre des problèmes pour leur représentation et l'interaction avec ces représentations, ceci est la conséquence des inférences qui exploitent les ontologies pour la génération dynamique d'éléments d'interfaces, afin de résoudre le problème de conception des interactions avec l'utilisateur et de leur ergonomie [6].

#### **5) La gestion**

L'activité permanente de gestion et de planification souligne qu'il est important d'avoir un travail de suivi et une politique globale pour détecter ou déclencher, préparer et évaluer les itérations du cycle et s'assurer que l'on reste dans le cercle vertueux des systèmes d'information [6].

### **1.3.3.3) Classification d'ontologies**

Différents travaux ont amené à classer les ontologies selon plusieurs critères, nous citons ci-dessous quatre classifications qui sont :

#### **1) Classification selon l'objet de conceptualisation**

Cette catégorisation est la plus courante, elle rassemble beaucoup de types d'ontologies et on distingue, les ontologies de représentation des connaissances, les ontologies communes, génériques, celles du domaine, à savoir de tâches, aussi les ontologies de domaines-tâches, d'application, en outre légères, etc.

## 2) Classification selon le niveau de complétude

Pour la classification des ontologies selon leur niveau de complétude, on trouve que deux types : niveau sémantique et niveau référentiel.

## 3) Classification selon le niveau de détail

Aussi, selon le niveau de détail il y a deux classifications telles que : les ontologies de granularité fine et d'autres de Granularité large.

## 4) Classification selon le niveau de formalisme

Selon le niveau de formalisme il y a quatre catégories qui sont : Les ontologies informelle, les ontologies semi-informelle, en plus celles qui sont formelle et celles d'autres semi-formelle.

### 1.3.3.4) Critères de conception d'une ontologie

Lorsque nous choisissons comment représenter des connaissances sous forme de modèle ontologique, nous prenons des décisions de conception, et pour guider et évaluer celle-ci il est utile d'avoir des critères objectifs fondés sur le but du modèle de l'ontologie résultant. Gruber a proposé un ensemble préliminaire de règles de conception pour des ontologies ayant pour but le partage de connaissances et l'interopérabilité entre programmes basés sur une conceptualisation partagée [7] :

#### 1) La clarté

Afin de communiquer efficacement le sens voulu des termes définis :

- Les définitions des notions doivent être objectives et complète par des conditions nécessaires et suffisantes.
- La définition des concepts doit être indépendante du contexte social ou informatique.
- Toutes les définitions des notions doivent être documentées en langage naturel.

## 2) La Cohérence

Une ontologie doit être cohérente et elle doit sanctionner les inférences qui sont compatibles avec les définitions : les axiomes et les concepts devraient être logiquement cohérents. Si une phrase qui peut être déduite des axiomes contredit une définition ou un exemple donné de manière informelle, alors l'ontologie est incohérente.

## 3) L'extensibilité

Une ontologie doit être conçue pour anticiper les utilisations du vocabulaire partagé, le but est de ne pas nécessiter une révision supplémentaire des définitions existantes :

- Elle devrait offrir une base conceptuelle pour un éventail de tâches prévues.
- La représentation devrait être conçue de manière à pouvoir étendre et spécialiser l'ontologie de manière monotone.

## 4) Un biais minimal de codage

La conceptualisation doit être spécifiée au niveau de la connaissance sans dépendre d'un codage particulier d'implémentation : les biais de codage doivent être minimaux, car les programmes de partage de connaissances peuvent être implémentés dans différents systèmes et styles de représentation.

## 5) Un engagement ontologique minimal

Une ontologie doit exiger un engagement ontologique minimal suffisant pour pouvoir réaliser toutes les activités de partage de connaissances prévues : une ontologie doit faire le moins de déclarations possible sur le monde modélisé, en ne définissant que les termes essentiels à la communication du savoir cohérent, permettant aux parties engagées de se spécialiser et d'instancier l'ontologie en fonction des besoins.

### 1.3.4) Les outils de construction d'ontologie

Les ontologies sont généralement spécifiées dans des langages permettant d'abstraction des structures de données et des stratégies de mise en œuvre, en pratique, les langages des ontologies ont un pouvoir expressif plus proche de la logique de premier ordre que les langages utilisés pour modéliser des bases de données [7]. Il existe maintenant des langages standards et une variété d'outils commerciaux et open source pour créer et utiliser des ontologies. On distingue deux familles d'outils :

#### 1.3.4.1) Les outils dépendants de formalisme de représentation

##### 1) Ontolingua

Ontolingua est un serveur d'édition d'ontologies. Pour décrire une ontologie il utilise les cinq notions (classes, des relations, des fonctions, des instances et des axiomes). Ce serveur offre la possibilité d'intégrer les ontologies Ontolingua, ce qui signifie qu'on peut avoir une construction modulaire des ontologies.

##### 2) OntoSaurus

Un outil de l'Information Science Institute de l'Université de Southern California, il se compose de deux modules : un serveur utilisant LOOM comme langage de représentation des connaissances, et un autre pour la navigation qui crée dynamiquement des pages HTML permettant d'afficher la hiérarchie de l'ontologie. Afin d'éditer une ontologie, des formulaires HTML sont utilisés.

##### 3) WebOnto

C'est une application Web pour la création, l'édition, plus la navigation et le développement collaboratif des ontologies sur le Web, avec OCML comme langage d'implémentation. Il distingue quatre types d'ontologies : ontologie de domaine, ontologie de tâche, ontologie de méthode, et ontologie d'application.

### 1.3.4.2) Les outils indépendants de formalisme de représentation

#### 1) Protégé

Protégé est une plate-forme gratuite, modulaire et à code source ouvert, permettant l'édition, la visualisation, le contrôle d'ontologies, plus l'extraction à partir de sources textuelles, et la fusion semi-automatique. Grâce à l'intermédiaire des plugins, Protégé est capable de manipuler bon nombre de formats tels que RDF, RDFS, OWL et XML Schéma. Il est très utilisé du fait de sa puissance [11].

#### 2) ODE et WebOde

L'outil ODE nous permet de construire des ontologies au niveau connaissance, comme le préconise la méthodologie METHONTOLOGY. Il permet de construire les ontologies dans un modèle de type frame, en spécifiant leur composants.

#### 3) OntoEdit

Il dote d'outils graphiques dédiés à la visualisation, est également un environnement de construction d'ontologies indépendant de tout formalisme. Plusieurs utilisateurs peuvent éditer une ontologie à l'aide d'un serveur intégré, ainsi contrôlent sa cohérence assurée à travers la gestion des ordres d'édition.

### 1.3.5) Apports d'ontologie

Les ontologies sont prometteuses pour la réalisation du web sémantique et apporte beaucoup au Web :

- Elles fournissent une compréhension commune d'un domaine d'intérêt afin de faciliter la communication entre agents humains et agents informatiques hétérogènes[1].
- Elles sont utilisées pour intégrer des bases de données hétérogènes, permettant une interopérabilité entre systèmes et pour spécifier des interfaces avec des services indépendants basés sur la connaissance. [19].
- Le modèle le plus formel, car une ontologie définit le sens des concepts en modélisant des contraintes qui limitent le nombre d'interprétations possibles, ce

qui aident à renforcer l'efficacité et l'uniformité de la description des ressources, ainsi permettre des fonctionnalités plus sophistiquées dans la représentation et récupération d'informations.

- Elles offriraient des possibilités d'inférence aux systèmes informatiques, de réflexivité sur leurs connaissances et leurs traitements, d'introspection, d'alignement dynamique pour permettre l'interopérabilité, d'évolution dynamique et de description des composants logiciels, afin de rendre leurs interactions plus dynamiques et leur gestion plus automatique, et d'aller vers une informatique plus autonome [6].

- Les ontologies décrivent les logiciels (par exemple, des services web), leurs caractéristiques fonctionnelles, et non fonctionnelles, ainsi permettre l'identification, l'invocation et la composition dynamique d'applications à l'échelle du web [6].

- Elles sont utilisées dans la description et l'application de règles de sécurité et de confidentialité décrites à de hauts niveaux d'abstraction, permettant de restreindre les accès avec une grande flexibilité, confidentialité et précision de l'information diffusée [6].

- On peut imaginer un nouveau paradigme de programmation, où les structures de données seraient des représentations basées sur des ontologies partagées, et où les applications seraient obtenues par composition de services (logiciels personnels, services en ligne, etc.) [6].

#### 1.4) Architecture du web sémantique

En plus des trois standards fondamentaux au cœur de l'architecture du Web qu'on connaît, à savoir l'URI (Universal Resource Identifier), le protocole HTTP et le HTML, de nombreux langages informatiques sont impliqués dans le Web sémantique et qui peuvent être traduits par une hiérarchie sous forme de couches, où chaque couche exploite les capacités et les résultats des couches qui se trouvent en dessous d'elle. Ces langages sont en amélioration permanente, alors l'architecture du Web sémantique est toujours en évolution au fil du temps

[12]. La figure suivante regroupe les différents standards et technologies utilisés afin présenter les principales couches de l'architecture tel qu'elles étaient décrites par Tim Bernes Lee.

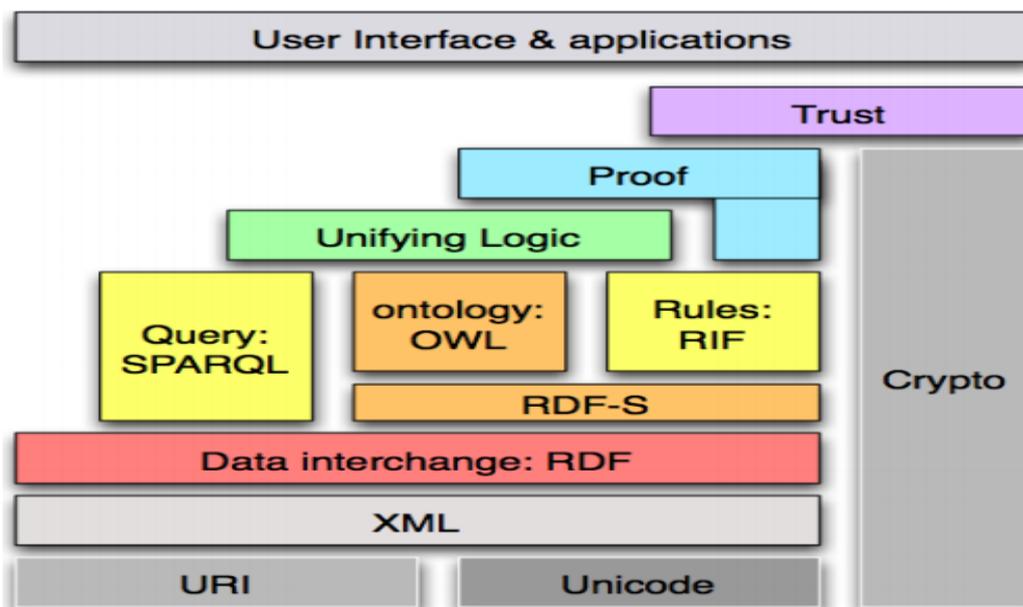


Figure 3 – L'architecture « Layer cake » proposée par Tim Berners Lee [12].

## 1.5) Technologies utilisées en Web sémantique

Afin d'organiser et relier les connaissances entre elles d'une manière flexible, ainsi présenter la donnée d'une façon syntaxiquement et sémantiquement interopérable, le W3C fournit une palette de technologies (Métadonnées, RDF, RDFS, RFI, URI, OWL, SKOS, SPARQL, ...), recommandées et d'autres standardisées, pour accéder à ces données. Ces outils sont décrits ci-après :

### 1.5.1) Les identificateurs de ressources uniformes

Les identificateurs de ressources uniformes (URI) sont des séquences de caractères avec une syntaxe restreinte, pouvant servir de référence à toutes choses qui possèdent une identité. Ils fournissent un moyen simple et extensible d'identifier une ressource d'une manière standard et unique, ainsi permettre une interprétation sémantique uniforme de cette dernière. Il englobe l'URL, l'URN et l'URC [13].

## 1.5.2) Les métadonnées

La fédération IFLA<sup>1</sup> donne une bonne définition aux métadonnées, ainsi : « Les métadonnées sont des données sur les données. Le terme fait référence à toutes les données utilisées pour faciliter l'identification, la description et la localisation des ressources électroniques en réseau. Il existe de nombreux formats de métadonnées, certains assez simples dans leur description, d'autres assez complexes et riches » [14].

Tim Berners-Lee a dit que les : “Metadata is machine understandable information about Web resources or other things” et elles peuvent se trouver dans le document, ou dans un document séparé, ou être transférées avec le document. [...] Des assertions sur des données. De telles assertions, généralement, lorsqu'elles sont représentées dans des systèmes informatiques, se présentent sous la forme d'un nom ou d'un type d'assertion et d'un ensemble de paramètres, tout comme dans le langage naturel une phrase prend la forme d'un verbe et d'un sujet, un objet et diverses clauses [15]. Ces définitions nous amènent au concept de données comprises par les ordinateurs lors de l'échange d'informations. Pour cela, le moyen le plus pratique de procéder consiste à créer un vocabulaire structuré commun de base, parfois appelées "couches supérieures RDF"[15].

Dans ces deux figures suivantes un exemple illustre comment porter plus de connaissances sur les informations avec les métadonnées en utilisant les URI est donné :

```
...  
<h2>The Trouble with Bob</h2>  
<p>Date: 2011-09-10</p>  
...
```

**Figure 4** – Données échangées sans utilisation de métadonnées. [16]

---

1. La Fédération internationale des associations de bibliothécaires et des bibliothèques est le principal organisme international représentant les intérêts des bibliothèques et des services d'information et de leurs utilisateurs.

```
...  
<h2 property="http://purl.org/dc/terms/title">The Trouble with Bob</h2>  
<p>Date: <span property="http://purl.org/dc/terms/created">2011-09-10</span></p>  
...
```

Figure 5 – Données échangées avec utilisation de métadonnées. [16]

### 1.5.3) Extensible Markup Language

Le langage Extensible Markup Language (XML) a été développé par le groupe de travail XML, également créé et organisé par le (W3C) en 1996. Il décrit une classe d'objets de données appelés documents XML et décrit en partie le comportement des programmes informatiques qui les traitent. C'est un format de texte simple et très flexible dérivé de SGML (ISO 8879). Un module logiciel appelé processeur XML est utilisé pour lire des documents XML et fournir un accès à leur contenu et à leur structure [17].

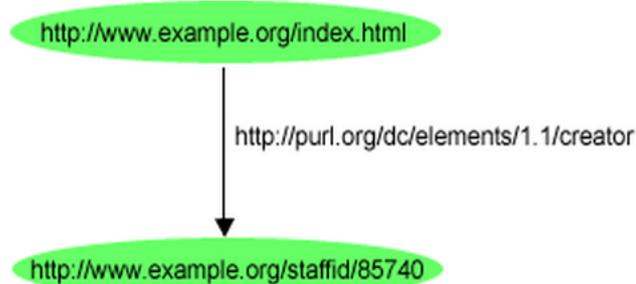
### 1.5.4) Resource Description Framework

Le RDF (Resource Description Framework) est un langage permettant l'expression, dans un cadre commun, d'informations sur les ressources, ces informations sont traitées et échangées entre applications sans perte de sens. Il est particulièrement destiné à la représentation de métadonnées, en se basant sur les URI, cela permet de représenter des instructions simples sur les ressources sous forme de graphique de nœuds et d'arcs représentant des triplets composés d'un sujet (un concept), d'un prédicat (propriété du concept) et d'un objet (la valeur de la propriété).

Ces triplets établissent un modèle de données qui est une ontologie. Egalement une syntaxe basée sur XML (appelée RDF / XML) est fourni pour l'enregistrement et l'échange de ces graphiques RDF. Ainsi par exemple, l'instruction RDF suivante : « `http://www.example.org/index.html` has a creator whose value is John Smith » peut être représenté par une déclaration RDF ayant [18] :

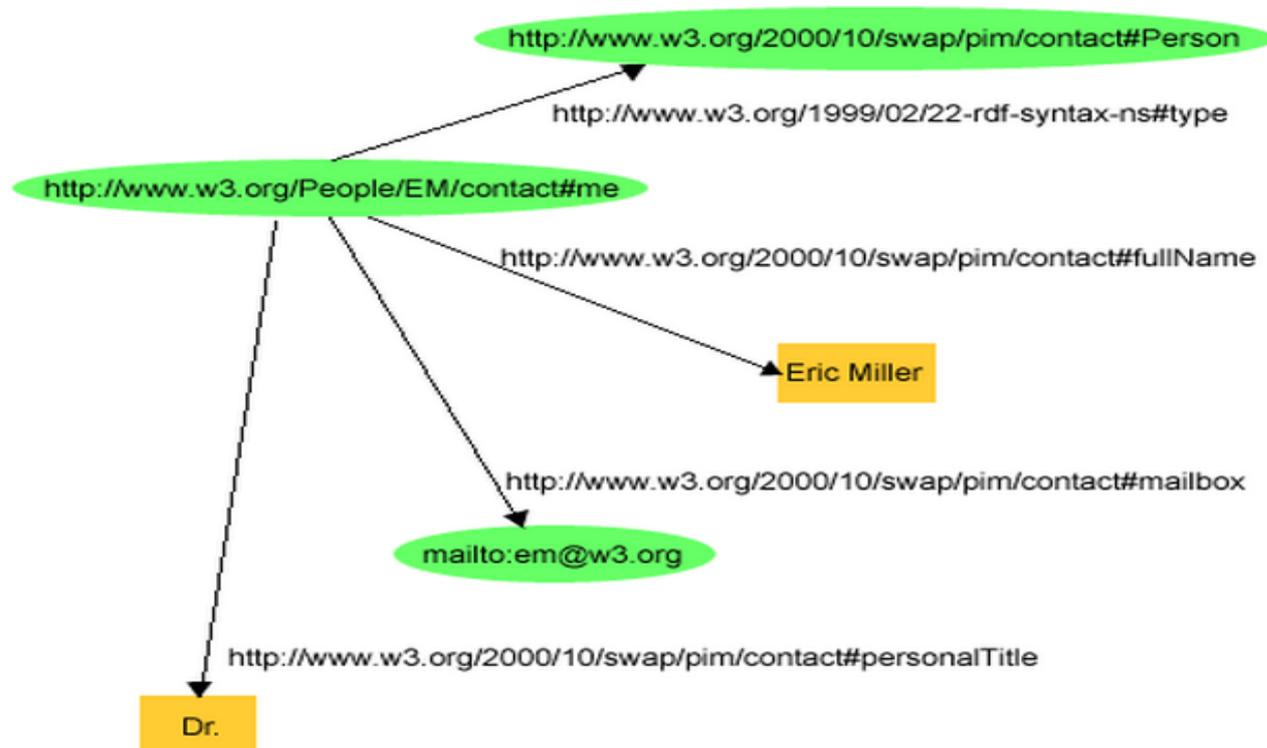
- Un sujet `http://www.example.org/index.html`
- Un prédicat `http://purl.org/dc/elements/1.1/creator`
- Et un objet `http://www.example.org/staffid/85740`

Et serait représentée par le graphique présenté à la figure ci-dessous :



**Figure 6** – Une simple déclaration représentée par un RDF graphique [18].

Ainsi, cette petite paragraphe "il existe une personne identifiée par «`http://www.w3.org/People/EM/contact#me`», dont le nom est «Eric Miller», son adresse de messagerie est «`em@w3.org`», et dont le titre est «Dr», pourrait être représenté par le graphe RDF à la [Figure - 4] ci-dessous :



**Figure 7** – Un graphique RDF décrivant la personne « Eric Miller » [18].

La [Figure - 8] qui suit est une petite partie de RDF dans RDF / XML correspondant au graphe de la [Figure - 7] précédente :

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#">

  <contact:Person rdf:about="http://www.w3.org/People/EM/contact#me">
    <contact:fullName>Eric Miller</contact:fullName>
    <contact:mailbox rdf:resource="mailto:em@w3.org"/>
    <contact:personalTitle>Dr.</contact:personalTitle>
  </contact:Person>

</rdf:RDF>
```

Figure 8 – RDF / XML décrivant « Eric Miller » [18].

### 1.5.5) Resource Description Framework Schema

C'est le langage de description de vocabulaire de RDF, il sert à une structure de base commune et graphique décrivant des moyens de définir la nature des concepts et propriétés correspondent aux prédicats dans les triplets, et de combiner des combinaisons de ceux-ci de manière significative [19].

Un exemple simple est présenté ici sous forme de diagramme, illustrant l'utilisation du vocabulaire de schéma RDF pour décrire des classes et des propriétés, ainsi que les relations entre données :

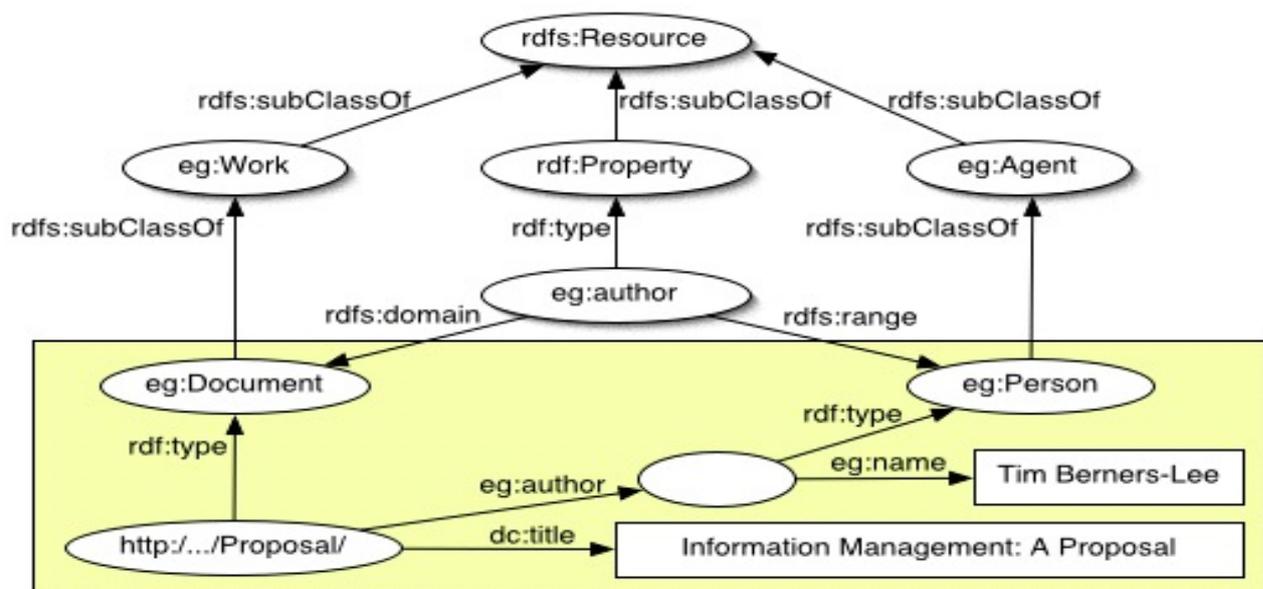


Figure 9 – Un exemple schématique de RDFS [20].

Cette logique appartient à un domaine de recherche qui a pour but d'aider le raisonnement sur une base de connaissances [21].

### 1.5.6) Web Ontology Language

Actuellement, OWL est le standard proposé par W3C pour concevoir les ontologies, il facilite mieux l'interprétation et le traitement du contenu Web qu'avec XML, RDF et RDFS en fournissant des moyens plus expressifs, par un vocabulaire supplémentaire plus complet pour la description d'ontologies complexes, comme par exemple la disjonction de classe, les types de propriétés plus riches, la cardinalité, les caractéristiques de propriété (symétrie, transitivité, ...), l'égalité et les classes énumérées etc ; ainsi qu'une sémantique plus formelle. OWL a trois sous-langages de plus en plus expressifs : OWL Lite, OWL DL et OWL Full [22] :

#### 1.5.6.1) Web Ontology Language LITE

C'est le plus simple et moins expressif parmi les trois langages, du fait de l'utilisation de quelques fonctionnalités du langage OWL. Par exemple, on ne peut utiliser que certains types de contraintes de classes, et celles-ci ne peuvent être définies qu'en termes de superclasses nommées (les superclasses ne peuvent pas être des expressions arbitraires) et seuls certains types de restrictions de classe peuvent être utilisés. En plus, permet une notion limitée de cardinalité (0 ou 1).

#### 1.5.6.2) Web Ontology Language LD

OWL DL qui correspond à la logique descriptive, est un langage d'expressivité plus que OWL Lite, avec une séparation de type (une classe ne peut pas être un individu ni une propriété, et vis vers ça), ainsi on ne peut appliquer des restrictions aux éléments de langage de OWL lui-même, ce qu'on trouve dans OWL Full.

### 1.5.6.3) Web Ontology Language FULL

Cet langage est fait pour ceux qui cherchent un maximum d'expressivité et la liberté syntaxique de RDF, ça veut dire pas de séparation entre classes, propriétés, individus et valeurs des données, ce qui fait qu'il contient toutes les constructions du langage OWL, cependant l'utilisation complète de ces fonctionnalités engendre une perte de certaines garanties pour les systèmes de raisonnement, ce que OWL DL et OWL Lite peuvent fournir.

### 1.5.7) Rule Interchange Format

Le format d'échange de règles du W3C (RIF) est une norme qui a été développée pour faciliter l'intégration et la synthèse de jeux de règles d'inférence sur le Web sémantique. C'est une famille de dialectes interconnectés représentant des langages de règles dotés de diverses caractéristiques. Il permet de découvrir et de générer de nouvelles relations sur la base de relations existantes entre les ressources. Ceci, pour améliorer la recherche d'informations pertinentes en interrogeant plusieurs bases de données, et augmenter la qualité de l'intégration de données sur le Web et analyser automatiquement les ressources, également pour découvrir les incohérences possibles dans les données intégrées, par conséquent, optimiser les problèmes de raisonnement dans les données [23][24].

### 1.5.8) SPARQL Protocol And Resource Query Language

Cette technique, recommandée par le W3C depuis 2008, est à la fois un protocole et un langage permettant d'exprimer des requêtes d'interrogation des graphes RDF par filtrage. La puissance de ce langage d'interrogation réside dans sa capacité à interroger plusieurs sources de données distribuées à l'échelle du Web [25].

### 1.5.9) Apache Jena

Apache Jena est un Framework Java permettant de construire des applications Web sémantiques. Il fournit un environnement de programmation pour RDF, RDFS et OWL, SPARQL et inclut un moteur d'inférence basé sur des règles [26].

### 1.6) Apports du Web sémantique

Le Web sémantique apportera une structure au contenu significatif des pages Web, en créant un environnement dans lequel les agents logiciels se déplaçant de page en page peuvent effectuer facilement des tâches sophistiquées aux humains, et exploitent directement la sémantique des ressources. Un tel agent venant sur la page Web d'une clinique saura que non seulement la page contient des mots clés tels que "traitement, médecine, physique, thérapie" (comme cela pourrait être codé aujourd'hui), mais aussi qu'un docteur de tel nom travaille dans cette clinique, en certaines journées de la semaine, et que le script prenne une plage de dates à un format précis comme "aaaa-mm-jj" et renvoie les heures de rendez-vous.

Dans un proche avenir, ces développements ouvriront la voie à de nouvelles fonctionnalités significatives, les machines devenant beaucoup plus aptes à traiter et à comprendre les données qu'elles affichent[27].

### 1.7) Conclusion

D'après ce que nous avons présenté dans ce chapitre, il s'avère que le Web sémantique offre des perspectives d'avenir très prometteuses nous permettant d'exploiter toutes les données enregistrées dans les bases de données. La banque de données "DBPedia" est l'un des exemples d'application du principe de données liées. Cette banque contient les données de Wikipédia sous formats de triplets RDF, également pour intégrer des liens vers d'autres ensembles de données sur le Web. La promesse du Web Sémantique est de libérer les utilisateurs des tâches encombrantes et pénibles.

# Alignement des ontologies

## 2.1) Introduction

## 2.2) Les sources d'hétérogénéités dans les ontologies

## 2.3) Définition de l'alignement d'ontologies

## 2.4) Les méthodes d'alignement

## 2.5) Opérations sur les ontologies

## 2.6) Conclusion

### 2.1) Introduction

Dans ce chapitre, nous définirons la signification du processus d'alignement d'ontologies après avoir présenté les raisons qui ont conduit à la nécessité de les aligner. A la fin du chapitre nous citons quelques opérations qu'on peut réaliser à travers l'alignement des ontologies.

## 2.2) Les sources d'hétérogénéités dans les ontologies

De nombreuses applications sémantiques ont entraîné la présence de plusieurs ontologies, parfois pour le même domaine. Ces ontologies sont développées avec différentes manières selon la personne ou l'organisation qui l'a modélisé, en vue d'obtenir l'abstraction des connaissances dans un domaine d'intérêt. Le résultat est l'existence d'un ensemble de structures et modèles ontologiques hétérogènes, d'où plusieurs problèmes d'ambiguïté et un risque inévitable de voir une hétérogénéité sémantique entre les ontologies, car elles ne sont homogènes ni dans leur structures, ni dans leurs sémantiques, parce qu'elles ont été conçues par différents développeurs et suivant différents principes et modèles.

Ainsi, afin d'établir des liens entre les connaissances exprimées dans chacune des représentations et modèles ontologiques, le processus d'alignement entre les ontologies peut également aider à la visualisation des correspondances entre les entités (concepts, propriétés ou instances) et générant le plus automatiquement possible des relations ou appariements entre eux [28]. Mais avant d'expliquer ce processus, nous présentons brièvement, les principales formes de problèmes issues de cette hétérogénéité en quatre niveaux dans ce qui suit [29] :

### 2.2.1) Le niveau syntaxique

Dans ce premier niveau, l'hétérogénéité est causée par l'utilisation de langages de représentation et de formalismes de modélisation différents, mais rien n'empêche, dans certains cas, de traduire les ontologies dans différents langages ontologiques sans préserver la signification.

### 2.2.2) Le niveau conceptuel

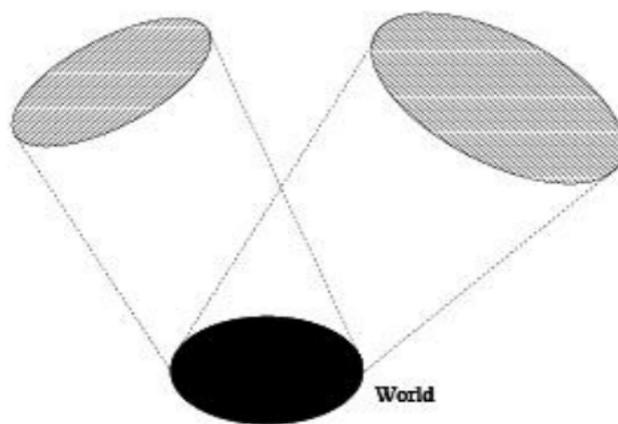
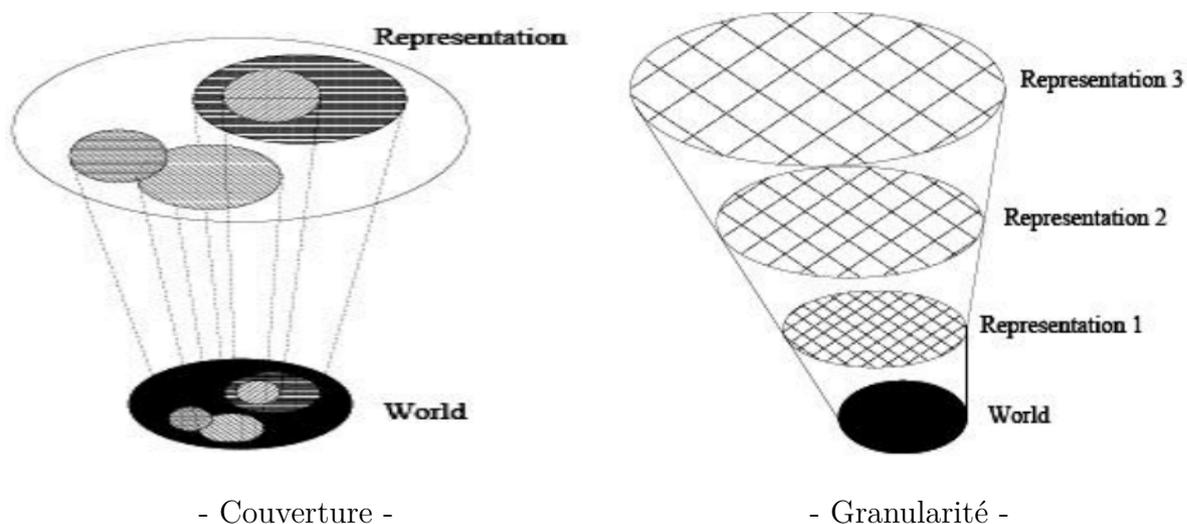
Les divergences à ce niveau peuvent être résumées en trois aspects :

- **La couverture** : Deux ontologies peuvent couvrir des sous-ensembles de connaissance d'un domaine donné ou alors des parties qui se chevauchent. Donc la différence entre elles peuvent être au niveau de la portée de la couverture du

domaine décrit.

- **La granularité** : des ontologies peuvent représenter les mêmes connaissances, mais avec différents degrés d'expression des détails.
- **La perspective** : deux ontologies décrivent un même domaine, avec un même degré d'expression des détails, mais avec des points de vue différents.

Ainsi la [Figure - 10] illustre graphiquement comment une ontologie peut différer d'une autre à travers ces trois aspects :



Perspective

**Figure 10** – Les trois dimensions de l'hétérogénéité au niveau conceptuel [29].

### 2.2.3) Le niveau terminologique

L'utilisation des sous-langages techniques spécifiques à un domaine de connaissances bien déterminé ou de différents langages naturels peuvent être les causes à ce niveau, et comme exemples pour ce type d'hétérogénéité nous citons :

- L'ambiguïté causée par la synonymie, la polysémie, l'homonymie qui nécessite bien souvent l'intervention humaine.
- L'utilisation des mots provenant de différentes langues naturelles pour désigner un même concept.
- Les variations syntaxiques du même mot comme l'insertion des abréviations, des préfixes et des suffixes, etc.
- Ce qui concerne l'encodage des données, que ce soit pour les dates, les unités (monnaie, distances, etc.).

### 2.2.4) Le niveau sémantique ou pragmatique

En intégrant à la manière dont les entités ontologiques sont interprétées par leurs utilisateurs, l'ambiguïté se produit lorsqu'il y a une différence d'interprétation de la même ontologie par différentes personnes ou différentes communautés. Ces différences d'interprétation sont dues principalement à la diversité des contextes et des domaines d'application des ontologies, ainsi ce type d'hétérogénéité reste difficile à détecter par la machine.

## 2.3) Définition de l'alignement d'ontologies

Dans la littérature, plusieurs définitions de l'alignement peuvent se rencontrer selon la nature de la structure alignée, par exemple en mathématiques discrètes pour faire correspondre des graphiques et des arbres, dans des bases de données pour rapprocher et fusionner des schémas, en apprentissage automatique pour regrouper des objets composés décrits dans une logique de premier ordre restreinte. Fondamentalement, l'alignement revient à

définir une distance entre entités (qui peut être aussi réduite qu'un prédicat d'égalité) et à calculer la meilleure correspondance sémantique des entités qui composent les ontologies, c'est-à-dire celle qui minimise la distance totale (ou maximise une mesure de similarité)[28].

C'est un processus exécuté selon une stratégie ou une combinaison de techniques de calcul de mesures de similarité, plus l'utilisation d'un ensemble de paramètres (ex : paramètres de pondération, seuils ...) et un ensemble de ressources externes (ex : thésaurus, lexique...). A la fin de cette tâche complexe, nous obtenons un ensemble de liens sémantiques, qui comprennent des relations d'équivalence, de généralisation/spécialisation, de chevauchement ou encore d'incompatibilité, reliant les composantes des ontologies alignées [30].

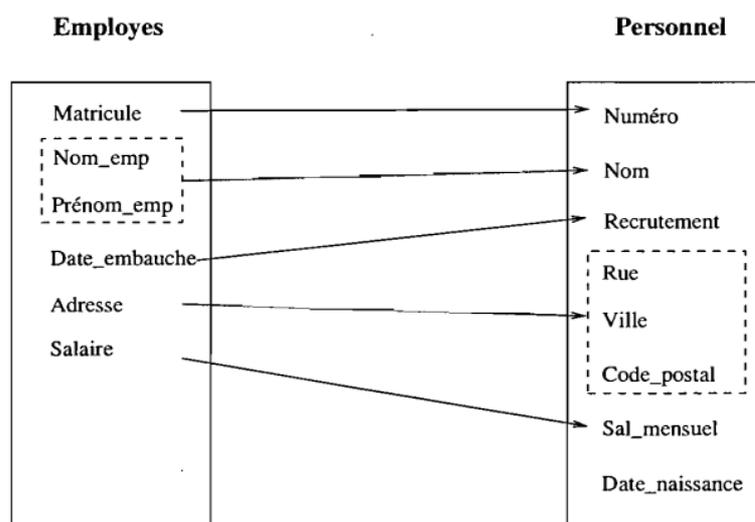


Figure 11 – Exemple d'alignement sémantique [31].

## 2.4) Les méthodes d'alignement

De nombreux travaux ont été développés dans le domaine de l'alignement d'ontologies et portent sur les techniques d'évaluation de mesures de similarité sémantique et sur les outils ou sur les Framework qui les intègrent [30]. L'objectif de ces mesures est d'évaluer la proximité sémantique entre les concepts, ainsi de déterminer s'ils sont similaires ou peuvent être également liés sémantiquement par des relations lexicales : synonymie, spécialisation, etc. Ces techniques

d'alignement sont classés selon plusieurs critères tel que : les techniques et les stratégies employées, les types d'information représentées dans les structures à aligner et l'objectif de l'alignement, cependant on peut les classifiées en deux grandes catégories qui sont [32].

### **2.4.1) Les méthodes locales**

Elles s'intéressent à calculer la similarité entre les couples d'entités en exploitant les informations locales (leurs URIs, leurs structures, etc.) sans tenir en considération l'aspect global de l'ontologie [32].

#### **2.4.1.1) Méthodes à base de la comparaison lexicale**

Ces méthodes, appelées aussi méthodes terminologiques, reposent sur l'analyse syntaxique entre autre l'égalité de chaînes, la distance de Hamming et la recherche de sous chaînes. Elles sont employées pour calculer la valeur de similitude des éléments textuels associé à l'entité (noms, des métadonnées sur les noms, des URIs des étiquettes, des commentaires, etc.) [30].

#### **2.4.1.2) Méthodes à base de traitement des langages naturels**

Ou méthodes linguistiques, reposent plutôt sur les principes du traitement de langages naturels pour construire des listes, par exemple, des synonymes et des hyponymes qui servent pour calculer la relation ou la similarité entre deux entités, à partir des liens sémantiques déjà existants dans les ressources externes (dictionnaires, taxonomies, WordNet, ...) [30].

#### **2.4.1.3) Méthode à base de comparaison structurelle des entités**

La comparaison de structures d'entités est une technique à base de plusieurs méthodes d'alignement et divisée en deux comparaison[32] :

**1) Les méthodes structurelles internes :** On calcule la similarité entre deux concepts en exploitant les informations relatives à leur structure interne

(restrictions et cardinalités sur les attributs, valeurs des instances, ...).

**2) Les méthodes structurelles externes ou conceptuelles :** Permettent d'exploiter la structure hiérarchique de l'ontologie, en se basant sur des techniques de comptage des liens d'hierarchie entre les entités pour déterminer la similarité sémantique.

#### 2.4.1.4) Méthodes extensionnelles

Ces méthodes déduisent la similarité entre deux concepts en réalisant des comparaisons sur leurs ensembles d'extensions (instances) au lieu de leurs interprétations [30].

#### 2.4.2) Les méthodes globales

Ces méthodes, en revanche, exploitent de manière extensive les ontologies en combinant les mesures de similarité existantes selon des critères donnés. Elles utilisent les similarités calculées par des méthodes locales, ces mesures ainsi obtenues sont exploitées par des algorithmes (des fonctions d'inférence, d'agrégation, etc.) pour produire des alignements [32]. Parmi ces méthodes globales nous pouvons situer :

- Le graphe de propagation de similarité
- Les méthode de composition
- Les méthodes d'apprentissage
- Les réseaux de neurones.

Certains proposent des systèmes d'aide au choix de mesures de similarité à la demande, où la principale différence réside dans la stratégie utilisée pour découvrir la similarité terminologique et/ou structurelle et/ou extensionnelle entre deux entités, et se comptent sur l'utilisation de plusieurs mesures de similarité afin d'avoir de meilleurs résultats. L'intervention d'un expert de domaine dans le processus d'alignement d'ontologies s'avère souvent essentielle pour éviter des incohérences. D'autres propose également de guider cet expert

pour choisir les méthodes de mesures de similarité les plus appropriées à sa requête et ce, dans le but de lui offrir une meilleure interaction et des suggestions de mappage pertinents [30].

En outre, des systèmes s'appuyant sur plusieurs approches d'alignement multi-stratégiques et qui peuvent être combinées de façon dynamique en prenant en considération les spécificités sémantiques des concepts en cours de réconciliation ont été réalisés. Ces systèmes se distinguent par leur faculté d'enrichir les sources de données par des informations sémantiques extraites de leur environnement de développement avant de procéder à leur alignement proprement dit, ce qui permet d'élaborer des mappages sémantiques en tenant compte du contexte et la nature diversifiée des sources à aligner [31].

### 2.4.3) WordNet

WordNet est une base de données lexicale très utilisée dans le traitement en langage naturel et la recherche d'informations. Plus récemment, il a également été adopté par la communauté de recherche sur le Web sémantique. Wordnet est un système de référence lexicologique. Les noms, les verbes, les adjectifs et les adverbes en anglais sont organisés en des ensembles des synonymes. Ces ensembles des synonymes, appelés synsets, sont reliés par différentes relations sémantiques [33].

En utilisant Wordnet, nous pouvons trouver les synonymes d'un mot ou d'un terme. Par exemple, l'un peut choisir le terme « voiture » pour le nom de la classe dénotant un individu, mais un autre peut décider d'employer le terme « automobile » comme nom alternatif. Il est principalement utilisé pour l'annotation et la récupération dans différents domaines, Il est également utilisé pour fonder d'autres vocabulaires tels que le schéma FOAF, en tant que connaissances de base des outils d'alignement d'ontologies et d'autres applications [33].

## 2.5) Opérations sur les ontologies

Plusieurs opérations peuvent être appliquées aux ontologies pendant le processus d'alignement, notamment dans les grands projets où il est possible d'élargir une ontologie en la combinant avec une ou plusieurs autres ontologies, comme on peut le faire juste avec un ensemble partiel de concepts, afin de créer un modèle de données selon le cas d'utilisation exigé. Ceci est l'ensemble d'opérations qu'on peut avoir : la fusion, l'unification, aussi l'héritage, le mappage, le raffinement et l'intégration. D'autres sont faites pour comparer entre les ontologies, nous citons : l'extension, l'équivalence, aussi l'identité, celle qui est fortement traduisible ou faiblement traduisible et approximativement traduisible.

## 2.6) Conclusion

Comme nous l'avons vu dans ce chapitre, la solution au problème d'hétérogénéité des ontologies semble se situer dans le développement des méthodes d'alignement. D'une part, décrire les ressources Web sous forme d'ontologie est une solution inévitable au problème de l'hétérogénéité. D'autre part, l'ontologie présente également le même problème vu de leurs différences. En conséquence, les méthodes d'alignements ne couvrent que certains aspects pouvant être exprimés dans une ontologie [32].

# Conception et implémentation

## 3.1) Introduction

## 3.2) Approche proposée

## 3.3) Conception de l'application

## 3.4) Implémentation et réalisation

## 3.5) Conclusion

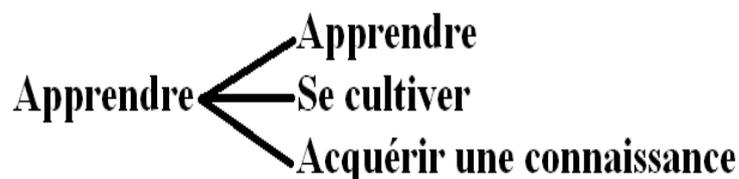
### 3.1) Introduction

Ce troisième chapitre est consacré pour détailler l'approche que nous avons proposé et qui vise à réduire considérablement les sources d'hétérogénéité. Nous allons voir ainsi la mise en œuvre du principe à travers la conception et l'implémentation d'une application, également identifier toutes les technologies exploitées pendant le développement et finir avec une conclusion.

### 3.2) Approche proposée

Nous rappelons que pour obtenir une interprétation similaire d'une ontologie par de nombreux utilisateurs et machines différentes, ce qui signifie une interprétation unique et similaire de la sémantique des données qu'elle contient, cela ne peut être fait qu'en éliminant les sources d'hétérogénéité qui se produisent lors de l'utilisation d'ontologies. La méthode que nous proposons a pour objectif de réduire l'ambiguïté causée par les diverses entités utilisées dans les ontologies, car chacune d'elles joue un rôle important dans la définition de la vraie signification des connaissances communiquées entre machines.

Notre point de vue sur l'échange d'informations entre les machines est basé sur les mêmes principes d'échange de dialogue entre les humains. Pour que deux personnes s'entendent sur une signification spécifique, elles peuvent utiliser le même concept commun et précédemment acquis qui fait référence à cette signification, comme elles peuvent également utiliser des concepts synonymes pour dire la même chose, ou bien utiliser d'autres concepts et les relier avec des règles logiques pré-apprises pour déduire la signification souhaitée.



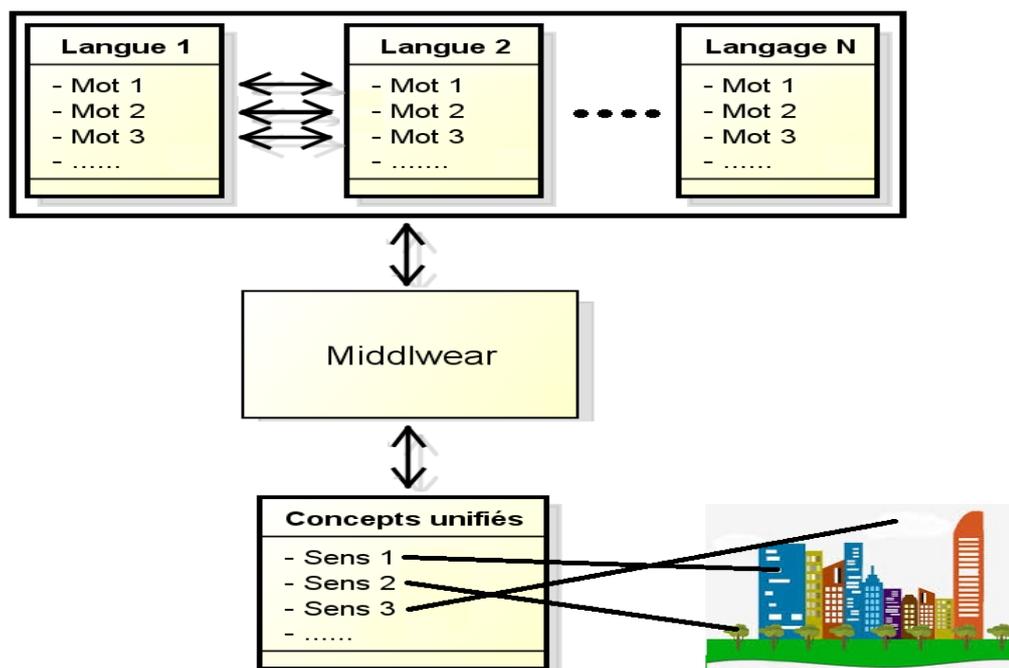
D'après ce que nous constatons, l'accord intervenu entre les deux parties résulte de l'échange d'une signification commune. Dans le premier cas, il y a eu échange d'une seule signification en utilisant un et un seul concept, et bien que les deux concepts diffèrent dans le second cas toujours c'est la même signification qui est échangée. Parce qu'en réalité à un certain niveau d'abstraction, c'est les significations qui sont échangées et le rôle d'un concept n'est que de le traduire selon un contexte donné à la signification qui lui est nécessaire. Ce qui renforce ce que nous disons, c'est l'accord qui s'est produit dans le troisième cas où une signification a été échangée sans utiliser un concept similaire en retour.

À travers ce que nous avons vu de l'échange de significations pendant le dialogue humain, nous voulions matérialiser la même chose lors de l'échange d'informations entre machines, et nous pensons donc qu'il est très utile que les significations elles-mêmes soient stockées sous forme de données dans les ontologies plutôt que d'être remplies de mots potentiels pour plusieurs significations.



Faire en sorte que les machines traitent la signification des données leur rapprochera la vraie signification, ce qui leur donnera une plus grande capacité de compréhension et une définition plus claire et plus précise de ce que sont les données, ce qui facilitera leur processus de traitement et fournira une interopérabilité complète. Ceci en dépit du fait qu'il existe de nombreuses perceptions humaines sur les phénomènes de toute nature, mais les significations elles-mêmes sont partagées par tous.

À partir d'ici, nous proposons la solution de concevoir un mécanisme unifié, un dictionnaire par exemple, qui inclut toutes les significations communes aux humains et définies avec précision, tout en identifiant les relations qui peuvent relier entre elles, ainsi que ces significations deviennent des données pour remplir les ontologies au début de leur conception. Travailler de cette manière nécessitera inévitablement le développement d'un intermédiaire permettant le traitement d'ontologies pleines de significations diverses qui peuvent également être traduites dans n'importe quelle langue naturelle comme le montre la figure (12) ci-dessous.



**Figure 12** – Schémat montrant comment choisir un mot spécifique à un sens unifié.

Lors de la conception de toute nouvelle ontologie et afin d'y saisir des données, nous n'avons qu'à choisir le mot ou le groupe de mots qui exprime la signification voulue selon le langage naturel spécifié, à partir duquel toutes les données utilisées dans toutes les ontologies sur le Web sémantique deviennent universelles et compatibles. Nous pensons que cette méthode réduira considérablement l'écart en termes d'incompatibilité et d'incohérence entre ontologies.

### 3.3) Conception de l'application

Dans le cadre de la mise en œuvre de notre approche, nous avons appliqué cette approche sur un domaine en développant une méthode qui fait correspondre toutes les données exploitées à un codage unifié. Afin de démontrer la faisabilité et l'utilité de notre approche, nous avons créé une application qui manipule deux ontologies avec des données écrites dans différentes langues (Anglais, Français). L'ensemble des données que nous avons choisi pour alimenter ces ontologies est un groupe d'entités que nous retrouvons dans les relations familiales, et nous les avons nommées "FamilleAllemande.owl"

et "FamilleFrançaise.owl".

### 3.3.1) Conception d'un dictionnaire unifié

Initialement, nous avons défini les différentes classes, sous classes, propriétés qui composent les ontologies. Après avoir collecté les données, nous avons entrepris de normaliser la signification de chaque entité proposée en donnant pour chaque une un identifiant unique. Ainsi, le tableau (1) suivant qui représente le dictionnaire unifié, affiche les entités et leurs identifiants :

| Les entités             | En français  | En anglais   | En allemand | L'ID |
|-------------------------|--------------|--------------|-------------|------|
| <b>Les classes</b>      | Personne     | Human        | Mensch      | c1   |
|                         | Individu     | Individual   | Individuum  | c2   |
|                         | Fraterie     | Fratery      | Geschwister | c3   |
|                         | Homme        | Man          | Mann        | c4   |
|                         | Femme        | Women        | Frau        | c5   |
|                         | Enfant       | Child        | Kind        | c6   |
|                         | Grand parent | Grand parent | Grobeltern  | c7   |
|                         | Parent       | Parent       | Eltern      | c8   |
|                         | Masculin     | Male         | Männliches  | c9   |
|                         | Feminin      | Female       | Weibliche   | c10  |
| <b>Les sous classes</b> | Frère        | Brother      | Bruder      | sc1  |
|                         | Soeur        | Sister       | Schwester   | sc2  |
|                         | Fils         | Son          | Sohn        | sc3  |
|                         | Fille        | Daughter     | Tochter     | sc4  |
|                         | Grand mère   | Grand mother | Grobmutter  | sc5  |
|                         | Grand père   | Grand father | Grobvater   | sc6  |
|                         | Mère         | Mother       | Mutter      | sc7  |
|                         | Père         | Father       | Vater       | sc8  |

| Les entités                         | En français                      | En anglais                      | En allemand                | L'ID |
|-------------------------------------|----------------------------------|---------------------------------|----------------------------|------|
| <b>Les propriétés d'objets</b>      | Se marier avec                   | To get married with             | Zu heiraten                | po1  |
|                                     | Est parent de                    | Is parent of                    | Ist eltern von             | po2  |
|                                     | Est en relation de fraterie avec | Is en relation of fraterie with | Ist in einer beziehung mit | po3  |
|                                     | Est enfant de                    | Is child of                     | Ist kind von               | po4  |
| <b>Les sous propriétés d'objets</b> | Est père de                      | Is father of                    | Ist vater von              | spo1 |
|                                     | Est mère de                      | Is mother of                    | Ist mutter von             | spo2 |
|                                     | Est frère de                     | Is brother of                   | Ist bruder von             | spo3 |
|                                     | Est soeur de                     | Is sister of                    | Ist schwester<br>De        | spo4 |
|                                     | Est fils de                      | Is son of                       | Ist sohn von               | spo5 |
|                                     | Est fille de                     | Is girl of                      | Ist mädchen<br>von         | spo6 |
| <b>Les propriétés de données</b>    | Nom                              | First name                      | Name                       | pd1  |
|                                     | Age                              | Age                             | Alter                      | pd2  |
|                                     | Nationalité                      | Nationality                     | Nationalitat               | pd3  |
| <b>Les types de données</b>         | Algérienne                       | Algerian                        | Algerische                 | td1  |
|                                     | Marocainne                       | Moroccan                        | Marokkanische              | td2  |

Tableau 1 – L'ensemble des entités en langues différentes avec leurs identifiants.

### 3.3.2) Création des ontologies

Dans ce qui suit, deux tableaux et deux images représentent respectivement les instances et les hiérarchies suggérées pour les deux ontologies. Les ontologies sont différentes les unes des autres dans leur structure hiérarchique, en plus de la différence dans le nombre et les significations de certains concepts où nous avons utilisé le mot "Personne", "Homme" et "Femme" dans l'ontologie "FamilleAllemande.owl" qui sont remplacés par "Individu", "Masculin" et "Féminin" dans l'ontologie "FamilleFrançaise.owl".

- L'ontologie "FamilleFrançaise.owl" :

| La classe       | Nom     | Age | Nationalité | Relations                      |
|-----------------|---------|-----|-------------|--------------------------------|
| <b>Masculin</b> | Mohamed | 70  | Algérienne  | Mohamed se marie avec Aicha    |
|                 | Omar    | 40  | Algérienne  | Omar est fils de Mohamed       |
|                 | Ali     | 38  | Algérienne  | Ali est fils de Mohamed        |
|                 | Khaled  | 15  | Marocaine   | Khaled se marie avec Amani     |
|                 | Farid   | 10  | Marocaine   | Farid est fils de Khaled       |
|                 | Ziad    | 10  | Algérienne  | Ziad est fils d'Omar et Manel  |
|                 | Jamel   | 5   | Algérienne  | Jamel est fils d'Omar et Manel |
| <b>Féminin</b>  | Aicha   | 69  | Algérienne  | Aicha est la mère de Fatima    |
|                 | Sonia   | 30  | Algérienne  | Sonia est fille de Aicha       |
|                 | Fatima  | 18  | Algérienne  | Fatima est fille de Aicha      |
|                 | Amani   | 5   | Algérienne  | Amani est fille de Sonia       |
|                 | Manel   | 25  | Marocaine   | Manel se marie avec Omar       |

Tableau 2 – Les instances de l'ontologie "FamilleFrançaise.owl".

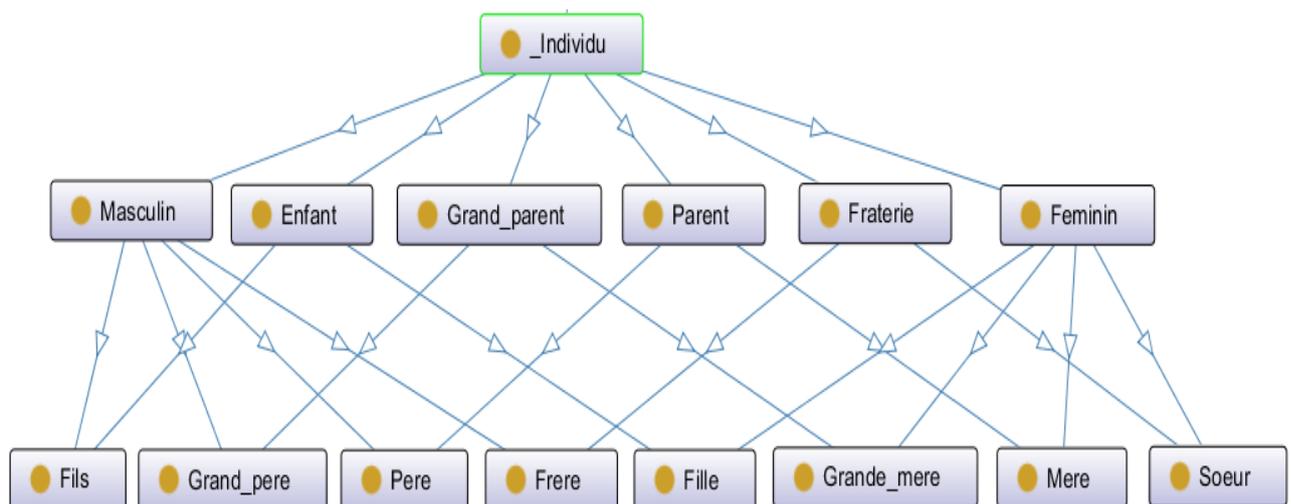


Figure 13 – Hiérarchie des concepts d'ontologie "FamilleFrançaise.owl".

- L'ontologie "FamilieAllemande.owl" :

| La classe  | La sous classe | Nom     | Age | Nationalité   | Relations                    |
|------------|----------------|---------|-----|---------------|------------------------------|
| Mannliches | Großvater      | Ibrahim | 80  | Marokkanische | Ibrahim zu heiraten Hadjer   |
|            |                | Ismail  | 10  | Marokkanische | Ismail ist sohn von Ibrahim  |
|            | Vater          | Isak    | 69  | Marokkanische | Izak ist sohn von Ibrahim    |
|            |                | Khaled  | 15  | Marokkanische | Khaled ist sohn von Ismail   |
|            | Sohn           | Farid   | 10  | Marokkansche  | Farid ist sohn von Khaled    |
|            |                | Moussa  | 18  | Marokkanische | Moussa ist sohn von Izak     |
| Weibliche  | Großmutter     | Hadjer  | 60  | Marokkanische | Hadjer ist mutter von Ismail |
|            | Mutter         | Safia   | 56  | Marokkanische | Safia zu heiraten Izak       |
|            |                | Somaia  | 69  | Marokkanische | Somaia zu heiraten Izak      |
|            | Tochter        | Manel   | 25  | Marokkanische | Manel ist madchen von Ismail |

Tableau 3 – Les instances de l'ontologie "FamilieAllemande.owl".

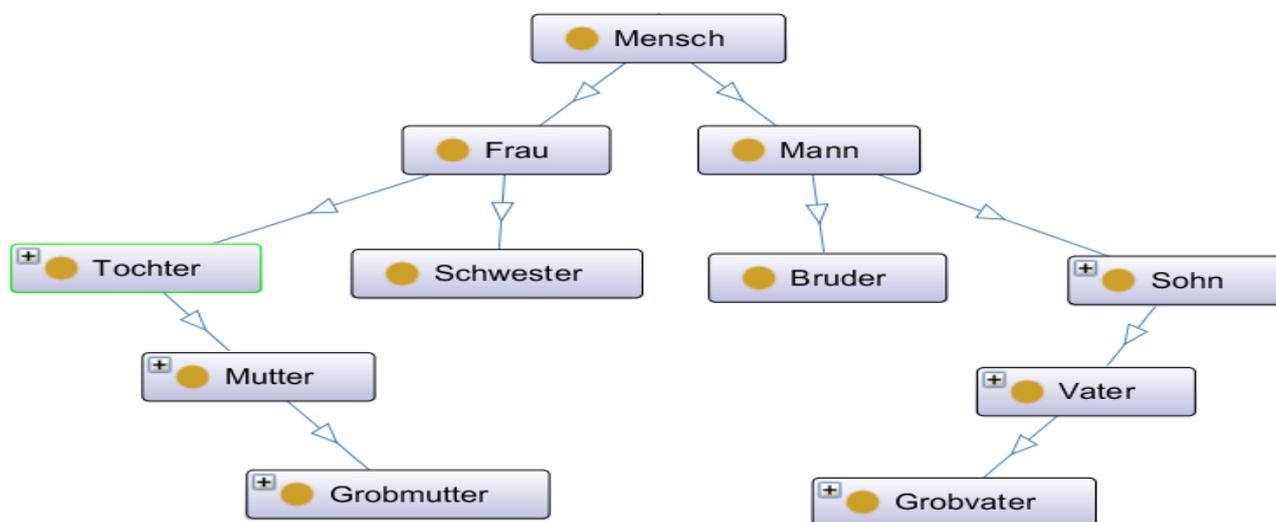


Figure 14 – Hiérarchie des concepts d'ontologie "FamilieAllemande.owl".

### 3.3.3) Fonctionnalités de l'application

Dans ce qui suit nous abordé les différentes fonctionnalités par lesquels l'application nous permet de manipuler les ontologies. Ces fonctionnalités sont :

- **Mise à jour des individus** : nous pouvons ajouter des nouvelles personnes aux ontologies, comme nous avons la possibilité de les modifier ou les supprimer.
- **Manipulation des ontologies** : en ce qui concerne leur création, modification et leur suppression.
- **Effectuer des requêtes** : le système nous donne la permission de consulter les données d'une ontologie choisie. De plus, le résultat de sortie peut être compatible avec les requêtes selon le choix de paramètres et de langue naturelle requis.
- **Opérations sur les ontologies** : Egalement, le système est susceptible de construire une ontologie en réalisant les opérations Union, Différence où Intersection de deux autres ontologies. Cela nous rend capable de récolter les connaissances qui existent dans leur intégralité.

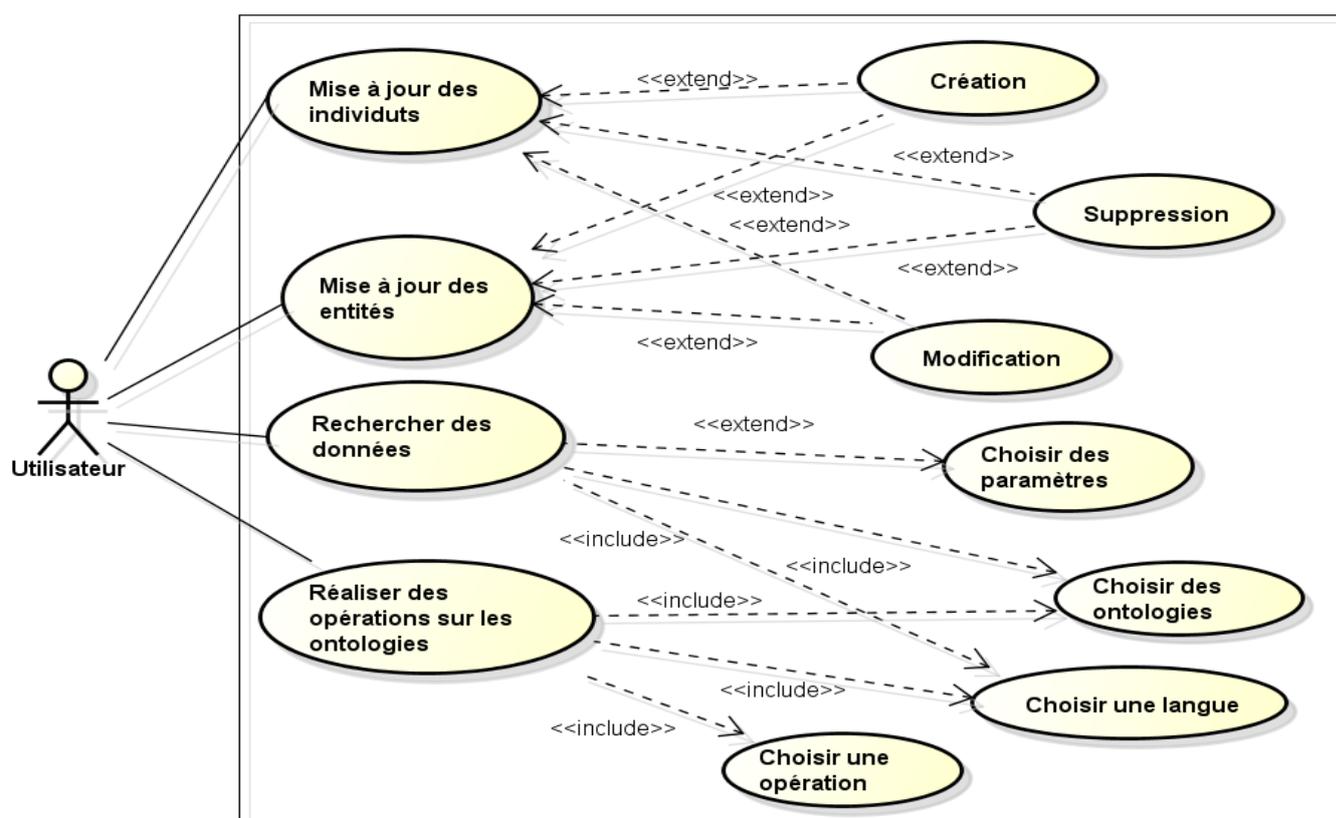


Figure 15 – Fonctionnalités de l'application.

### 3.3.4) Architecture de l'application

La figure (16) illustre l'architecture de l'application mise en œuvre. Elle décrit ses composants et les différentes interactions qui se déroulent entre eux. Les parties développées dans le système sont : le middleware, les ontologies, les dictionnaires, les règles d'inférences et l'interface.

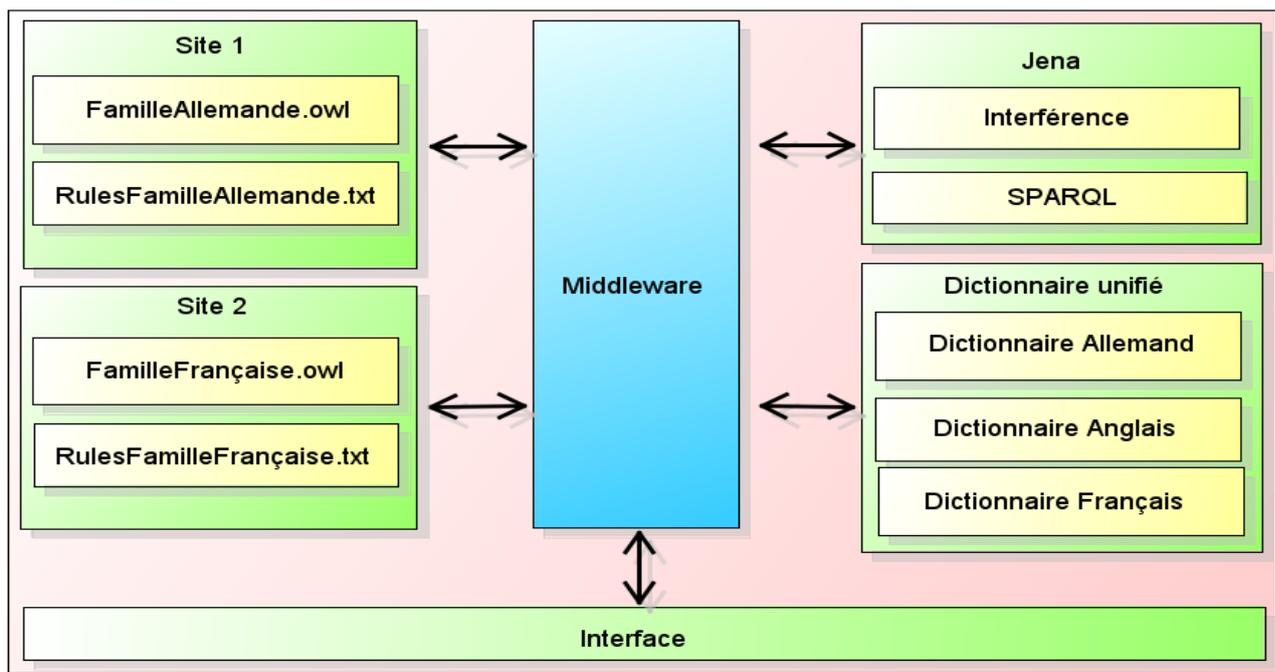


Figure 16 – L'architecture de l'application.

### 3.4) Implémentation et réalisation

Pour l'implémentation de l'application, nous avons opté pour un ensemble de technologies de programmation Open source et graphiques, nous les citons dans ce que suit :

#### 3.4.1) Les outils utilisés

Ceci est l'ensemble des outils de programmations que nous avons utilisés afin d'atteindre notre objectif :

### 3.4.1.1) Prototégé 5.5.0

Nous avons créé les ontologies en utilisant Protégé 5.5.0. C'est une plateforme libre, open-source qui a un écran distinct et facile à utiliser. Il est riche en structures et fonctions qui prennent en charge tous les processus pouvant être exécutés sur les ontologies dans différents formats de description des connaissances.

### 3.4.1.2) Langage OWL

Nous avons utilisé le langage OWL en raison de la grande précision en terme d'expressivité des connaissances.

### 3.4.1.3) Netbeans 8.0.2

L'environnement de développement NetBeans est un EDI gratuit permettant aux programmeurs d'écrire, de compiler, de déboguer et de déployer des programmes, il peut supporter plusieurs langages de programmation.

### 3.4.1.4) Langage JAVA

Java est un langage de programmation orienté objet, développé par Sun Microsystems. Il donne la possibilité de développer des programmes pour plusieurs systèmes d'exploitation (Windows, Linux, Macintosh, Solaris), tout en créant des logiciels qui s'exécutent en desktop, téléphones portables, comme aussi, il peut être utilisé sur Internet intégré dans les pages web (applet), ou encore comme langage serveur (jsp).

### 3.4.1.5) Langage SPARQL

Comme nous l'avons abordé dans le premier chapitre, SPARQL est, en même temps, un protocole et un langage d'interrogation des bases de connaissances alimentées en graphes RDF.

### 3.4.1.6) Langage de règles d'inférence Jena

C'est un sous-système d'inférence de l'api Jena, il est conçu pour permettre à une gamme de moteurs d'inférence ou de raisonneurs d'être connectés à Jena. De tels moteurs sont utilisés pour dériver des assertions RDF supplémentaires provenant d'une base RDF, ainsi que des informations éventuelles sur l'ontologie, les axiomes et les règles associés au raisonneur [34]. La figure (17) qui suit représente un ensemble de règles proposées pour faire des inférences utiles aux besoins des requêtes introduites.

```

1  @include <OWLMicro>.
2  @prefix ns: <http://ontologies/Famille#>.
3  @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
4  #      B enfant de A      B se marie avec C      C -> femme      ---> C -> epouse de son fils
5  [rule1: (?B ns:spo4 ?A) (?B ns:po1 ?C) (?C rdf:type ns:c5) -> (?C rdf:type ns:c11)]
6  #      B fils de A      --->      B -> enfant      B enfant de A
7  [rule2: (?B ns:spo5 ?A) -> (?B rdf:type ns:c6) (?B ns:po4 ?A)]
8  #      B fille de A      --->      B -> enfant      B enfant de A
9  [rule3: (?B ns:spo6 ?A) -> (?B rdf:type ns:c6) (?B ns:po4 ?A)]
10 #      B enfant de A      --->      B -> enfant
11 [rule4: (?B ns:po4 ?A) -> (?B rdf:type ns:c6)]
12 #      B enfant de A      --->      B -> parent
13 [rule5: (?B ns:po4 ?A) -> (?A rdf:type ns:c8)]
14 #      A enfant de B      B enfant de C ---> C -> grand parent
15 [rule6: (?A ns:po4 ?B) (?B ns:po4 ?C) -> (?C rdf:type ns:c7)]
16 #      AC -> grand parent      B -> femme      ---> C -> grande mere
17 [rule7: (?C rdf:type ns:c7) (?C rdf:type ns:c5) -> (?C rdf:type ns:sc5)]
18 #      C -> grand parent      C -> homme      ---> C -> grand pere
19 [rule8: (?C rdf:type ns:c7) (?C rdf:type ns:c4) -> (?C rdf:type ns:sc6)]
20 #      A fille de B      B -> homme      -----> B -> pere
21 [rule9: (?A ns:spo6 ?B) (?B rdf:type ns:c9) -> (?B rdf:type ns:sc8)]
22 #      K fils de F      F -> homme      -----> F -> pere
23 [rule10: (?k ns:spo5 ?F) (?F rdf:type ns:c9) -> (?F rdf:type ns:sc8)]
24 #      A fille de B      B -> femme      -----> B -> mere
25 [rule11: (?A ns:spo6 ?B) (?B rdf:type ns:c10) -> (?B rdf:type ns:sc7)]
26 #      K fils de F      F -> homme      -----> F -> mere
27 [rule12: (?k ns:spo5 ?F) (?F rdf:type ns:c10) -> (?F rdf:type ns:sc7)]

```

Figure 17 – Règles d'inférences programmées pour l'application.

### 3.4.2) Interfaces et fonctionnalités

Dans cette section, nous présentons quelques images que nous avons prises pour les interfaces de l'application développées avec certaines fonctions qui peuvent être exécutées, car nous avons pu incarner notre méthode. Dans ce

programme, nous avons réalisé deux interfaces qui sont :

### - Interface "Affichage" :

Celle-là concerne l'affichage des données en l'interrogeant avec diverses requêtes (paramètres, non-paramètres). Ici nous pouvons observer l'incarnation de notre idée où nous pouvons obtenir des résultats quelle que soit la différence entre le langage de requête et le langage des données d'ontologies.

The screenshot shows the 'Application Sémantique' interface. At the top, there are tabs for 'Affichage', 'Opérations', 'Famille AL', and 'Famille FR'. The main interface is divided into several sections:

- Choisir :**
  - Une ontologie : Famille FR (dropdown)
  - Une langue :  Allemand  Anglais  Français
- Choisir une requête :**
  - Mensch (dropdown)
  - Name\_  Alter  Nationalitat
  - Aucun (dropdown) | 0 (dropdown) | Aucune (dropdown)
- Signification de la requête :**
  - Lister le(s) [Name\_, Alter, Nationalitat] de(s) Mensch
- Paramètres :**
  - DISTINCT  LIMIT
  - ORDER BY: Name\_ (dropdown)
  - ASC  DESC
- Le résultat de la recherche :**

|          |        |                 |
|----------|--------|-----------------|
| "Manel"  | "25"^^ | "Marokkanische" |
| "Ismail" | "47"^^ | "Marokkanische" |
| "Isak"   | "37"^^ | "Marokkanische" |
| "Somaia" | "33"^^ | "Marokkanische" |
| "Hadjer" | "60"^^ | "Marokkanische" |
| "Safia"  | "56"^^ | "Marokkanische" |
| "Moussa" | "18"^^ | "Marokkanische" |
| "Khaled" | "35"^^ | "Marokkanische" |
- La requête au format SPARQL :**

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX ns: <http://ontologies/Famille#>
SELECT ?Name_?Alter ?Nationalitat
WHERE {
?Mensch rdf:type ns:Mensch.
?Mensch ns:Name_ ?Name_.
?Mensch ns:Alter ?Alter.
?Mensch ns:Nationalitat ?Nationalitat.
}
```
- Buttons:** Rechercher, Opération,  Requête manuelle

Figure 18 – Interface "Affichage".

Les figures (19,20,21) montrent comment fonctionne l'application dans trois langues différentes et obtenir le résultat de la demande envoyée. Ainsi, avec l'approche proposée nous pouvons traiter n'importe quel type de base de connaissances, même si elles sont différentes dans ses modèles ainsi que

dans le langage de données codées, nous pouvons simplement accéder à ces données et les exploiter. Dans l'exemple suivant nous avons interrogé l'ontologie "FamilleAllemande.owl" avec différentes requêtes qui demandent de lister le nom, l'âge et la nationalité de toutes les grandes mères. Les requêtes sont générées selon la langue utilisée.

```

1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX ns: <http://ontologies/Famille#>
3 SELECT ?Name ?Alter ?Nationalitat
4 WHERE {
5   ?Grobmutter rdf:type ns:Grobmutter.
6   ?Grobmutter ns:Name ?Name.
7   ?Grobmutter ns:Alter ?Alter.
8   ?Grobmutter ns:Nationalitat ?Nationalitat.
9 }

```

| Name     | Alter  | Nationalitat    |
|----------|--------|-----------------|
| "Safia"  | "56"^^ | "Marokkanische" |
| "Hadjer" | "60"^^ | "Marokkanische" |

Figure 19 – Une requête SPARQL et son résultat obtenu en allemand.

```

1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX ns: <http://ontologies/Famille#>
3 SELECT ?First_name ?Age ?Nationality
4 WHERE {
5   ?Grand_mother rdf:type ns:Grand_mother.
6   ?Grand_mother ns:First_name ?First_name.
7   ?Grand_mother ns:Age ?Age.
8   ?Grand_mother ns:Nationality ?Nationality.
9 }

```

| First_name | Age    | Nationality |
|------------|--------|-------------|
| "Safia"    | "56"^^ | "Moroccan"  |
| "Hadjer"   | "60"^^ | "Moroccan"  |

Figure 20 – Une requête SPARQL et son résultat obtenu en anglais.

```

1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX ns: <http://ontologies/Famille#>
3 SELECT ?Nom ?Age ?Nationalite
4 WHERE {
5   ?Grande_mere rdf:type ns:Grande_mere.
6   ?Grande_mere ns:Nom ?Nom.
7   ?Grande_mere ns:Age ?Age.
8   ?Grande_mere ns:Nationalite ?Nationalite.
9 }

```

| Nom      | Age    | Nationalite |
|----------|--------|-------------|
| "Safia"  | "56"^^ | "Marocaine" |
| "Hadjer" | "60"^^ | "Marocaine" |

Figure 21 – Une requête SPARQL et son résultat obtenu en français.

Ce qui nous fait croire que notre approche est si importante pour fonctionner

en douceur et efficacement dans le Web sémantique, elle nous permet non seulement de traiter les données des bases de connaissances qui sont souvent différentes, plus que cela, en utilisant notre connaissance des significations des données et des relations entre elles avec précision, nous pouvons traiter même de nouvelles informations cachées dans les ontologies grâce aux règles d'inférence. Nous offrons un exemple concret de cela, on voit que dans l'ontologie "FamilleAllemande.owl", il n'y a pas de concept de "Enfant", de "Grand-parent" ou de "Parent" mais le fait que les données ont un codage global, on peut former des règles d'inférence dans n'importe quelle langue que nous voulons et afficher ces concepts avec facilité.

```

6 #      B fils de A      --->    B -> enfant      B enfant de A
7 [rule2: (?B ns:spo5 ?A) -> (?B rdf:type ns:c6) (?B ns:po4 ?A)]
8 #      B fille de A    --->    B -> enfant      B enfant de A
9 [rule3: (?B ns:spo6 ?A) -> (?B rdf:type ns:c6) (?B ns:po4 ?A)]
10 #     B enfant de A   --->    B -> enfant
11 [rule4: (?B ns:po4 ?A) -> (?B rdf:type ns:c6)]

```

Figure 22 – Règles d'inférence ajoutées pour représenter l'entité "Enfant".

La requête suivante demande d'afficher le nom, l'âge et la nationalité des enfants :

|  | Nom      | Age    | Nationalite |
|--|----------|--------|-------------|
| 1 SELECT ?Nom ?Age ?Nationalite        | "Moussa" | "18"^^ | "Marocaine" |
| 2 WHERE {                              | "Farid"  | "1"^^  | "Marocaine" |
| 3 ?Enfant rdf:type ns:Enfant.          | "Isak"   | "37"^^ | "Marocaine" |
| 4 ?Enfant ns:Nom ?Nom.                 | "Ismail" | "47"^^ | "Marocaine" |
| 5 ?Enfant ns:Age ?Age.                 | "Khaled" | "35"^^ | "Marocaine" |
| 6 ?Enfant ns:Nationalite ?Nationalite. | "Manel"  | "25"^^ | "Marocaine" |
| 7 }                                    |          |        |             |

Figure 23 – La requête SPARQL pour afficher les enfants ainsi que son résultat.

## - Interface "Opérations" :

Grâce à cette interface, nous pouvons exécuter des opérations sur les ontologies en les fusionnant selon la fonction (Union, Différence, Intersection) et la langue (Allemand, Anglais, Français) que nous choisissons.

**Application Sémantique**

Affichage Opérations Famille AL Famille FR

**Choisir :**

Une ontologie :  Famille AL  Famille FR

Une opération :  Union  Différence  Intersection

Une langue :  Allemand  Anglais  Français

Nom de l'ontologie :

**Signification de la requête :**

Lister le(s) [Name\_, Alter, Nationalitat] de(s) Mensch

**Paramètres :**

DISTINCT  LIMIT  ASC  DESC

ORDER BY

**Le résultat de la recherche :**

| Name_     | Alter  | Nationalitat    |
|-----------|--------|-----------------|
| "Ibrahim" | "80"^^ | "Marokkanische" |
| "Farid"   | "1"^^  | "Marokkanische" |
| "Manel"   | "25"^^ | "Marokkanische" |
| "Ismail"  | "47"^^ | "Marokkanische" |
| "Isak"    | "37"^^ | "Marokkanische" |
| "Somaia"  | "33"^^ | "Marokkanische" |
| "Hadjer"  | "60"^^ | "Marokkanische" |

**La requête au format SPARQL :**

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX ns: <http://ontologies/Famille#>
SELECT ?Name_ ?Alter ?Nationalitat
WHERE {
?Mensch rdf:type ns:Mensch.
?Mensch ns:Name_ ?Name_.
?Mensch ns:Alter ?Alter.
?Mensch ns:Nationalitat ?Nationalitat.
}
```

Rechercher

Opération

Requête manuelle

Figure 24 – Interface "Opérations".

Notons que la différence de langage de données dans les deux ontologies conçues n'est pas un problème fondamental avec la façon dont laquelle nous procédons. La figure qui suit est le résultat de l'opération "Union" sur les deux ontologies "FamilleAllemande.owl" et "FamilleFrançaise.owl" avec le choix de la langue française pour l'encodage des données.

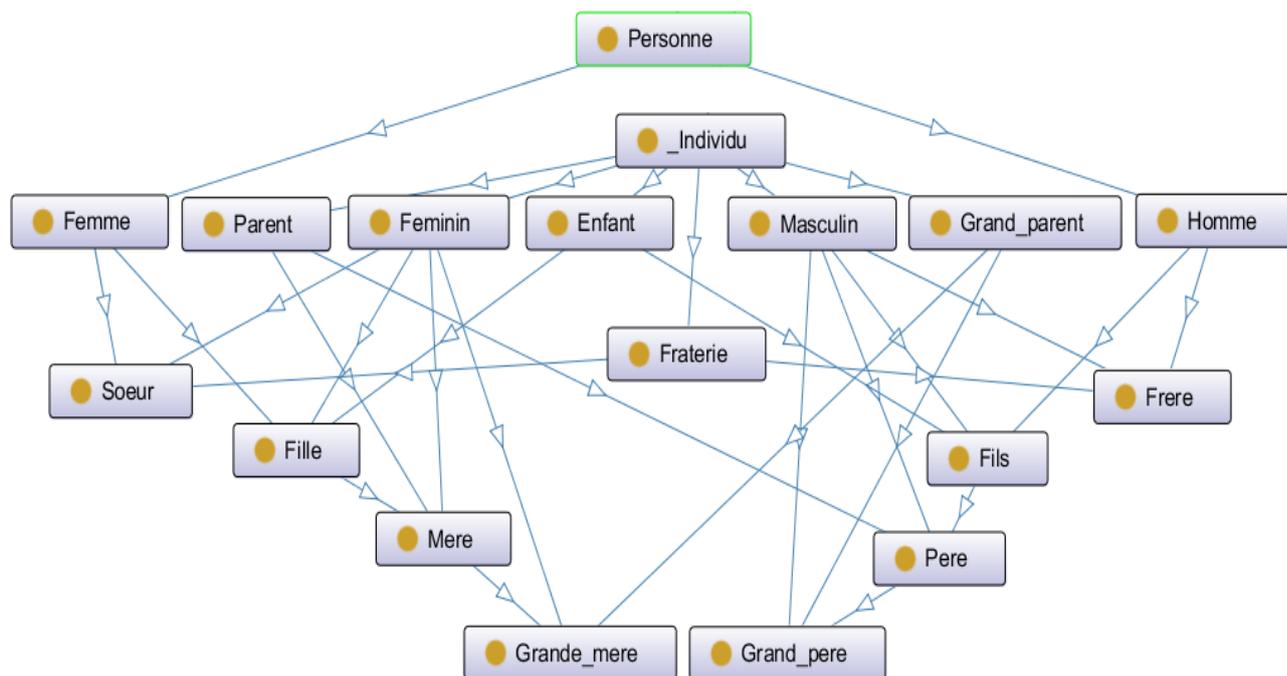


Figure 25 – Hiérarchie des concepts des deux ontologies obtenue avec l'opération "Union".

Dans cette nouvelle ontologie "Union.owl", on voit que la conception de ces entités est cohérente et logique en raison de la présence d'harmonie et de similitude dans leur signification, même la différence dans le vrai sens de certaines entités, cela n'a pas posé de problème dans la logique générale de l'ontologie, et nous avons testé l'ontologie avec toutes les requêtes et les résultats étaient tous corrects.

Bien que les deux requêtes suivantes portent sur deux entités différentes en signification mais nous avons pu obtenir le même résultat, à l'exception de l'entité "Homme", l'application n'affiche que ces instances sans ajouter celles de l'entité "Masculin" car il n'y a pas de lien entre les deux entités (même chose pour l'entité "Femme" avec l'entité "Féminin"). Et comme nous l'avons dit plus tôt, on peut toujours utiliser notre connaissance préalable de la signification des données, et donc il est possible de mettre des algorithmes reliant une entité particulière à toutes les entités qui peuvent avoir un sens proche d'elle selon leur signification.

Donc nous avons remédié au problème manuellement et nous avons ajouté deux règles d'inférence qui sont :

```

1 #          B -> Masculin  --->  B -> Homme
2 [rule22:  (?B rdf:type ns:c9) -> (?B rdf:type ns:c4)]
3 #          B -> Féminin  --->  B -> Femme
4 [rule21:  (?B rdf:type ns:c10) -> (?B rdf:type ns:c5)]

```

Avec ces deux règles, maintenant nous pouvons dire qu'on peut répondre à toute requête interrogeant l'ontologie "Union.owl".

|  |   |
|--|---|
| <pre> 1 SELECT ?Nom ?Age ?Nationalite 2 WHERE { 3     ?Homme rdf:type ns:Homme. 4     ?Homme ns:Nom ?Nom. 5     ?Homme ns:Age ?Age. 6     ?Homme ns:Nationalite ?Nationalite. 7 } </pre> | <pre> 1 SELECT ?Nom ?Age ?Nationalite 2 WHERE { 3     ?Masculin rdf:type ns:Masculin. 4     ?Masculin ns:Nom ?Nom. 5     ?Masculin ns:Age ?Age. 6     ?Masculin ns:Nationalite ?Nationalite. 7 } </pre> |
|--|---|

Figure 26 – Les requêtes SPARQL pour afficher les hommes et les masculins.

Ainsi que le résultat des deux requêtes est le même ce qui obtenu dans la figure au dessous :

| Nom       | Age  | Nationalite  |
|-----------|------|--------------|
| "Mohamed" | "70" | "Algerienne" |
| "Ibrahim" | "80" | "Marocaine"  |
| "Ismail"  | "47" | "Marocaine"  |
| "Omar"    | "40" | "Algerienne" |
| "Khaled"  | "35" | "Marocaine"  |
| "Isak"    | "37" | "Marocaine"  |
| "Ziad"    | "10" | "Algerienne" |
| "Jamel"   | "5"  | "Algerienne" |
| "Moussa"  | "18" | "Marocaine"  |
| "Farid"   | "1"  | "Marocaine"  |

### 3.5) Conclusion

Notre travail vise à présenter une solution qui permet de réduire les sources d'hétérogénéité à tous les niveaux afin d'assurer des travaux plus efficaces et plus faciles entre les machines. L'application grâce à laquelle nous avons pu incarner notre vision, nous a permis de rechercher et d'afficher les données des ontologies "FamilleAllemande.owl» et "FamilleFrançaise.owl" selon trois langues (allemand, anglais et français). Ceci sans recourir à la traduction ou à l'alignement basés sur le calcul de la similarité, mais à travers un dictionnaire unifié pour se mapper entre les entités des deux ontologies différentes.

La conception d'otologies à l'aide de données globales dont les significations ont été déterminées à l'avance donne une vision plus claire des ontologies et deviennent compréhensibles avec une grande précision, dans la mesure où on peut même déduire le point de vue pour lequel ces ontologies ont été conçues facilement, tant qu'il est possible de connaître toutes leurs entités et comment les traiter avec précision, et c'est grâce aux règles d'inférence qui peuvent également être échangées entre machines. En outre, il est même possible d'établir des règles strictes uniformes pour la conception des ontologies et le type de relations qui peuvent exister entre leurs entités, de sorte qu'il ne se produise pas, par exemple, qu'un concept soit placé au-dessus d'un autre concept plus général que lui.

# Conclusion générale

Grâce au Web sémantique et les données liées, par l'annotation de définitions et de détails échangées, les machines bénéficient plus de coopération, d'évolution dynamique et avoir plus de détails sur la description des ressources (matérielles ou logiciels) qu'elles interagissent avec elles, ouvrant la voie à de nouvelles fonctionnalités considérées déjà comme impossibles.

C'est dans ce sens, nous avons partagé notre méthode décrite précédemment, et nous pensons qu'elle peut gagner plus de temps, d'efficacité et de précision pour ce domaine. Bien sûr, il reste beaucoup de travail et de tests autour de notre méthode, et des recherches pour découvrir dans quelle mesure elles peuvent être exploitées à grande échelle au sein du réseau sémantique, nous ne considérons donc pas que cela met fin, comme par exemple à la traduction ou à l'alignement des ontologies, mais cela peut être considérablement réduit.

Le Web sémantique a tendance de s'intégrer à tous les objets autour de nous (montres, salles, lunettes, voiture, télé, abris bus, etc.). Aujourd'hui il a dépassé sa forme initiale d'un espace documentaire, offre et lie maintenant tous types de ressources numériques (données, services) et non numériques (personnes, lieux, évènements, entreprises, etc.). Cette évolution permet toujours plus d'interconnexion et d'intégration à notre quotidien. [...] Plus il s'enrichit et plus il est attractif pour de nouveaux objets et applications connectés. Et plus il attire d'objets et d'applications qui produisent et consomment données et services et plus il devient attractif [28].

# Références

- [1] Miltiadis Lytras, Miguel-Angel Sicilia, John Davies et Vipul Kashyap, Emerald, "Semantic Web technologies for digital libraries", V26, Issue 4/5, p191, 2005.
- [2] <https://www.inria.fr/centre/grenoble/actualites/l-emergence-d-un-web-des-connaissances>, Publié le : 25 Septembre 2012, Consulté le : 05 Juin 2019.
- [3] Miltiadis Lytras, Miguel-Angel Sicilia, John Davies et Vipul Kashyap, Emerald, "Semantic Web technologies for digital libraries", V26, Issue 4/5, p190, 2005.
- [4] Allan Brimicombe et Chao Li, *Location-Based Services and Geo-Information Engineering*, p 26, Wiley, 2009.
- [5] Tim Berners Lee, <https://www.larecherche.fr/informatique-technologie/tim-berners-lee-«-le-web-change-de-dimension-»-0>, Publié le : Mardi 12 Mars 2019, Consulté le : 25 Mai 2019.
- [6] Fabien Gandon, "Ontologies informatiques", <https://interstices.info/ontologies-informatiques>, Publié le : 22 Mai 2006, Consulté le : 12 Juin 2019.
- [7] Thomas R. Gruber, "Toward Principles for the Design of Ontologies Used for Knowledge Sharing", Stanford Knowledge Systems Laboratory, *International Journal Human-Computer Studies* 43, p.907-928, Révisé le : 23 Août 1993.

- 
- [8] Pascal Hitzler, Markus Krötzsch, Bijan Parsia, Peter F. Patel-Schneider, Sebastian Rudolph, "OWL 2 Web Ontology Language", Primer (Second Edition) , [https://www.w3.org/2007/OWL/wiki/Primer#Classes.2C\\_Properties.2C\\_and\\_Individuals\\_.E2.80.93\\_And\\_Basic\\_Modeling\\_With\\_Them](https://www.w3.org/2007/OWL/wiki/Primer#Classes.2C_Properties.2C_and_Individuals_.E2.80.93_And_Basic_Modeling_With_Them), Modifiée le : 16 Octobre 2012, Consulté le : 22 Juin 2019.
- [9] Thomas R. Gruber, "Ontology", <https://tomgruber.org/writing/ontology-definition-2007.htm>, Consulté le : 12 Juillet 2019.
- [10] Xavier Aimé. "Gradients de prototypicalité, mesures de similarité et de proximité sémantique : une contribution à l'Ingénierie des Ontologies", Artificial Intelligence. Université de Nantes, Soumis le : 18 Janvier 2012.
- [11] <https://www.w3.org/2001/sw/wiki/Protege>, Modifié le : 18 Décembre 2009, Consulté le : 04 Juin 2019.
- [12] Hiba Melhem, Thèse pour obtenir le grade de "Docteur de la communauté université Grenoble Alpes", Spécialité : "Sciences de l'Information et de la Communication, Titré par : "Usages et applications du web sémantique en bibliothèques numériques", Url : "<https://tel.archives-ouvertes.fr/tel-01742957v2/document>", Soutenu le : 5 juin 2018.
- [13] T. Berners-Lee, <http://www.faqs.org/rfcs/rfc2396.html>, août 1998, Consulté le : 13 Août 2019.
- [14] <https://www.ifla.org/node/9337>, Modifiée le : 24 Octobre 2005, Consulté le : 15 Juin 2019.
- [15] Tim Berners-Lee, <https://www.w3.org/DesignIssues/Metadata.html>, Modifiée le : 27 Août 2009, Consulté le : 06 Juillet 2019.
- [16] <https://tcuvelier.developpez.com/tutoriels/web-semantique/rdfa/introduction>, Publié le : 11 Septembre 2012, Consulté le : 30 Juin 2019.
- [17] <https://www.w3.org/XML>, Modifiée le : 7 Février 2013, Consulté le : 05 Septembre 2019.

- 
- [18] <https://www.w3.org/TR/rdf-primer>, Modifiée le : 25 Février 2014, Consulté le : 10 Août 2019.
- [19] <https://www.w3.org/TR/rdf-schema>, Consulté le : 14 Août 2019.
- [20] <https://www.w3.org/TR/2002/WD-rdf-schema-20020430>, Consulté le : 14 Août 2019.
- [21] [https://www.w3.org/2007/OWL/wiki/Primer#Classes.2C\\_Properties.2C\\_and\\_Individuals\\_.E2.80.93\\_And\\_Basic\\_Modeling\\_With\\_Them](https://www.w3.org/2007/OWL/wiki/Primer#Classes.2C_Properties.2C_and_Individuals_.E2.80.93_And_Basic_Modeling_With_Them), Consulté le : 11 Août 2019.
- [22] <https://www.w3.org/TR/owl-ref>, Consulté le : 16 Août 2019.
- [23] <https://www.w3.org/TR/rif-primer>, Consulté le : 13 Août 2019.
- [24] <https://www.w3.org/standards/semanticweb/inference>, Consulté le : 13 Août 2019.
- [25] <https://www.w3.org/TR/sparql11-overview>, Consulté le : 25 Août 2019.
- [26] [https://www.w3.org/2001/sw/wiki/Apache\\_Jena](https://www.w3.org/2001/sw/wiki/Apache_Jena), Consulté le : 05 Septembre 2019.
- [27] Fabien Gandon, "Inria célèbre les 25 ans du Web avec Tim Berners-Lee", <https://www.inria.fr/actualite/actualites-inria/inria-celebre-les-25-ans-du-web>, Publié le : 5 Juin 2014, Consulté le : 23 Juin 2019.
- [28] Jérôme Euzenat, Petko Valtchev, *Similarity-based ontology alignment in OWL-Lite*, Proc, 16 th european conference on artificial intelligence (ECAI), Aug 2004, Valencia, Spain, pp.333-337, fhal-00918127.
- [29] Warith-Eddine DJEDDI, « Alignement sémantique des ontologies de grande Taille », Thèse Doctorat en Informatique Université Badji Mokhtar –Annaba, 2013.
- [30] Mina Ziani, Danielle Boulanger, Guilaine Talens, "Système d'aide à l'alignement d'ontologies métier Application au domaine géotechnique", Université Jean Moulin.

- [31] Youssef Bououlid Idrissi, *"Une approche multi-stratégique et adaptative pour un alignement sémantique intégrant le contexte des données à apparier"*, Thèse présentée en vue de l'obtention du grade de Philosophie Doctor (Ph.D.) en informatique, Université de Montréal, Août 2008.
- [32] Mohamed TOUZANI, *"Alignement des ontologies OWL-Lite"*, Mémoire présenté en vue de l'obtention du grade de Maîtrise ès sciences (M.Sc.) en informatique, Département informatique et de recherche opérationnelle Faculté des arts et des sciences, Université de Montréal, Avril 2005.
- [33] <https://www.w3.org/TR/wordnet-rdf>, Consulté le : 25 Août 2019.
- [34] <https://jena.apache.org/documentation/inference>, Consulté le : 11 Août 2019.

---

## *Résumé*

*L'ontologie est utilisée dans la représentation des connaissances et du web sémantique, ce dernier offre des perspectives d'avenir très prometteuses car il nous permet d'exploiter toutes les données accessibles sur Internet. À cet égard, nous avons présenté dans ce travail une idée de codage des données qui facilite et donne une interopérabilité plus souple entre les machines.*

**Mots clés :** *Web sémantique, données liées, ontologies, alignement, interférence ...*

## *Abstract*

*Ontology is used in the representation of knowledge and of the semantic web, this latter offers very promising future prospects because it allows us to exploit all the data accessible on the Internet. In this regard, we have presented in this work an idea of data encoding that facilitates and gives more flexible interoperability between machines.*

**Key words :** *Semantic web, linked data, ontologies, matching, interference...*