

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université de Djilali BOUNAAMA Khemis Miliana



**Faculté des Sciences et de la Technologie**  
**Département de Mathématiques et d'Informatique**

Mémoire Présenté

Pour l'obtention de diplôme de

**Master** en Informatique

**Option** : ontologies et web sémantique

Titre :

**Alignement d'ontologie**

**Réalisé par :**

- Mansouri Kelthom

- Laoufi Noura

**Encadré par :**

Boudali Fatiha

Soutenu le : 24/06/2018

**Devant le jury composé de :**

D.bahloule

P.haniche

H.hachichi

Président

examineur 1

examineur 2

## *Remerciement*

*Nous tenons dans un premier temps à remercier DIEU tout puissant de nous avoir donné la chance et le privilège d'étudier et de nous avoir permis d'en arriver là.*

*Nous adressons nos remerciements aux personnes qui nous ont aidé dans la réalisation de ce mémoire.*

*En premier lieu, nous remercions notre enseignante BOUDALI qui nous a guidé dans notre travail et nous a aidé à trouver des solutions pour avancer.*

*Enfin, nous remercions toutes les personnes ayant contribué de près ou de loin à l'élaboration de ce mémoire*

## *Dédicace*

*Tous d'abord je remercie le bon dieu qui m'a donné le courage pour arriver à ce stade de fin d'étude.*

*Je dédie ce modeste travail à :*

*A mes parents .Aucun hommage ne pourrait être à la hauteur de l'amour Dont ils ne cessent de me combler. Que dieu leur procure bonne santé et longue vie, sans oublié ma grand-mère.*

*A celui que j'aime beaucoup et qui m'a soutenue tout au long de ce projet : mon fiancé Sahraoui Larbi, et bien sur A mes frères Ahmed et Abd el latif, Mohamed tayeb, et ma sœur meriem.*

*A toute ma famille, et la famille BOUNEDJAR,*

*A mon binôme Noura*

*Et à tous ceux qui ont contribué de près ou de loin pour que ce projet soit possible, je vous dis merci.*

*KELTHOUM*

## *Dédicace*

*J'ai l'immense plaisir de dédier ce modeste travail à ceux qui sont très chers pour moi, ceux qui m'ont encouragé durant ma vie. Mes chers parents*

*que dieu les protège*

*A mon cher fiancé nacer*

*A mon petit frère rafik*

*Mes chères sœurs : Amel, Siham et son époux Lotfi,*

*Mon grand cher frère Oussama*

*A ma chère binôme KeltHOME*

*A ma promotrice Mme. Boudali et tout les personne qui m'ont aidé à réaliser ce travail*

*A Mes amies : elham ; soumia ; imene ; hayet ; zakya ; et à tous les personnes qui m'ont aidé dans ma vie.*

*Laoufi noura*

## LISTE DES ABREVIATIONS

---

### Liste des abréviations

**Rdf** : Ressource Description Framework.

**Rdfs**: Ressource Description Framework Schéma (schéma léger).

**Spyder** : Scientific Python Développment Environment.

**Onto-Edit** : l'éditeur d'ontologie.

**IA** : Intelligence Artificielle.

**OWL**: Ontology Web Language.

**RACER**: Renamed Abox and Concept Expression Reasoner.

**XML**: eXtensible Markup Language.

**O-Graphe** : Ontology Graphe .

**DS** : Distance.

**E-commerce** : Electronic commerce.

**SMA** : système multi-agents.

**SUMO**: Software Update Monitor.

**FMA**: Future Medea Architects.

**QOM** : Quick Ontology Matching.

**NOM** : Naive Ontology Matching.

**SODA** : Structural Ontology OWL -DL Alignement.

## LISTE DES FIGURES ET DES TABLEAUX

### Liste des figures

Figure	Désignation	Numéro de page
Figure I.1	Cycle de vie d'une ontologie	09
Figure I.2	Processus de construction d'ontologie	12
Figure II.1	Les trois dimensions de l'hétérogénéité au niveau conceptuel	22
Figure II.2	Communication entre deux agents ayant deux ontologies distinctes	25
Figure II.3	Composition d'un service web	27
Figure II.4	Les trois dimensions de l'alignement	29
Figure II.5	Exemples de configurations de multiplicité entre les entités de deux ontologies	30
Figure II.6	Le graphe RDF et le graphe RDF biparti d'une ontologie $O_A$	42
Figure III.1	architecture généraliste de fonctionnement l'application.	47
Figure III.2	Représentation O-Grappe d'ontologie « humain »	48
Figure III.3	Représentation O-Grappe d'ontologie « rat »	48
Figure IV.1	Environnement de développement (Anaconda)	58
Figure IV.2	L'interface d'accueil de l'application "Alignement Ontologie"	61
Figure IV.3	La fenêtre de l'explorateur pour choisir le fichier d'ontologie	62

### Liste des tableaux

Figure	Désignation	Numéro de page
Tableau II.1	Comparaison des méthodes par rapport à leurs dimensions externes	43
Tableau II.2	Comparaison des méthodes par rapport à leurs dimensions internes	45
Tableau IV.1	Résultats des correspondances obtenus par l'algorithme proposé	62
Tableau IV.2	Résultats des correspondances inférieures à seuil proposé	63

# Résumé

---

## ***Résumé:***

Le web sémantique a un but de rendre le contenu des documents web accessible afin d'assurer un échange d'information et une interopérabilité entre les différents systèmes ; Les ontologies sont proposées comme élément central du web sémantique.

L'hétérogénéité entre les connaissances décrites dans les ontologies dans le web ont fait surgir plusieurs questions de recherche intéressantes telles que l'intégration, l'évolution et l'alignement des ontologies. Donc notre projet était de développer un outil informatique qui permet d'aligner deux ontologies.

Pour réaliser l'application; nous avons exploité la base de données lexical « Wordnet » pour découvrir si le concept concerné possède au moins un synonyme dans cette dernière, après nous avons utilisé la méthode qui est appelé distance qui calcule la distance entre les entités à aligner, est basée sur les chaînes de caractères qui est une implémentation de la nouvelle version de la méthode « Levenshtein Distance ». Puis trouver les valeurs des similarités. A partir de ces valeurs de similarités les correspondances entre les concepts de deux ontologies sont identifiées donc l'alignement est identifié.

On a testé notre système avec deux ontologies de domaine biomédicale réaliste qui existent dans le web ; l'une sur l'anatomie de l'être humaine et l'autre sur l'anatomie des souris ; et on a obtenu des meilleurs résultats expérimentaux de mesure des similarités.

## ***Summary***

The semantic web has a purpose to make the content of web documents accessible to ensure an exchange of information and interoperability between different systems; Ontologies are proposed as central element of the semantic web.

The heterogeneity between the knowledge described in ontologies in the web has led to several interesting research questions such as the integration, evolution and alignment of ontologies. So our project was to develop a computer tool that aligns two ontologies.

To realize the application; we have exploited the lexical database "Wordnet" to discover if the concept concerned has at least one synonym in the latter, after we used the method which is called distance which computes the distance between the entities to

## Résumé

---

align, is based on the strings that is an implementation of the new version of the method "Levenshtein Distance". Then find the values of the similarities. From these similarity values the correspondences between the concepts of two ontologies are identified so the alignment is identified.

We tested our system with two realistic biomedical domain ontologies that exist in the web; one on the anatomy of the human being and the other on the anatomy of the mice; and better experimental results of similarity measurement were obtained.



# SOMMAIRE

---

Remerciement et dédicaces

Liste des figures

Liste des tableaux

Résumé

Sommaire

Introduction générale .....01/02

## Chapitre I

## Généralité sur les ontologies

I.1. Introduction .....	03
I.2. historique.....	03
I.3. notion de terme d'ontologie .....	03
I.4. rôle des ontologies .....	05
I.5. composants d'ontologie.....	06
I.5.1. Concepts.....	06
I.5.2. Relations.....	06
I.5.3. Fonctions.....	06
I.5.4. Axiomes .....	06
I.5.5. Instances.....	07
I.6. Classifications d'ontologies .....	07
I.6.1. Classification selon l'objet de conceptualisation.....	07
I.6.2. Classification selon le niveau de complétude.....	08
I.6.3. Classification selon le niveau de détail .....	08
I.6.4. Classification selon le formalisme utilisé .....	08
I.7. Le cycle de vie des ontologies .....	09
I.7.1. Besoins et évaluation .....	10
I.7.2. Conception et évolution .....	10
I.7.3. Diffusion .....	10
I.7.4. Utilisation.....	11
I.7.5. Gestion .....	11
I.8. Processus de construction d'ontologie .....	11

## SOMMAIRE

---

I.8.1. Conceptualisation .....	11
I.8.2. Ontologisation.....	11
I.8.3. Opérationnalisation.....	11
I.9. Les méthodologies et les outils de construction d'ontologie .....	12
I.9.1. Les méthodologies de construction d'ontologie.....	12
I.9.1.1. ENTREPRISE.....	12
I.9.1.2. TOVE .....	13
I.9.1.3. METHONTOLOGY .....	13
I.9.2. Outils de construction d'ontologie.....	14
I.9.2.1. Terminae.....	14
I.9.2.2. OntoEdit .....	14
I.9.2.3. L'Editeur d'Ontologies Protégé.....	14
I.9.2.4. Les systèmes de raisonnement.....	15
I.10. Les langages de construction d'ontologie.....	16
I.10.1. Le langage OWL.....	16
I.11. Conclusion.....	17

## Chapitre II

## Alignement D'ontologie

II.1. Introduction.....	18
II.2. Comment est né le besoin à l'alignement d'ontologie .....	19
II.2.1. Problème issu de l'hétérogénéité .....	20
II.3. Définition de l'alignement d'ontologies .....	23
II.4. Quelques domaines d'application de l'alignement d'ontologies.....	25
II.5. Le processus d'alignement .....	28
II.6. Les techniques d'alignement.....	30
II.6.1. Les méthodes de base pour mesurer la similarité.....	30
II.6.1.1. La similarité.....	30
II.6.2. Les méthodes terminologiques .....	32
II.6.2.1. Méthodes syntaxiques .....	32
II.6.2.2. Méthodes linguistiques.....	32
II.6.2.2.1. Méthodes intrinsèques.....	32

## SOMMAIRE

II.6.2.2.2. Méthodes extrinsèques.....	33
II.6.3. Méthodes structurelles.....	34
II.6.3.1. Méthodes structurelles internes.....	34
II.6.3.2. Méthodes structurelles externes.....	35
II.6.4. Les méthodes extensionnelles.....	35
II.6.5. Le méthodes sémantiques .....	36
II.6.5.1. Les techniques basées sur les ontologies externes.....	36
II.6.5.2. Les techniques déductives.....	37
II.6.6. Les méthodes d'alignement combinées.....	37
II.7. Quelques approches d'alignement d'ontologies.....	39
II.8. Comparaison des méthodes d'alignement.....	42
II.8.1. Comparaison selon les dimensions externes d'une méthode d'alignement .....	42
II.8.2. Comparaison selon les dimensions internes d'une méthode d'alignement.....	43
II.9. Conclusion.....	45

### Chapitre III

### conception

III.1. Introduction.....	46
III.2. Approche suivie .....	46
III.2.1.modèle de graphe « O-graphe » .....	47
III.2.2. le problème de calcul des similarités.....	48
III.2.2.1. la méthode utilisé pour le calcul de la distance .....	49
III.3. Algorithme d'alignement proposé.....	50
III.3.1 Les grandes lignes du déroulement de l'algorithme .....	50
III.3.2. Calculer les distances entre les classes et les sous-classes .....	51
III.3.3. Calculer les distances entre les propriétés .....	53
III.3.4. Création des matrices .....	54
III.3.5. La matrice des classes « Class_matrix ».....	54
III.3.6. La matrice des propriétés « Prop_matrix ».....	55
III.4. Conclusion.....	56

### Chapitre IV

### implémentation

## SOMMAIRE

---

IV.1 Introduction.....	57
IV.2. Outils de développement .....	57
IV.3. Les ontologies de tests .....	59
IV.3.1.L'ontologie « humain » et L'ontologie « rat » .....	59
IV.4. Implémentation de notre système .....	60
IV.4.1 Description des classes de tâche du système «Objet Alignement ».....	60
IV.4.2. Présentation de notre système.....	61
IV.4.2.1. Page d'accueil .....	61
IV.5. Résultats expérimentaux .....	62
IV.5.1. Les Concepts ayant une valeur de similarité inférieure au seuil proposé (0.4).....	63
IV.6 Conclusion.....	63
Conclusion Général.....	64
Bibliographié .....	65

## INTRODUCTION GENERALE

Le but principal du web sémantique est de rendre le contenu des documents web accessible par les applications afin d'assurer un échange d'information et une interopérabilité entre les différents systèmes. Les ontologies sont proposées comme élément central du web sémantique. Ces dernières permettent de décrire la structure et la sémantique de données contenues dans un document web. Elles assurent aussi l'organisation des informations sous forme d'une taxonomie de concepts et de relations entre ces derniers, d'autant plus que ça, les machines et les agents logiciels peuvent interpréter leurs sémantiques. Cependant, il n'existe pas d'ontologie universelle partagée, adoptée par tous les utilisateurs d'un domaine donné. Les problématiques et les tentatives d'amélioration de l'interopérabilité du système comptent donc sur la réconciliation des différentes ontologies utilisées dans un domaine par les différents systèmes. Cette réconciliation consiste à trouver les liens de correspondances entre ces ontologies, on parle donc de l'**alignement d'ontologies**.

L'alignement d'ontologies cherche des correspondances sémantiques entre les différents systèmes d'information et joue un rôle important pour permettre l'interopérabilité sémantique. Cet ensemble de correspondances ou encore alignement peut être par la suite utilisé pour fusionner les ontologies, créer une troisième ontologie à partir des entités des deux ontologies en entrée....

Les techniques d'alignement jouent un rôle crucial dans la construction d'un lien sémantique entre les ontologies d'un même domaine. Quelques approches d'alignement considèrent que l'utilisation d'une connaissance sur le domaine est une manière assurant la correspondance sémantique entre la dissimilarité syntaxique des ontologies. L'obtention de la bonne connaissance sur le domaine est primordiale. D'autres approches n'exploitent pas une connaissance sur le domaine et ne réalisent pas un modèle sémantique formel pour l'alignement des structures produites. Dans ce cas, la structure obtenue est difficile à exploiter, pour répondre aux requêtes interrogeant les ontologies. En outre, les approches courantes d'alignement d'ontologies sont basées sur les mesures de similarité entre chaînes de caractères et des structures composites. Les ontologies à aligner peuvent être représentées avec différents langage.

La découverte des alignements d'une manière manuelle est une tâche coûteuse en temps, inefficace et peut produire des erreurs. C'est pour cette raison que plusieurs méthodes d'appariement ont été proposées.

## Problématique :

L'hétérogénéité des ontologies rend la problématique de l'alignement des ontologies particulièrement importante pour favoriser l'interopérabilité sémantique. Pour cela l'objectif de notre projet porte sur l'alignement des ontologies et consiste en le développement d'un système qui fait l'alignement d'ontologies structurés et décrites avec OWL, nous avons effectué une proposition réaliste applicable au domaine médicale « c'est l'anatomie ». Notre proposition consiste sur l'utilisation d'une ressource externe qui est Wordnet (une base de données lexicale) pour identifier les correspondances sémantiques, ainsi par l'utilisation de la synonymie on peut déduire des équivalences entre concepts dont les labels ne sont pas syntaxiquement similaires. La deuxième phase est basée sur les chaînes de caractères en particulier la mesure de Levenstein pour calculer la distance entre les entités des deux ontologies et déterminer les valeurs de la similarité ; on a utilisé la nouvelle version de la méthode Levenshtein- distance c'est la méthode "distance".

## Plan général du mémoire :

Ce mémoire est composé de quatre chapitres:

- **Chapitre 1 :** ce chapitre présente une introduction sur les ontologies, en donnant les différentes définitions, les types, cycle de vie, le rôle des ontologies ; ainsi que ses différents composants, les langages de représentations et aussi les outils et Les méthodologies utilisés pour leur développement.
- **Chapitre 2 :** ce chapitre décrit l'alignement des ontologies. nous allons présenter comment est né le besoin à l'alignement des ontologies, Nous présentons le concept d'alignement, quelques domaines d'application de l'alignement des ontologies, les différentes techniques de calcul des similarités, ainsi que les méthodes d'alignement utilisées.
- **Chapitre 3 :** ce chapitre est dédié à la conception de notre system, nous allons présenter une architecture générale pour le fonctionnement de notre application ; puis on a présenté les deux graphe ontologique de nos ontologies choisis ; ainsi que le problème de calcule des similarités et après la méthode utilisée pour calculer la distance entre les entités d'ontologies avec quelques exemples de calcule des valeurs de distance et en fin nous avons citer quelques étapes principale de notre algorithme d'application et son déroulement pour mesurer les valeurs des similarités et afficher la résultat finale obtenu .
- **Chapitre 4 :** ce chapitre est consacré à l'implémentation de notre système .il contient la représentation des outils utilisés pour la réalisation de notre application, Ainsi que les ontologies de test dans notre application et les résultats expérimentaux obtenus. Il contient aussi une illustration sur notre système à l'aide de quelques copies d'écran.

## I.1. Introduction

L'explosion du nombre de sources d'informations accessibles via le Web multiplie les besoins de techniques permettant l'intégration de ces sources. En définissant les concepts associés à des domaines particuliers, les ontologies sont un élément essentiel des systèmes d'intégration, car elles permettent à la fois de décrire le contenu des sources à intégrer et d'explicitier le vocabulaire utilisable dans les requêtes des utilisateurs [1].

Afin qu'une communication efficace soit possible entre le système et les utilisateurs ; Le système doit alors, avoir accès non seulement aux termes utilisés par l'être humain mais aussi à la sémantique qui leurs est associée [2].

Dans ce cadre, les ontologies vont servir à l'interprétation de ces connaissances en spécifiant la sémantique de la représentation utilisée, afin de pouvoir les exploiter par des machines [3].

Dans ce chapitre, nous allons présenter la définition de la notion d'ontologie et le rôle des ontologies. Nous présentons aussi les différents composants d'ontologie et Nous décrivons les Différents types d'ontologies ainsi que leurs Cycle de vie, Les méthodologies et les outils de construction d'ontologies et les langages de construction des ontologies

## I.2. Historique

À l'origine, l'ontologie est une notion philosophique, et on considérait que l'étude de l'ontologie était une partie de la métaphysique, qui s'intéresse à l'étude des propriétés de l'être, et par extension de l'existence.

John McCarthy a introduit l'ontologie en intelligence artificielle en 1980, par le principe que les concepteurs de systèmes intelligents fondés sur la logique devraient d'abord établir une base de données complète de ce qui existe, et ensuite utiliser ces données. Les ontologies sont apparues en intelligence artificielle, comme réponses aux problématiques de représentation et de manipulation des connaissances au sein des systèmes informatiques [4].

## I.3. notion de terme « ontologie »

**I.3.1 Définition 1:** ce terme est construit à partir des racines grecques « onto » qui veut dire ce qui existe, l'être, l'existant, et « logos » qui veut dire l'étude, le discours, d'où sa traduction par « l'étude de l'être » et par extension de l'existence [5]. ; nous pouvons dire que le mot « ontologie » provient du domaine de la philosophie, qui sert à désigner une théorie basée sur l'étude de

l'être, d'une coté, et il peut aussi être interprété comme l'ensemble de ce qui existe avec ses relations, restrictions, axiomes et vocabulaires dans le domaine de l'Intelligence Artificiel [6].

Dans le cadre de l'intelligence artificielle ; « une ontologie définit les termes et les relations de base de vocabulaire d'un domaine, ainsi que les règles qui indiquent comment combiner les termes et les relations de façon à pouvoir étendre le vocabulaire» [4].

**I.3.2 Définition 2:** cette notion est définie comme suit«Une ontologie est une spécification formelle et explicite d'une conceptualisation partagée» avec la signification des termes suivants:

- **Formelle** : réfère au fait qu'une ontologie doit être compréhensible par la machine, c'est-à-dire que cette dernière doit être capable d'interpréter la sémantique de l'information fournie ;
- **Explicite** : signifie que le type de concepts utilisés et les contraintes sur leur utilisation doivent être explicitement définis ;
- **Conceptualisation** : se réfère à un modèle abstrait de certains phénomènes dans le monde qui identifie les concepts appropriés de ce phénomène ;
- **Partagée** : indique que l'ontologie supporte la connaissance consensuelle, et elle n'est pas restreinte à certains individus mais est acceptée par un groupe [7].

Ainsi une ontologie représente un modèle conceptuel sur lequel il est possible de développer des systèmes de connaissances qui soient partageable et réutilisables permettant ainsi, l'interopérabilité entre plusieurs sources d'information et de connaissances [5].

**I.3.3 Définition 3:** Historiquement, le terme ontologie a tout d'abord été défini en philosophie comme une branche de la métaphysique qui s'intéresse à l'existence, à l'être en tant qu'être et aux catégories fondamentales de l'existant. Plus tard, le concept d'ontologie est apparu en pleine lumière dans le domaine de l'intelligence artificielle, afin de résoudre les problèmes de modélisation des connaissances et, plus précisément, en ingénierie des connaissances [3].

Techniquement parlant, une ontologie, tel qu'elle est utilisée en informatique, peut être minimalement définie en tant qu'artéfact conceptuel qui rassemble un vocabulaire de termes, des spécifications quant à la signification de ces termes et une description de la manière dont ils sont interreliés les uns aux autres. Les termes en question correspondent normalement à ce que l'on appelle « classes », « concepts » ou « choses ». Les spécifications correspondent pour leur part aux « propriétés » ou « attributs » que devrait avoir un objet tombant sous un certain concept. Et les descriptions d'interrelation représentent quant à elles les « relations » que certains concepts entretiennent conceptuellement avec d'autres concepts. Les ontologies peuvent être utilisées pour décrire différents types de modèles. Elles peuvent par exemple décrire des taxonomies



élémentaires comme elles peuvent décrire des modèles plus élaborés écrits dans une variante de la logique classique du premier ordre, comme la logique descriptive.

Au-delà de son simple rôle de modèle standardisant des éléments de données, les ontologies visent à expliciter la sémantique (la signification) des ressources informationnelles décrites par les données RDF. En ce sens, on peut dire des ensembles de données RDF qui réfèrent à des ontologies comme modèles qu'ils sont auto-descriptifs : la manière d'interpréter les données fait partie intégrante des données elles-mêmes et ne dépend d'aucun système particulier pour ce faire. C'est pourquoi l'usage de telles données sémantiques constitue une avancée notable en matière d'interopérabilité des systèmes [8].

#### I.4. Rôle des ontologies

Nées des besoins de représentation des connaissances, les ontologies sont à l'heure actuelle au cœur des travaux menés dans le Web sémantique. Visant à établir des représentations à travers lesquelles les machines peuvent manipuler la sémantique des informations, la construction des ontologies demande à la fois une étude des connaissances humaines et la définition de langages de représentation, ainsi que la réalisation de systèmes pour les manipuler. Les ontologies participent donc pleinement aux dimensions scientifiques et techniques de l'Intelligence Artificielle (IA) : scientifiques comme étude des connaissances humaines et plus largement de l'esprit humain, ce qui rattache l'IA aux sciences humaines, et techniques comme création d'artefacts possédant certaines propriétés et capacités en vue d'un certain usage.

Au fur et à mesure des expérimentations, des méthodologies de construction d'ontologies et des outils de développement adéquats sont apparues. Émergeant des pratiques artisanales initiales, une véritable ingénierie se constitue autour des ontologies, ayant pour but leur construction mais plus largement leur gestion tout au long d'un cycle de vie. Les ontologies apparaissent ainsi comme des composants logiciels s'insérant dans les systèmes d'information et leur apportant une dimension sémantique qui leur faisait défaut jusqu'ici.

Le champ d'application des ontologies ne cesse de s'élargir et couvre les systèmes conseillers (systèmes d'aide à la décision, systèmes d'enseignement assisté par ordinateur -*e-Learning*, etc.), les systèmes de résolution de problèmes et les systèmes de gestion de connaissances (par exemple dans le domaine du biomédical). Un des plus grands projets basé sur l'utilisation des ontologies consiste à ajouter au Web une véritable couche de connaissances permettant, dans un premier temps, la recherche d'information aussi bien au niveau syntaxique qu'au niveau sémantique. L'enjeu de l'effort engagé est de rendre les machines suffisamment sophistiquées pour qu'elles puissent intégrer le sens des informations, qu'à l'heure actuelle, elles ne font que manipuler formellement. Mais en attendant que des ordinateurs (charges)

d'ontologies et de connaissances nous soulagent en partie du travail de plus en plus lourd de gestion des informations dont le flot a tendance à nous submerger, de nombreux problèmes théoriques et pratiques restent à résoudre [7].

## I.5. Composants d'ontologie

Les ontologies rassemblent les connaissances propres à un domaine donné. En représentation des connaissances, ces ontologies existent sous la forme de concepts et de relations, et permettent d'en fixer la sémantique selon un degré de formalisme variable. Nous détaillons ci-dessous les différents composants de l'ontologie : [3]

### I.5.1 Concepts

Un concept peut représenter un objet, une notion, une idée. Un concept peut être divisé en trois parties : un terme (ou plusieurs), une notion et un ensemble d'objets.

- Le terme est un élément lexical qui permet d'exprimer le concept en langue naturelle, il peut admettre des synonymes.

-La notion également appelée *intension* du concept, contient la sémantique du concept, exprimée en termes de propriétés et attributs, et de contraintes.

-L'ensemble d'objets appelé *extension* du concept, regroupe les objets manipulés à travers le concept ; ces objets sont appelés instances du concept. Par exemple, le terme « table » renvoie à la fois à la notion de table comme objet de type « meuble » possédant un « plateau » et des « pieds », et à l'ensemble des objets de ce type.

### I.5.2 Relations

Si certains liens conceptuels existant entre les concepts peuvent s'exprimer à l'aide de propriétés portées par les concepts, d'autres doivent être représentés à l'aide de relations autonomes. Une relation permet de lier des instances de concepts, ou des concepts génériques. Elles sont caractérisées par un terme (voire plusieurs) et une signature qui précise le nombre d'instances de concepts que la relation lie, leurs types et l'ordre des concepts, C'est-à-dire la façon dont la relation doit être lue. Par exemple, la relation « écrit » lie une instance du concept «personne» et une instance du concept «texte», dans cet ordre.

### I.5.3 Fonctions

Les fonctions constituent des cas particuliers de relations, dans laquelle un élément de la relation, le nième (extrant) est défini en fonction des n-1 éléments précédents (intrants).

### I.5.4 Axiomes

Les axiomes constituent des assertions acceptées comme vraies à propos des abstractions du domaine traduit par l'ontologie. Interviennent dans la définition des concepts ou des relations, dans l'inférence de nouvelles informations, ....

### I.5.5 Instances

Les instances constituent la définition extensionnelle de l'ontologie ; ces objets véhiculent les connaissances (statiques, factuelles) à propos du domaine du problème.

## I.6. Classifications d'ontologies

On peut distinguer quatre typologies selon plusieurs critères: [5]

- ✓ Classification selon le niveau de complétude.
- ✓ Classification selon l'objet de conceptualisation.
- ✓ Classification selon le niveau du formalisme.
- ✓ Classification selon le niveau de détail.

### I.6.1 Classification selon l'objet de conceptualisation

Les ontologies peuvent être subdivisées en plusieurs niveaux qui sont, entre autres :

**-Les ontologies de type thesaurus** : Sont aussi appelées taxonomie, elles servent à définir un vocabulaire de référence.

**-Les ontologies de domaine** : Ces ontologies expriment des conceptualisations spécifiques à un domaine. Elles sont réutilisables pour plusieurs applications de ce domaine. L'ontologie du domaine caractérise la connaissance du domaine où la tâche est réalisée.

**-Les ontologies applicatives** : Ces ontologies contiennent des connaissances du domaine nécessaire à une application donnée, elles sont spécifiques et non réutilisable.

**-Les ontologies génériques ou ontologies de haut niveau (top-ontologies)** : Ces ontologies expriment des conceptualisations valables dans différents domaines. Son sujet est l'étude des catégories des choses qui existent dans le monde. Comme les concepts de haute abstraction tels que les entités, les événements, les états, les actions, le temps, l'espace, les relations, etc.

**-Les ontologies de représentation ou méta-ontologies**: Ces ontologies conceptualisent les primitives des langages de représentation des connaissances.

Les ontologies géographiques : les ontologies de l'espace plus spécifiquement dédiées à la description des concepts qui caractérisent l'espace comme le point, la ligne, etc... Ces ontologies sont typiquement élaborées par de grands organismes de normalisation. Les ontologies spatialisées (ou spatio-temporelles), sont des ontologies dont les concepts sont localisés dans l'espace. Une composante temporelle est souvent nécessaire en complément pour la modélisation de l'information géographique.

### I.6.2. Classification selon le niveau de complétude

On peut définir trois niveaux de complétude :

**-Niveau sémantique :** Tous les concepts, caractérisés par un terme/libellé, doivent respecter les quatre principes différentiels :

- ✓ Communauté avec l'ancêtre ;
- ✓ Différence, spécification, par rapport à l'ancêtre ;
- ✓ Communauté avec les concepts frères, situés au même niveau ;
- ✓ Différence par rapport aux concepts frères.

Ces principes correspondent à l'engagement sémantique et assurent que chaque concept aura un sens univoque et non contextuel associé. Deux concepts sont identiques si l'interprétation du terme/libellé à travers les quatre principes différentiels aboutit à un sens équivalent.

**-Niveau-Référentiel :** Les concepts référentiels ou formels, se caractérisent par un terme/libellé dont la sémantique est définie par une extension d'objets. L'engagement ontologique spécifie les objets du domaine qui peuvent être associés au concept, conformément à sa signification formelle. Deux concepts formels seront identiques s'ils possèdent la même extension.

**Niveau-Opérationnel :** Les concepts du niveau opérationnel ou computationnel sont caractérisés par les opérations qu'il est possible de leur appliquer pour générer des interfaces ou engagement computationnel.

### I.6.3. Classification selon le niveau de détail

On peut différencier les ontologies selon le niveau de description utilisé :

**-Granularité fine :** Ce niveau correspond à des ontologies très détaillées, possédant ainsi un vocabulaire plus riche capable d'assurer une description détaillée des concepts pertinents d'un domaine ou d'une tâche.

**-Granularité large :** Ce niveau correspond à des vocabulaires moins détaillés. Par exemple les scénarios d'utilisation spécifiques ou les utilisateurs sont déjà préalablement d'accord à propos d'une conceptualisation sous-jacente. Les ontologies de haut niveau possèdent une granularité large, compte tenu que les concepts qu'elles traduisent sont normalement raffinés ultérieurement dans d'autres ontologies de domaine ou d'application.

### I.6.4 Classification selon le formalisme utilisé

On peut distinguer les ontologies selon le formalisme utilisé pour les exprimer.

- **Informelle :** l'ontologie est exprimée en langage naturelle. Cela peut permettre de rendre plus compréhensible l'ontologie pour l'utilisateur, mais cela peut rendre plus difficile la vérification de l'absence de redondances ou de contradiction ;

- **Semi-informelle** : l'ontologie est exprimée dans une forme restreinte et structurée de la langue naturelle ; cela permet d'augmenter la clarté de l'ontologie tout en réduisant l'ambiguïté ;
- **Semi-formelle** : l'ontologie est exprimée dans un langage artificiel défini formellement ;
- **Formelle** : l'ontologie est exprimée dans un langage artificiel disposant d'une sémantique formelle, permettant de prouver des propriétés de cette ontologie. L'intérêt d'une ontologie formelle est la possibilité d'effectuer des vérifications sur l'ontologie : complétude, non redondance, consistance, cohérence, etc.

### I.7. Le cycle de vie des ontologies

Il existe plusieurs types d'activités liées aux ontologies ; on peut citer : [4]

- Les activités de gestion ;
- Les activités de développement (spécification, conceptualisation, formalisation) ;
- Les activités transversales de support.

Un cycle de vie d'une ontologie débute du besoin qui se transforme en idée, la concrétisation de l'idée qui se traduit par la conception qui est diffusée pour son utilisation. Vient ensuite l'étape de l'évaluation qui donne naissance, le plus souvent à une étape d'évolution et de maintenance du modèle. Une réévaluation de l'ontologie et des besoins devra se faire après chaque utilisation significative.

L'ontologie peut être étendue et, si nécessaire, en partie reconstruite. La validation du modèle de connaissances est au centre du processus et se fait de manière itérative.

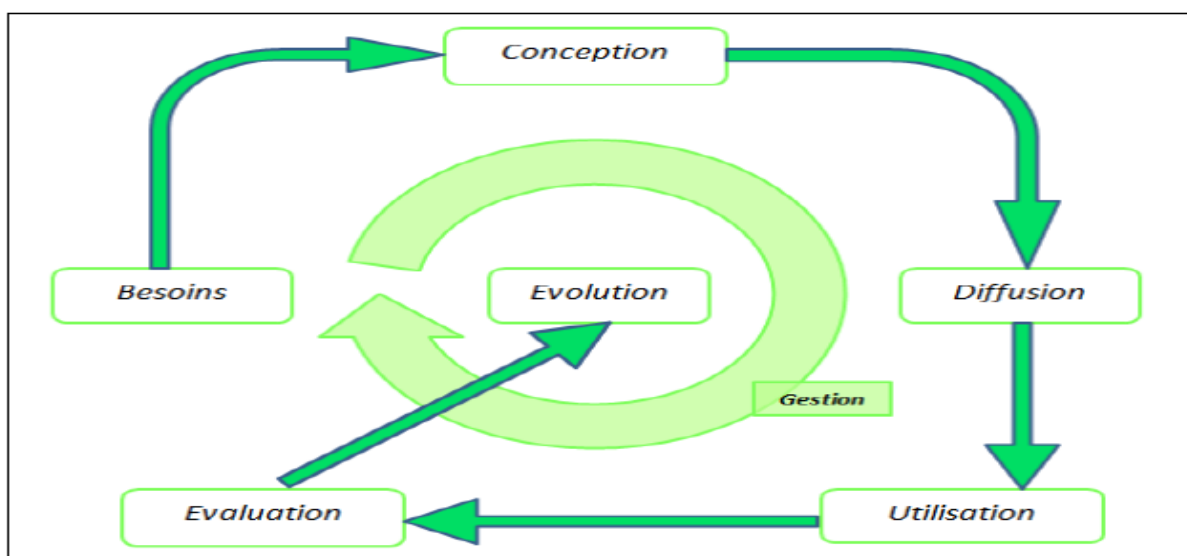


Figure (I.1) : Cycle de vie d'une ontologie. [4]

### **I.7.1. Besoins et évaluation**

Dans l'utilisation d'une ontologie, des problèmes méthodologiques de recueil d'information, d'analyse et d'identification peuvent surgir. Afin d'éviter ces problèmes dans l'utilisation d'une ontologie, un état des lieux initial approfondi doit être élaboré dans l'étape de détection des besoins, car cette étape ne peut reposer sur des études précédentes ou des retours d'utilisation comme dans le cas de l'activité d'évaluation.

### **I.7.2. Conception et évolution**

Les phases de conception initiale et d'évolution ont en commun un certain nombre de points :

- spécification des solutions ;
- conceptualisation et modélisation ;
- formalisation (logiques de description, graphes conceptuels, formalismes du web sémantique RDF, RDF(S) et OWL) ;
- intégration de ressources existantes ;
- implantation (graphes conceptuels, logiques de description).

Le choix de représentation et de conceptualisation faits dans l'ontologie représente un problème de conception et d'évolution. Notons aussi que l'évolution pose le problème de la maintenance de ce qui repose déjà sur l'ontologie. L'ontologie est à la fois un ensemble évolutif et un ensemble de primitives pour décrire des faits et des algorithmes sur ces faits. Ses changements donc, ont un impact direct sur tout ce qui a été construit sur la base de cette ontologie. Le maintien de la cohérence dans une ontologie est un des points clés dans son utilisation. La maintenance de l'ontologie soulève donc des problèmes d'intégration technique et des problèmes d'intégration aux usages. Les domaines de la maintenance comme l'historique et la gestion des versions, la propagation des changements après modification, sont des questions importantes de la recherche actuelle.

### **I.7.3 Diffusion**

Le déploiement et à la mise en place de l'ontologie interviennent dans la phase de diffusion, où les problèmes sont fortement contraints par l'architecture des solutions. Pour l'application Web on utilisera des technologies adéquates et pour le partage de fichiers, nous pouvons utiliser des architectures distribuées ou

l'architectures peer to peer. Dans toutes les architectures dans les serveurs web, services web, peer to peer, etc. la distribution des ressources et leur hétérogénéité du point de vue syntaxiques,

sémantiques, protocolaires, contextuelles, ou autres posent des problèmes de recherche sur l'interopérabilité et le passage à l'échelle (larges bases, optimisation d'inférences, propagation de requêtes, etc.).

#### **I.7.4 Utilisation**

Les activités reposant sur la disponibilité de l'ontologie, comme l'annotation des ressources, la résolution de requête, la déduction de connaissances, etc. constituent la phase d'utilisation.

Les problèmes de la conception des interactions avec l'utilisateur et de leur ergonomie concernant les interfaces dynamiques, les profils et contextes d'utilisation, sont posés dans toutes ces activités. Sur ce point, l'ontologie apporte à la fois de nouvelles solutions et de nouveaux problèmes.

#### **I.7.5 Gestion**

L'existence d'une activité permanente de gestion et planification est importante pour assurer une pérennité dans le travail de suivi et la politique globale pour la détection, la préparation et l'évaluation des itérations d'un cycle et s'assurer que l'ensemble de ces causes améliorent les systèmes d'information.

### **I.8. Processus de construction d'ontologie**

Le processus de construction d'ontologie est une collaboration qui réunit des experts du domaine de connaissance, des ingénieurs de la connaissance, voire les futurs utilisateurs de l'ontologie [9].

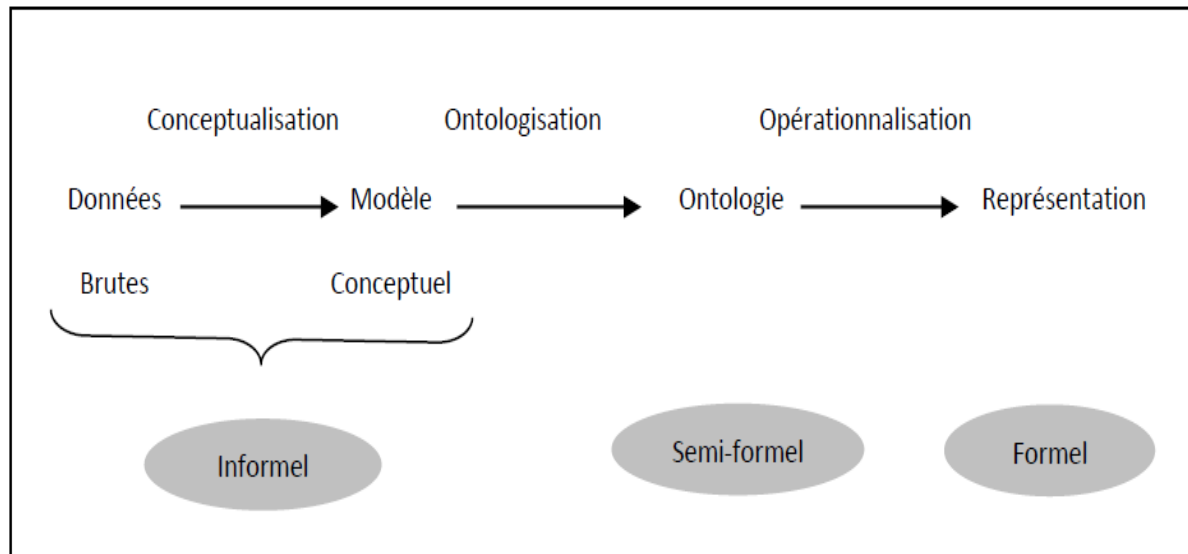
Cette collaboration ne peut être fructueuse que si les objectifs du processus ont été clairement définis, ainsi que les besoins qui en découlent. La construction d'ontologie s'effectue en trois étapes :

**I.8.1. Conceptualisation** : identification des connaissances contenues dans un corpus représentatif du domaine. Ce travail doit être mené par un expert du domaine, assisté par un ingénieur de la connaissance ;

**I.8.2. Ontologisation** : formalisation, autant que possible, du modèle conceptuel

Obtenu à l'étape précédente. Ce travail doit être mené par l'ingénieur de la connaissance, assisté de l'expert du domaine ;

**I.8.3. Opérationnalisation** : transcription de l'ontologie dans un langage formel et opérationnel de représentation de connaissances. Ce travail doit être mené par l'ingénieur de la connaissance ;



**Figure (I.2):** Processus de construction d'ontologie [9].

Il est à noter que ce processus n'est pas linéaire et que de nombreux aller-retour sont à priori nécessaires pour bâtir une ontologie opérationnelle adaptée aux besoins.

## I.9. Les méthodologies et les outils de construction d'ontologie

La construction des ontologies, nécessite le passage d'une étape indispensable, qui est l'analyse des méthodologies et des outils de construction d'ontologies qui existent, dans la perspective de pouvoir identifier une méthodologie et/ou un outil qui permet de construire une ontologie qui répond aux maximum aux besoins des utilisateurs.

### I.9.1 Les méthodologies de construction d'ontologie

L'absence de directives et de méthodes consensuelles entrave, d'une part, le développement d'ontologies communes et acceptées par les équipes et entre elles, et d'autre part, l'extension d'une ontologie donnée à partir d'autres, sa réutilisation dans d'autres ontologies et dans des applications finales.

Bien qu'aucune méthode générale n'ait pour l'instant réussi à s'imposer, de nombreux critères de construction d'ontologies ont été proposés. Ces méthodes peuvent porter sur l'ensemble du processus et guider l'ontologiste dans toutes les étapes de la construction.

ENTERPRISE, TOVE et METHONTOLOGY sont les méthodes les plus représentatives pour construire des ontologies [10].

**I.9.1.1 Entreprise :** la méthode ENTERPRISE repose sur quatre étapes :

- Identifier le rôle et la portée de l'ontologie,



- Dans cette étape, l'ontologie est réellement construite. Les activités Suivantes sont distinguées : identifier les concepts et relations fondamentaux et des définitions provisoires de ces éléments, coder l'ontologie dans un langage adapté, intégrer des ontologies existantes,
- Evaluer l'ontologie,
- Rédiger une documentation et une trace des actions réalisées lors des différentes phases. Les étapes et sous-tâches de la méthode ENTERPRISE, sont décrites de façon abstraite. Les techniques utilisées pour les sous-tâches ne sont pas précisées (par exemple : Comment identifier les concepts fondamentaux ? Quel langage utiliser pour représenter l'ontologie ?).

**I.9.1.2 Tove :** la méthode TOVE repose sur les principales étapes suivantes :

- Capturer des scénarios de motivations : Cette étape consiste à identifier des scénarios qui clarifient le domaine que l'on investit et les différentes applications dans lesquelles l'ontologie sera employée.
- Formulation des questions de compétences informelles : Cette étape consiste à formuler un ensemble de questions (basées sur les scénarios), exprimées en langage naturel, afin de déterminer la portée de l'ontologie.

Ces questions et leurs réponses sont utilisées pour extraire les concepts principaux, leurs propriétés et les relations qui existent entre ces concepts.

- Spécifications de la terminologie de l'ontologie : Cette étape consiste à représenter les termes (concepts, propriétés et relations), identifier dans l'étape précédente, en utilisant le formalisme de la logique du premier ordre. Les concepts seront représentés sous forme de constantes ou bien des variables. Par ailleurs, les propriétés et les relations seront représentées par des prédicats (fonctions).
- Evaluation de la complétude de l'ontologie : La méthode TOVE, reste spécifiée de façon abstraite. Ni les différentes étapes ni les techniques ne sont décrites en détail.

**I.9.1.3 Methontology:** L'approche METHONTOLOGY distingue les étapes suivantes :

- Cadrage** : Cette étape, consiste à cerner l'étendue de l'ontologie et le domaine à prendre en compte.
- Conceptualisation** : Le but de cette étape est d'identifier et de structurer les connaissances du domaine en utilisant un ensemble de représentations intermédiaires semi-formelles (des tables et des graphes), faciles à comprendre par les experts du domaine et qui sont indépendants du formalisme à utiliser pour représenter l'ontologie.

**-Implémentation:** Cette étape consiste à formaliser le modèle conceptuel obtenu dans l'étape précédente par un formalisme de représentation d'ontologie. Puis, coder l'ontologie dans un langage d'ontologie formel.

METHONTOLOGY s'inspire d'une méthode de développement de Système à base de connaissances. Elle spécifie de façon très détaillée l'étape de conceptualisation [9].

### **I.9.2 Outils de construction d'ontologie**

De nombreux outils de construction d'ontologies utilisant des formalismes variés et offrant différentes fonctionnalités ont été développés citant :

#### **I.9.2.1. Terminae :**

Terminae, développé au LIPN de l'Université Paris-Nord, permet, à travers l'outil d'ingénierie linguistique LEXTER, d'extraire d'un corpus textuel les candidats termes d'un domaine. Ces concepts doivent ensuite être triés par un expert et organisés hiérarchiquement, puis la sémantique du domaine est précisée à travers des axiomes. TERMINAE offre ainsi une aide à la conceptualisation.

#### **I.9.2.2. OntoEdit :**

OntoEdit (Ontology Editor) développé par la compagnie Ontoprise, est également un environnement de construction d'ontologies basé sur une méthodologie. Il permet l'édition des hiérarchies de concepts et de relations dans le cadre du paradigme des frames et l'expression d'axiomes algébriques portant sur les relations, et de propriétés telles que la généralité d'un concept. Des outils graphiques dédiés à la visualisation d'ontologies sont inclus dans l'environnement.

OntoEdit intégré, dans sa version commerciale, est un serveur destiné à l'édition d'une ontologie par plusieurs utilisateurs ainsi qu'un plug-in permettant le test de la cohérence d'une ontologie. Enfin, un plugin nommé ONTOKICK offre la possibilité de générer les spécifications de l'ontologie par l'intermédiaire de questions de compétence. OntoEdit gère de nombreux formats de représentation de connaissance dont DAML+OIL, RDFS et FLogic [2].

#### **I.9.2.3. L'Editeur d'Ontologies Protégé:**

**Protégé:** est le plus connu et le plus utilisé des éditeurs d'ontologie. Open-source, développé par l'Université de Stanford, il a évolué depuis ses premières versions (Protégé-2000), c'est un éditeur qui permet de construire une ontologie pour un domaine donné, de définir des formulaires d'entrée de données, et d'acquérir des données à l'aide de ces formulaires sous forme d'instances de cette ontologie. Protégé est également une librairie Java qui peut être étendue pour

créer de véritables applications à bases de connaissances en utilisant un moteur d'inférence pour raisonner et déduire de nouveaux faits par application de règles d'inférence aux instances de l'ontologie et à l'ontologie elle-même (méta-raisonnement) [2].

Protégé possède une interface graphique qui offre un ensemble de fonctionnalités qui permet d'éditer les ontologies. Des extensions permettent de sauvegarder les ontologies créées sous le format OWL-DL, RDF et RDFS [6].

#### I.9.2.4. Les systèmes de raisonnement

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

- **RACER** (Renamed Abox and Concept Expression Reasoner) : est un système de raisonnement des logiques de descriptions. Permet le test de **satisfiabilité** d'un concept (vérifier qu'un concept « C » admet des instances) le test de **subsumption** de concepts (vérifier qu'un concept « C » est subsumé par un concept « D »), et le test d'**instanciation** (vérifier qu'un individu « a » est un instance d'un concept « C », si seulement si « a » appartient à « C » [3].

Il est également possible de raisonner sur les ontologies en utilisant un moteur d'inférence général tel que JESS qui peut être facilement intégré à Protégé. Les logiques de description permettent de définir les bases logiques des différents formalismes de représentation de la connaissance tant sur le plan de la représentation que sur le raisonnement. Dans les formalismes de représentation de la connaissance, il est souvent nécessaire de restreindre l'expressivité pour rendre certains types de raisonnement, tels que la classification automatique, faisables («tractables») [2].

Les programmes Racer, Jess<sup>1</sup> et Jena<sup>2</sup>, sont des outils qui aident à développer des applications à base d'ontologie. Jess et Jena sont basés sur la modélisation du raisonnement par règles de production. Jess fait un lien entre l'ontologie et les règles, parce qu'il sert à la modélisation du raisonnement de règles et Jena est une plateforme en Java qui a pour but de construire des applications permettant la manipulation d'ontologies décrites en RDF, RDFS et OWL [6].

---

<sup>1</sup> <http://herzberg.ca.sandia.gov/jess/>

<sup>2</sup> <http://jena.sourceforge.net>

## I.10 Les langages de construction d'ontologie

Les langages de construction d'ontologie ont été considérablement améliorés depuis la création du premier environnement ; parmi lesquels on cite :

### I.10.1 Le langage OWL :

Le langage OWL fournit des mécanismes pour créer tous les composants d'une ontologie : classes, instances, propriétés et axiomes. OWL repose également sur la syntaxe des triplets RDF et réutilise certaines des constructions RDFS. Comme en RDFS, les classes peuvent avoir des sous-classes, fournissant ainsi un mécanisme pour le raisonnement et l'héritage des propriétés. Par contre, en OWL, on distingue :

- les propriétés d'objet (Object Property) : les relations, qui relient des instances de classes à d'autres instances de classes. C'est l'équivalent des triplets RDF dont l'objet est une ressource.
- les propriétés type de données (data type property) : les attributs, qui relient des instances de classes à des valeurs de types de données (nombres, chaînes de caractères,...). C'est l'équivalent des triplets RDF dont l'objet est une valeur littérale.
- Les axiomes fournissent de l'information au sujet des classes et des propriétés, spécifiant par exemple l'équivalence entre deux classes. Donc OWL permet de définir des ontologies comme un jeu de définition de classes, de propriétés et de contraintes.

Toute classe définie dans une ontologie OWL est une sous-classe d'owl:Thing.

OWL a été fractionné en trois langages distincts chacune étant une extension de la précédente:

[11]

- **OWL Lite** : Convient aux utilisateurs qui ont principalement besoin d'une hiérarchie de classification et de contraintes simples. Ce sous langage reprend tous les constructeurs de RDF (c'est-à-dire fournit des mécanismes permettant de définir un individu comme instance d'une classe, et de mettre des individus en relation).
- **OWL DL** : Convient aux utilisateurs qui veulent le maximum d'expressivité, ce sous langage reprend tous les constructeurs d'OWL LITE, il permet tout entier positif dans les contraintes de cardinalité, et le tire son nom de sa correspondance avec les logiques de descriptions.
- **OWL FULL** : ce sous langage reprend tous les constructeurs d'OWL DL, et tous les constructeurs de RDF Schéma. Il permet d'utiliser une classe en position d'individu dans les constructeurs.

**I.11. Conclusion**

Nous avons présenté dans ce chapitre les concepts de base des ontologies. Nous avons présenté, également, quelques méthodologies et outils de construction d'ontologies, ainsi que les langages de construction d'ontologie.

Le chapitre suivant se focalise sur les travaux de recherche autour de l'alignement d'ontologies.

## II.1. Introduction

Le terme ontologie a été utilisé initialement en philosophie depuis le 19<sup>ème</sup> siècle. Dans ce domaine, il désigne l'étude de ce qui existe, l'ensemble des connaissances sur le monde [12].

Dans le domaine de la représentation des connaissances, les ontologies ne sont considérées que relatives aux différents domaines de connaissances. Elles répondent aux problèmes de représentation et de manipulation des connaissances. Les ontologies sont très utilisées dans la représentation des connaissances sur le Web [13].

Participer au web de données suppose d'être capable de s'interconnecter avec les données et ontologies déjà présentes et disponibles. Publier une nouvelle ontologie nécessite donc au préalable de la relier avec les "bonnes" ontologies publiées sur le web de données comportant des concepts similaires dans des domaines similaires [14].

La nécessité d'établir plusieurs ontologies pour une conceptualisation donnée vient du fait que le monde ne s'interprète pas de façon unique pour l'ensemble des individus et des machines [15].

Partant du fait que plusieurs connaissances peuvent prendre des représentations différentes, nous trouvons de nos jours plusieurs ontologies de domaine pour un même champ d'application. Il est alors nécessaire de disposer d'outils permettant de faire le lien entre les connaissances exprimées dans chacune des ontologies [16].

Relier deux ontologies consiste en fait à les aligner, trouver des correspondances entre les entités (concepts, propriétés ou instances) des deux ontologies [14].

Les techniques d'alignement représentent un cadre général, dans lequel plusieurs ontologies peuvent être exploitées [15]. L'alignement d'ontologies permet d'identifier un ensemble de correspondances entre les entités ontologiques appartenant aux deux ontologies à aligner. Les techniques d'alignement entre deux structures ont été développées dans de nombreux domaines. [17]

De nombreuses méthodes d'alignement dédiées aux ontologies ont vu le jour cette dernière décennie [18]. Les auteurs ont classé les méthodes d'alignement 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 [17].

Dans ce chapitre, nous allons présenter comment est né le besoin à l'alignement des ontologies, quelques domaines d'application de l'alignement des ontologies, la définition

de l'alignement avec quelques précisions terminologiques, Dimensions de l'alignement (input, processus de l'alignement, output ) puis nous allons présenter les techniques et les méthodes utilisées dans la littérature qui attaque le problème de recherche de la similarité, de la dissimilarité ou de la correspondance entre deux entités en général, qu'elles apparaissent dans des schémas, ou dans des ontologies.

## II .2. Comment est né le besoin à l'alignement des ontologies ?

L'évolution du web a permis d'intégrer la sémantique au web, en donnant un sens à ces ressources. Cette sémantique est représentée avec des ontologies (l'abstraction des connaissances d'un domaine). La réalisation de cette abstraction se fait de différentes manières selon la personne ou l'organisation qui modélise l'ontologie, ce qui va créer un ensemble d'ontologies hétérogènes [19].

La notion d'ontologie est devenue un élément clé dans toute une gamme d'applications faisant appel à des connaissances. Une ontologie est définie comme la conceptualisation des objets reconnus comme existant dans un domaine, de leurs propriétés et des relations les reliant. La structure d'une ontologie permet de représenter les connaissances d'un domaine sous un format informatique en vue de les rendre utilisables pour différentes applications [20].

Le besoin d'intégrer et d'analyser des grandes masses est présent dans de nombreux domaines d'applications. Le problème de l'alignement d'ontologies/schémas dont le résultat est un ensemble de correspondances entre différentes représentations du monde réel, est au centre du processus d'intégration des données.

Cependant, les ontologies à aligner ont des structures différentes et n'utilisent pas le même vocabulaire (c'est-à-dire des termes différents pour décrire les mêmes concepts) parce qu'elles ont été conçues indépendamment par différents développeurs suivant différents principes et modèles. En outre, la diversité de leur hétérogénéité : syntaxique, terminologique (ou lexicale) et structurelle, ainsi que leur taille et leurs formats rendent la tâche d'alignement d'ontologie très difficile [21]. Pour bien expliquer d'où vient cette hétérogénéité dans la modélisation d'ontologie, on va donner l'exemple suivant. On a deux personnes ou organisations qui veulent modéliser une ontologie pour le même domaine, c'est sûr que chacun entre eux va donner une modélisation selon le contexte dont il voit l'information ou la connaissance, aussi il va utiliser sa propre langue et ses propres termes pour réaliser la représentation terminologique de cette modélisation, par ex. : Une

personne américaine peut modéliser le sport « football » avec le terme « soccer » alors qu'une autre en Europe utilise le terme « football ». Aussi ils peuvent utiliser de différentes sources de connaissances, car il peut y avoir plusieurs ressources qui représentent la même connaissance. De même pour l'extraction des connaissances et informations peut être fait différemment à partir d'une même ressource [13].

Le choix d'une ontologie particulière ou l'exploitation de plusieurs d'entre elles devient difficile. La nécessité de les comparer, de passer de l'une à l'autre ou de les intégrer devient donc nécessaire. La tâche d'alignement d'ontologies consiste à générer le plus automatiquement possible des relations ou appariements entre les concepts de deux ontologies [20].

### II .2.1. Problème issu de l'hétérogénéité

Actuellement la notion d'ontologie constitue une des voies les plus prometteuses quant à la modélisation et à la représentation formelle des systèmes informatiques. Cependant, beaucoup de projets opérationnels et de recherche ont abouti à la création de multiples ontologies, parfois pour un même domaine. Cet engouement sans précédent, en particulier en ingénierie des connaissances, a un risque inévitable de voir une hétérogénéité sémantique entre les ontologies car elles ne sont homogènes ni dans leur structure, ni dans leurs sémantique [18].

L'hétérogénéité n'est pas seulement due à la divergence des domaines que peuvent couvrir les ontologies mais aussi aux formalismes requis pour leur développement [19]. Dans la littérature, Il existe plusieurs classifications d'hétérogénéité entre les ontologies [20], se basent sur l'étude du décalage sémantique et structurel qui peut exister entre les ontologies. D'autres classifications élucident le degré l'hétérogénéité selon les niveaux d'interopérabilité sémantique [19]. Ce qui suit, nous présentons les principales formes d'hétérogénéité en quatre niveaux : [20]

#### a) Le niveau syntaxique

Dans ce niveau, l'hétérogénéité se produit quand deux ontologies sont décrites avec deux langages ontologiques différents [19], elle est dépendante du choix du format de représentation. Celui-ci consiste à décrire et à coder les entités d'un domaine donné de manière à ce qu'une machine puisse les manipuler afin de raisonner ou de résoudre des problèmes [18], Cette hétérogénéité se produit aussi quand deux ontologies sont modélisées en utilisant des formalismes différents. Il est possible dans certains cas de



traduire les ontologies dans différents langages ontologiques à condition de préserver la signification [19].

Dans la littérature, il existe de nombreux langages informatiques spécialisés dans la création et la manipulation des ontologies comme (XML, RDF, OWL..) et chacun d'entre eux est basé sur une syntaxe propre et parfois ils sont utilisés pour représenter la même ontologie [18].

### **b) Le niveau terminologique:**

Dans ce niveau, la disparité est liée au processus de nommage des entités (classes, propriétés, relations) qui constituent une ontologie à partir d'un langage public, alors qu'elles désignent le même objet [18].

La cause d'une telle hétérogénéité revient à l'utilisation de différents langages naturels, ou des sous-langages techniques spécifiques à un domaine de connaissances bien déterminé [19]. Dans ce qui suit, nous citons des exemples de ce type d'hétérogénéité [20]:

- Synonymie** : Différents noms utilisés pour désigner une même entité.
- Polysémie** : Le même mot désigne plusieurs entités.
- Au contraire, le même terme peut représenter différents concepts ; l'homonymie est un problème qui nécessite bien souvent l'intervention humaine.
- Langage** : Des mots provenant de différentes langues (Français, Anglais, Italien, etc.) utilisés pour désigner une même entité.
- Variations syntaxiques** : variations syntaxiques du même mot (différentes prononciations, abréviations, utilisation des préfixes et des suffixes, etc.).
- Finalement, l'encodage des données au sein de l'ontologie diffère bien souvent, que ce soit pour les dates, les unités (monnaie, distances, etc.).

### **b) Le niveau conceptuel**

est appelée aussi hétérogénéité sémantique [22] ou la différence logique [23]. Chaque ontologiste, selon son domaine de prédilection pour représenter les concepts de manière propre et dans une hiérarchie spécifique [18]. Les divergences à ce niveau peuvent être résumées en trois aspects :

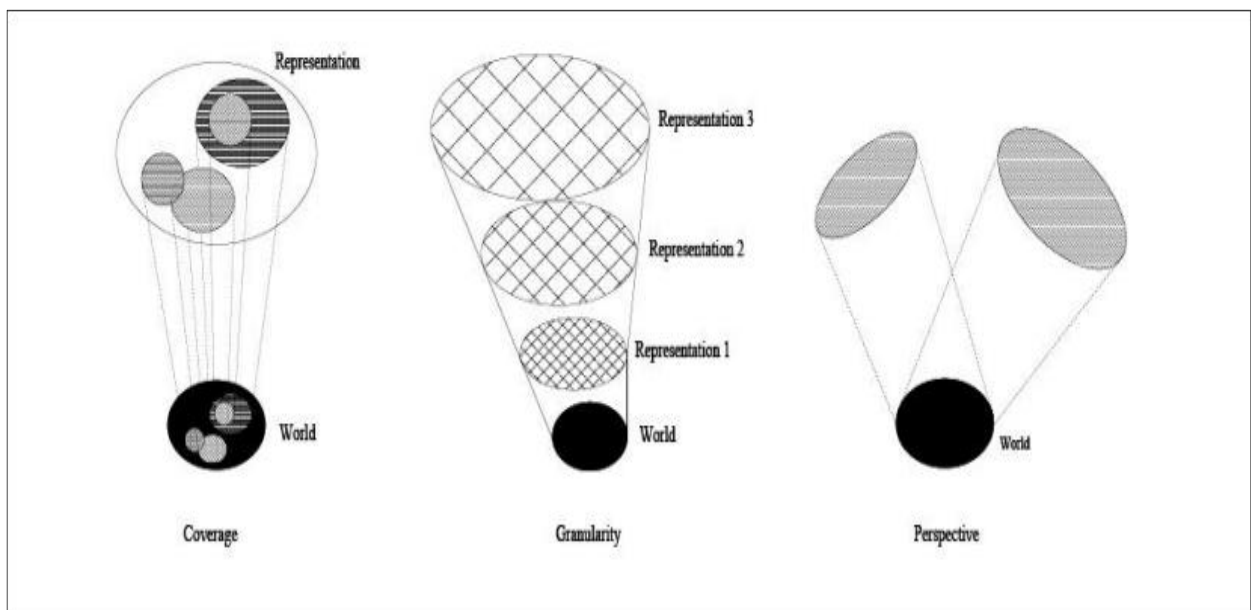
- **La couverture** : la différence entre deux ontologies peut être au niveau de la portée de la couverture du domaine décrit. Elles peuvent couvrir des sous ensembles de

connaissance d'un domaine donné ou alors des parties qui se chevauchent, par exemple : une ontologie sur le sport couvre le sport de la course automobile qu'une autre ignorerait complètement [20].

- **La granularité** : des ontologies peuvent représenter les mêmes connaissances [18], mais avec différents degrés d'expression des détails, par exemple : une ontologie concernée par les comptabilités va considérer le concept générique du document alors qu'une ontologie décrivant le domaine des bibliothèques va distinguer entre les différents types de documents : romans, nouvelles, biographies, manuscrits, etc... [13].

- **La perspective** : deux ontologies décrivent un même domaine, avec un même degré d'expression des détails, mais avec des points de vue différents. Par exemple : le concept de la chaleur chez un Norvégien sera forcément différent du même concept chez un Sénégalais [20].

La figure 1 est une représentation graphique de ces trois aspects, à travers lesquelles une ontologie peut différer d'une autre ontologie au niveau conceptuel :



**Figure II.1.** Les trois dimensions de l'hétérogénéité au niveau conceptuel [20].

### c) Le niveau sémantique ou pragmatique :

Ce type d'hétérogénéité intervient lorsqu'il y a différence d'interprétation de la même ontologie par différentes personnes ou différentes communautés [13].

Cette hétérogénéité s'intéresse à la manière dont les entités ontologiques sont interprétées par leurs utilisateurs. Ainsi, les entités ayant les mêmes interprétations sémantiques peuvent être interprétées de différentes manières par l'Homme. Ces différences

d'interprétation sont dues principalement à la diversité des contextes et des domaines d'application des ontologies. Par conséquent, la manière de mettre en œuvre les entités ontologiques influence leurs interprétations. De plus, ce type d'hétérogénéité reste difficile à détecter par la machine [19].

La compréhension des différentes formes d'hétérogénéité des ontologies est primordial pour la réussite de l'alignement de ces dernières, car il est très risqué de réaliser un alignement entre des entités, en se basant seulement sur les liens sémantiques [13]. Mais grâce à l'alignement des ontologies, le problème de l'hétérogénéité des données sur le web a été résolu, en plus on peut lier entre les ontologies même si l'hétérogénéité existe. Cette liaison se fait en cherchant les correspondances entre les ontologies. Afin de pouvoir réaliser quelques tâches comme l'échange, l'intégration... etc... [19].

### II.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 les schémas de base de données (de type relationnel, XML, ...), ontologies (décrites en OWL, RDFS, ...etc.) [18].

**Définition 1 :** L'alignement d'ontologies (ontology matching) a pour objectif de combler le fossé sémantique entre deux ontologies. Le résultat de ce processus, appelé alignement (alignment), est un ensemble de correspondances entre ces deux ontologies définis à une correspondance entre deux ontologies  $o$  et  $o'$  comme étant un 5-uplet  $(id, e, e', r, n)$  dans lequel  $id$  représente l'identifiant de la correspondance,  $e$  et  $e'$  sont des entités de  $o$  et  $o'$ ,  $r$  est une relation entre  $e$  et  $e'$ , et  $n$  est une valeur de confiance attribuée à la correspondance. La relation  $r$  est souvent un lien d'équivalence ( $\equiv$ ) ou de subsomption (plus spécifique-que,  $\sqsubseteq$ ), mais peut également être une disjonction ( $\equiv$ ), une généralisation (plus-général-que,  $\supseteq$ ), etc [13]

**Définition 2 :** « Une fonction qui prend en entrée deux schémas de base de données  $S1$  et  $S2$  et qui retourne un alignement entre l'ensemble des éléments de  $S1$  vers celui de  $S2$ . Chaque élément de l'alignement exprime une relation entre un élément de  $S1$  et un élément de  $S2$  » [18].

**Définition 3 :** « L'alignement de structures est le processus de mise en correspondance sémantique<sup>1</sup> des entités qui les composent. » [Euzenat et al., 2007b]

Ces structures peuvent être des ontologies, des schémas XML ou des bases de données. Les liens sémantiques comprennent les relations: d'équivalence ( $=$ ), de généralisation/spécialisation ( $\supseteq$ ,  $\sqsubseteq$ ), de chevauchement ( $\cap$ ) ou encore d'incompatibilité ( $\cap = \emptyset$ ). L'évaluation de la véracité de ces liens peut être booléenne ou par le biais d'autres mesures telles que : les probabilités, les mesures symboliques, les mesures de similarité [20].

**Définition 4 :** L'alignement d'ontologies consiste à chercher les concordances entre les concepts, les relations et les individus des diverses ontologies. Le but c'est de trouver les points de jonctions (les entités en commun.) qui permettront de concevoir des ponts entre ces ontologies, ou de procéder à d'autres opérations de manipulation comme la fusion ou l'intégration partielle. Pour cette raison, l'alignement est considéré comme une technique à base de toute autre opération de réconciliation entre les ontologies [25].

**Définition 5 :** L'alignement d'ontologies est le processus de mise en correspondance sémantique des entités qui les composent. Le processus est exécuté selon une stratégie ou une combinaison de techniques de calcul de mesures de similarité et utilise un ensemble de paramètres (ex : paramètres de pondération, seuils...) et un ensemble de ressources externes (ex : thésaurus, lexique...). Au final, nous obtenons un ensemble de liens sémantiques reliant les entités qui composent les ontologies. Ces derniers comprennent des relations d'équivalence, de généralisation/spécialisation, de chevauchement ou encore d'incompatibilité. De nombreux travaux ont été développés dans le domaine de l'alignement d'ontologies et portent sur les techniques de recherche de similarité et sur les outils ou sur les Framework qui les intègrent [26].

---

<sup>1</sup> Liens basés sur la signification des termes.

#### II.4. Quelques domaines d'application de l'alignement des ontologies:

Les champs d'application de l'alignement relèvent de plusieurs domaines. Ces domaines regroupent: l'intelligence artificielle, les systèmes d'information et les bases de données, l'E-commerce, la fouille d'information, le web sémantique, la communication dans les SMA (système multi-agents), data warehouse, l'intégration des schéma/Ontologies, etc. ) [27]. Le but par ces cas d'application est de montrer l'intérêt de l'alignement en général-celui des ontologies en particulier.

##### Cas 1 : La Communication entre agents

Les agents mobiles sont des entités caractérisées par leur mobilité, leur autonomie et leur capacité d'interaction. Les agents mobiles connaissent une utilisation accrue dans les domaines du E-commerce, la fouille de l'information etc [25].

Lorsque deux agents autonomes et conçus indépendamment se rencontrent, ils ont la possibilité de s'échanger des messages mais peu de chances pour se comprendre s'ils ne partagent pas le même langage et la même ontologie. L'alignement de leurs ontologies respectives intervient à ce niveau pour traduire les messages ou bien intégrer des passerelles entre leurs axiomes dans le modèle propre à chaque agent (pour pouvoir interpréter les messages) [13].

L'alignement d'ontologies permet aux systèmes multi-agents de s'attaquer au problème d'hétérogénéité sémantique dans la communication où chaque agent a sa propre représentation du monde [18]. (voir figure 2)

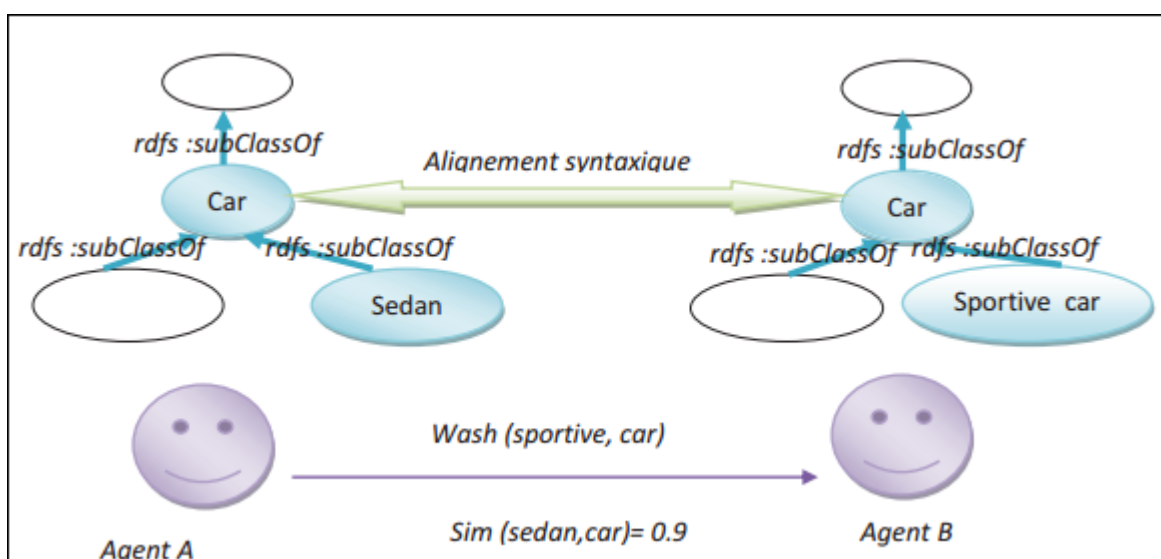


Figure II.2. Communication entre deux agents ayant deux ontologies distinctes [18].

**Cas 2 : Le Partage des informations dans les systèmes pair-à-pair (P2P)**

La technologie des réseaux Peer-to-Peer est un modèle de communication distribué dans lequel les pairs ont des capacités fonctionnelles équivalentes dans les échanges de données et de services [13].

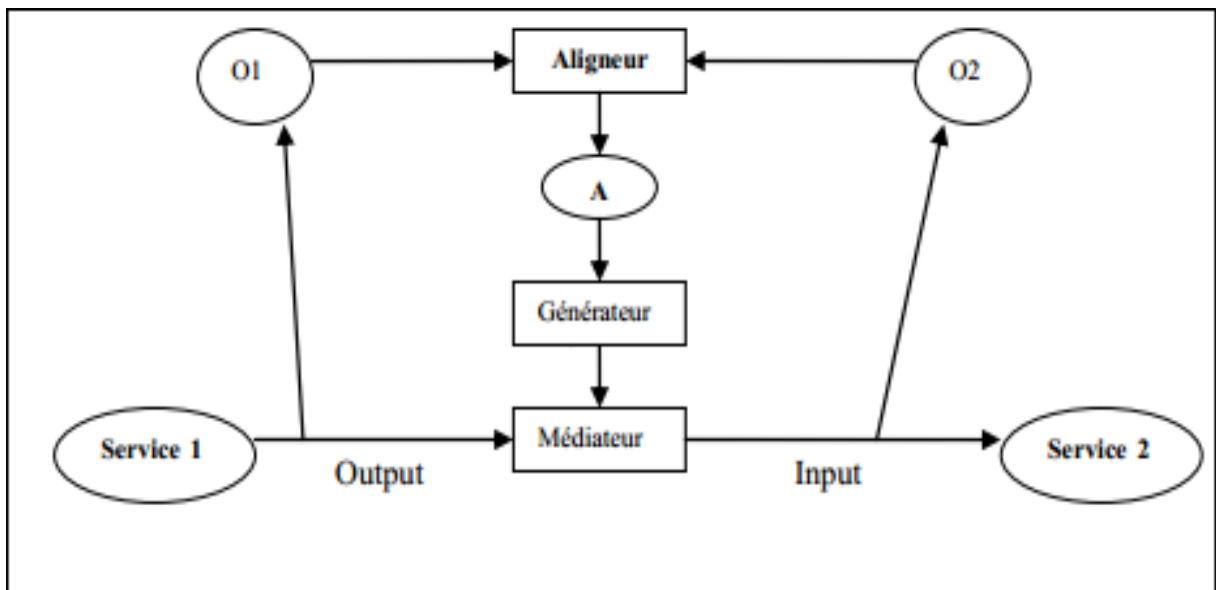
P2P a été déployée pour le partage des ressources entre les applications. Ces réseaux se caractérisent par leur dynamisme et leur flexibilité. Toute fois, les nœuds des réseaux (les pairs) sont autonomes en contenu, en langages de description des ressources et en la manière de les échanger. En effet, ils peuvent adopter des terminologies disparates (les problèmes de traduction des requêtes et de leurs réponses) ou faire usage à des ontologies. Pour assurer un échange efficace d'information entre les nœuds, il faudrait identifier et caractériser les relations de leurs schémas/ontologies respectives d'une manière instantanée lors de l'échange de données, autrement dit, réaliser un alignement d'ontologies [25].

**Cas 3 : La Composition des services web**

Un web service est un programme informatique permettant la communication et l'échange de données entre applications et systèmes hétérogènes dans des environnements distribués. De nos jours les entreprises ont adopté les Web Services comme instrument nécessaire à l'intégration de leur métier dans l'E-commerce [18]. La découverte et l'intégration des services Web consiste à découvrir un service Web particulier et/ou combiner plusieurs services pour fournir un autre service plus complexe.

Les services web sémantiques fournissent un moyen plus riche et plus précis de décrire les services à travers les langages de représentation des connaissances et des ontologies. Ces derniers peuvent jouer un rôle primordial pour décrire explicitement la sémantique des services. Toutefois, la description de tels services fait référence à des ontologies multiples généralement exprimées dans des modèles et des langages hétérogènes. Une telle hétérogénéité entraîne souvent des ambiguïtés sémantiques. Par conséquent, l'opération d'intégration des services nécessite l'établissement de correspondances entre les termes d'ontologies [25].

Par exemple, un service web fournit la description de son output à l'aide d'une ontologie et un autre service web utilise une seconde ontologie pour d'écrire son input, aligner ces deux ontologies permettrait de vérifier si ce qui a été délivré par le premier service correspond à ce qui était attendu par le second service et cela grâce à un médiateur entre ces deux services, généré à partir de l'alignement des deux ontologies précédemment citées [20].



**Figure II.3.** Composition d'un service web [20].

#### Cas 4 : Le Web Biomédical

Le domaine biomédical est par ailleurs caractérisé par l'existence de nombreux standards terminologiques, thésaurus, et langages, partagés par les communautés biomédicales, qu'ils soient généralistes ou dédiés à un domaine de spécialité, ainsi que de riches banques de documents généralistes ou plus spécifiques (par exemple, sur les maladies rares ORPHANET, RARE DISEASE), qui représentent un acquis important mais aussi une contrainte forte puisqu'il n'est pas envisageable de les ignorer. Les ontologies doivent fournir les concepts et les relations utilisés pour le marquage sémantique des données en vue du Web Sémantique avec une signification partagée et réutilisable pour différentes applications et différents usagers.

L'alignement des ontologies aide à trouver rapidement sur le Web, avec le minimum de bruit possible, une information scientifique récente, a un intérêt non seulement pour le chercheur qui doit accéder à des bases hétérogènes et réitérer régulièrement les interrogations sur ces bases, pour les patients à la recherche d'informations, mais aussi dans la pratique médicale quotidienne où médecins et industriels pharmaceutiques sont amenés à rechercher de l'information. De plus l'alignement des ontologies joue un rôle déterminant dans la mise en place d'une recherche biomédicale associant un plus grand nombre d'acteurs, en facilitant la constitution d'entrepôts de données ou d'entrepôts d'informations fédérés, articulés autour d'ontologies communes [20].

### Cas 5 : la recherche dans les catalogues d'E-commerce

Les applications B2B ont recours à des catalogues électroniques sous forme de taxonomies hiérarchisées de concepts avec leurs attributs afin de stocker et d'indexer le volume important de produits. En effet, des standards comme UNSPSC6 est largement utilisé pour représenter les besoins du vendeur, alors que le standard ect@ss7 est utilisé pour représenter les besoins de l'acheteur. D'un autre côté, il faudrait permettre à des applications de fouiller et de restituer de l'information pertinente à partir de ces catalogues dont le contenu n'est pas nécessairement décrit dans le même modèle (le Cas des deux standards UNSPESC et ecl@ss). Pour ce faire, il faudrait que ces catalogues soient alignés moyennant des procédés automatiques pour que les outils et les engins de recherche puissent accéder à leurs contenus. [25]

### II.5. Le processus d'alignement

Le processus d'alignement est une tâche pendant laquelle à partir d'une ontologie  $O$  et une autre  $O'$ , il détermine un alignement  $A'$  entre ces deux ontologies, cette tâche est réalisée en utilisant une stratégie ou une combinaison de techniques d'alignement. On verra dans la suite de ce chapitre ces techniques en détail [19]. L'alignement regroupe trois dimensions : l'input, le processus de l'alignement et l'output.

**L'input :** est constitué essentiellement des structures destinées à être alignées et qui peuvent être, des schémas de base de données (de type relationnel, XML, objet), des ontologies (décrites en OWL, RDFS,...etc.) ou des instances d'une base de données ou d'une ontologie.

**Remarque :** L'input ou bien l'entrée peut être enrichi par un alignement en entrée (qui aurait besoin d'être complété par une nouvelle itération d'alignement) [13].

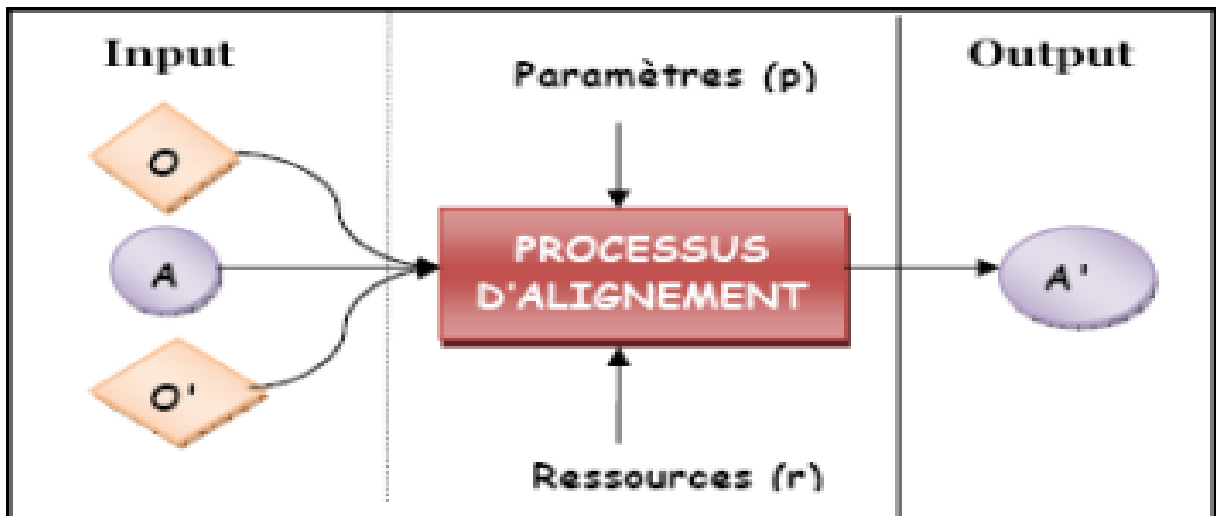
#### Le processus d'alignement :

Comme le montre la Figure 4 ; l'alignement peut être considéré comme une fonction  $f$ , qui à partir d'une paire d'ontologies  $O$  et  $O'$ , un alignement en entrée  $A$  (optionnel), un ensemble de paramètres  $p$  (ex : paramètres de pondération, seuils ...) et un ensemble de ressources externes  $r$  (ex : thésaurus, lexique, etc.), détermine un alignement  $A'$  entre ces deux ontologies :

$$A'=f(O, O', A, p, r)$$



Ceci peut être représenté schématiquement de la manière suivante Figure II.4 [20]:



**Figure II.4.** Les trois dimensions de l'alignement [13].

**L'output** : est un ensemble d'alignement reliant les entités qui constituent les deux ontologies. Un alignement est décrit comme un ensemble de cinq éléments :  $\langle \text{id}, e, e', r, n \rangle$  telle que ;

- **Id** : un identifiant de l'alignement
- **e** : une entité à aligner et qui appartient à O (classe, propriété, contrainte, instance)
- **e'** : une entité à aligner appartenant à l'ontologie O'
- **r** : la relation qui lie e à e'.
- **n** : la mesure de confiance de la relation r, généralement une valeur réelle comprise dans l'intervalle [0,1], Plus le n est proche du 1, plus la relation est considérée comme étant forte [19].

**L'output est caractérisé par :**

**La multiplicité** : (contraintes sur les relations entre les entités des deux ontologies). Si on considère, d'une part, les valeurs suivantes:

1 : une et une seule relation

? : de 0 à 1 relation

+ : de 1 à plusieurs relations

\* : de 0 à plusieurs relations

} à partir d'une entité d'ontologie

D'autre part, les deux orientations possibles d'un alignement entre deux ontologies ( $O \rightarrow O'$ ) et ( $O' \rightarrow O$ ), la multiplicité peut prendre les valeurs suivantes :

1 : 1, 1 : ?, ? : 1, 1 : +, + : 1, 1 : \*, \* : 1, ? : ?, ? : +, + : ?, ? : \*, \* : ?, + : \*, \* : +, + : +, \* : \*

La figure II.5 : montre quelques exemples sur les configurations de multiplicité entre deux ontologies, constituée chacune de trois entités [13].

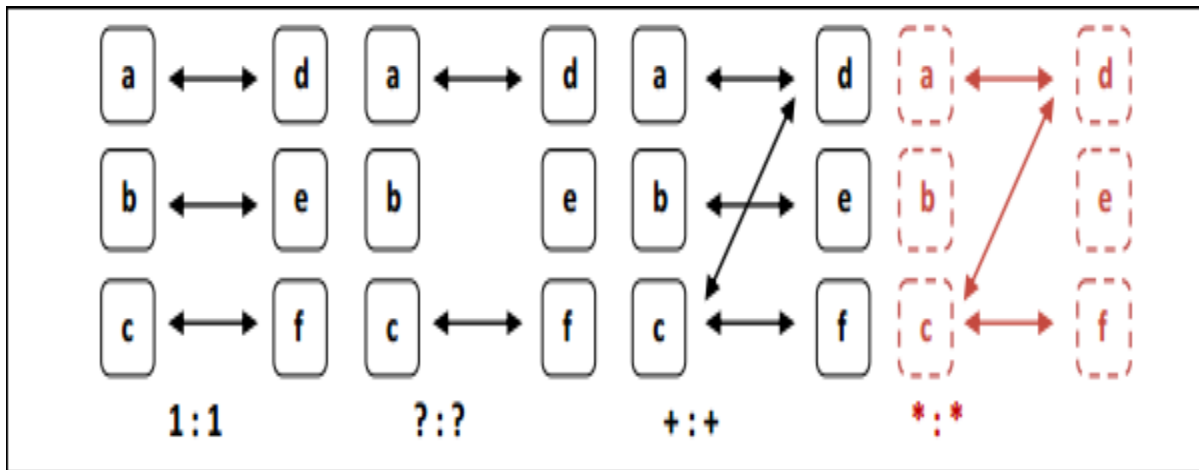


Figure II.5. Exemples de configurations de multiplicité entre entités de deux ontologies [20].

## II.6. Les techniques d'alignement :

Les ontologies à aligner ont des structures différentes et n'utilisent pas le même vocabulaire (c'est-à-dire des termes différents pour décrire les mêmes concepts). En outre, la diversité de leur hétérogénéité : syntaxique, terminologique (ou lexicale) et structurelle, ainsi que leur taille et leurs formats rendent la tâche d'alignement d'ontologie très difficile.

Donc, nous allons examiner les techniques et les méthodes utilisées dans la littérature qui s'attaquent au problème de recherche de la similarité, de la dissimilarité ou de la correspondance entre deux entités en général pour identifier les liens de correspondance entre les ontologies [21].

avant de présenter les méthodes de mesure de la similarités, nous tenons à définir la similarité comme suivant :

- **La similarité :**

La similarité est un concept important et largement utilisé, qui est lié à un domaine d'application particulier ou à une forme de représentation des connaissances, on peut trouver en psychologie ou en mathématiques. En psychologie sociale, la similarité se rapporte

à comment les attitudes, les valeurs, les intérêts et la personnalité correspondent entre les personnes. En mathématiques, plusieurs relations d'équivalence sont appelées similarité, En topologie, la similarité est une fonction telle que sa valeur est plus grande quand deux points sont plus proches (contrairement à la distance, qui est une mesure de dissimilarité : plus les points sont proches, plus la distance est petite).

Dans notre contexte, la notion de similarité sémantique est vue comme celle de la similarité topologique en mathématiques, où on l'associe à une fonction, appelée fonction de la similarité [13].

**Définition 1 (Similarité) :** La similarité est une fonction d'une paire d'entités à un nombre réel exprimant la similarité entre ces deux entités telle que : [19]

- $\forall a, b \in O, S(a, b) \geq 0$  (positivité)
- $\forall a, b, c \in O, S(a, a) \geq S(b, c)$  et  $S(a, a) = S(a, b) \leftrightarrow a = b$  (autosimilarité ou maximalité)
- $\forall a, b \in O, S(a, b) = S(b, a)$  (symétrie)
- $\forall a, b, c \in O, S(a, b) = S(b, c) \rightarrow S(a, b) = S(a, c)$  (transitivité)
- $\forall a, b \in O, S(a, b) \leq \infty$  (finitude)

La dissimilarité est parfois utilisée au lieu de la similarité. Elle est définie de manière analogue à la similarité, sauf qu'elle n'est pas transitive :

**Définition 2 (Dissimilarité) :** La dissimilarité  $DS : O \times O \rightarrow \mathbb{R}$  est une fonction d'une paire d'entités à un nombre réel exprimant la dissimilarité entre ces deux entités telle que: [17]

- $\forall a, b \in O, DS(a, b) \geq 0$  (positivité)
- $\forall a, b, c \in O, DS(a, a) \leq DS(b, c)$  et  $DS(a, a) = 0$  (minimalité)
- $\forall a, b \in O, DS(a, b) = DS(b, a)$  (symétrie)
- $\forall a, b \in O, DS(a, b) \leq \infty$  (finitude)

La distance est une mesure utilisée aussi souvent que les mesures de similarité.

Elle mesure la dissimilarité de deux entités, elle est inverse de la similarité : si la valeur de la fonction de similarité de deux entités est élevée, la distance entre ces entités est petite et vice-versa. Elle est donc définie dans comme suit :

**Définition 3 (Distance) :** La distance  $D : O \times O \rightarrow R$  est une fonction de la dissimilarité satisfaisant la définitivité et l'inégalité triangulaire. [19]

- $\forall a, b \in O, D(a, b) = 0 \Leftrightarrow a = b$  (définitivité)
- $\forall a, b, c \in O, D(a, b) + D(b, c) \geq D(a, c)$  (inégalité triangulaire)

Les valeurs de similarité sont souvent normalisées. Si la valeur de similarité et la valeur de dissimilarité entre deux entités sont normalisées, notées S et DS , alors on a  $1 = DS + S$ . [19]

**Définition 4 (Normalisation) :** Une mesure est une mesure normalisée si les valeurs calculées par cette mesure ne peuvent varier que dans un intervalle de 0 à 1. Ces valeurs calculées sont appelées valeurs normalisées. Les fonctions du calcul sont appelées fonctions normalisées et notées f [13].

### II.6.1. Les méthodes terminologiques

Ces méthodes sont employées pour calculer la valeur de similitude des entités textuelles (comparent les chaînes de caractères afin d'en déduire la similarité (ou dissimilarité)), telles que des noms, des méta-données sur les noms, des étiquettes, des commentaires, des descriptions... [26]. Ces méthodes peuvent encore être divisées en deux sous-catégories: méthodes syntaxiques et linguistiques.

#### II.6.1.1. Méthodes syntaxiques

Ces méthodes analysent la structure des chaînes à comparer, plus l'ordre des caractères dans la chaîne, le nombre d'apparitions d'une lettre dans une chaîne pour concevoir des mesures de la similarité, plus elles partageront de caractères en commun. Par contre, elles n'exploitent pas la signification des termes.

Généralement ces méthodes exigent un prétraitement qui consiste à normaliser les chaînes à comparer avant de les fournir aux fonctions calculant la similarité [18]. Par exemple, les mesures dans cette catégorie retournent une grande valeur de similarité

(jusqu'à 1) si elles comparent les termes « élève » et « élèves », mais une petite valeur, voire la valeur 0, si elles comparent les termes « élève » et « écolier ».

### **II.6.1.2. Méthodes linguistiques :**

Les méthodes linguistiques utilisant des ressources externes (dictionnaires, taxonomies,...), ces méthodes permettent de déterminer la similarité entre deux entités. Ces entités sont représentées par des termes (ou mots). La similarité est calculée à partir des liens sémantiques déjà existants dans les ressources externes. Nous pouvons déduire cette dernière entre ces termes en s'appuyant sur des connaissances de la langue naturelle (les méthodes intrinsèques) et / ou sur des vocabulaires et des dictionnaires (les méthodes extrinsèques) [17].

#### **II.6.1.2.1. Méthodes intrinsèques :**

Les informations intrinsèques sont des propriétés linguistiques internes des termes, telles que des propriétés morphologiques ou syntaxiques.

Une même entité ou un même concept peut être référencé par plusieurs termes (synonymie) ou par plusieurs variantes d'un même terme. Les méthodes intrinsèques fonctionnent avec le principe de chercher la forme canonique ou représentative d'un mot ou d'un terme (lemme) à partir de ses variantes linguistiques (lexème).

La similarité entre deux termes est donc décidée en comparant leurs lemmes. Par exemple, le résultat de la mesure de similarité exacte de deux mots « ran » et « running » sera égal à 0 (c. -à-d. ils sont différents), alors que le résultat de la même mesure pour les lemmes de ces mots sera égal à 1, ce qui indique que « ran » et « running » sont similaires [20]. La recherche du lemme d'un mot peut être effectuée dans un dictionnaire.

Une autre approche qui est automatique et plus légère et plus efficace est d'utiliser des stemmers. Un stemmer est un programme ou un algorithme qui détermine la forme radicale à partir d'une forme infléchiée ou dérivée d'un mot donné. Les radicaux (stems) trouvés par les stemmers n'ont pas besoin d'être identiques à la racine morphologique du mot. Il suffit que les mots similaires soient associés à un même radical, même si ce radical n'est pas une racine de mot valide. Un stemmer pour le français, par exemple, devrait identifier les chaînes de caractères « maintenaient », « maintenait », « maintenant », ou « maintenir » comme basées sur la racine "mainten".

Une approche plus complexe pour déterminer le radical exact d'un mot est la lemmatisation. Ce processus comprend la détermination de la partie du discours

(catégorie lexicologique) d'un mot, et l'application des règles de normalisation différentes pour chaque partie du discours. Cette approche exige la connaissance de la grammaire d'une langue, des règles différentes... Elle est donc lourde, compliquée et difficile à implémenter [19].

Le premier stemmer publié a été écrit par Julie Beth Lovins. Ensuite un autre stemmer a été développé par Martin Porter. Ce dernier est très largement utilisé, et est devenu l'algorithme standard utilisé pour chercher des radicaux dans la langue anglaise [20].

### **II.6.1.2 .2. Méthodes extrinsèques**

Les informations extrinsèques exploitent des ressources externes telles que des dictionnaires ou des vocabulaires .Ces méthodes calculent la valeur de similarité entre deux termes en employant des ressources externes qui regroupent des dictionnaires, des lexiques ou des vocabulaires. La similarité entre deux termes est calculée en exploitant les liens sémantiques existants dans ces ressources externes. Ces liens regroupent les synonymes (pour l'équivalence), des liens hyponymes/ hyperonymes (pour la subsomption). Par exemple, à l'aide des ressources des synonymes, «élève» et «écolier» sont dites similaires. Typiquement, WordNet, un système lexicologique, est employé pour trouver des relations telles que la synonymie entre des termes, ou pour calculer la distance sémantique entre ces termes, en utilisant des liens sémantiques dans WordNet, afin de décider s'il existe une relation entre eux. Les ressources externes utilisées dans les méthodes extrinsèques peuvent aussi être des vocabulaires ou des dictionnaires multilingues [13].

### **II.6.2. Méthodes structurelles :**

Ce sont les méthodes qui déduisent la similarité entre deux entités en exploitant leurs positions dans une hiérarchie et en fonction des informations structurelles. En effet, les entités sont reliées entre elles par des liens sémantiques ou syntaxiques. On peut distinguer entre deux méthodes structurelles. L'une qui n'exploite que des informations concernant des attributs d'entités (les méthodes structurelle interne) et l'autre qui considère des relations entre des entités (les méthodes structurelles externes) [17].

#### **II.6.2.1. Méthodes structurelles internes:**

elles calculent la similarité entre deux concepts en exploitant les informations relatives à leur structure interne, dans la plupart des cas, ce sont des informations

concernant des attributs de l'entité (restrictions et cardinalités sur les attributs, valeurs des instances, ...).

Les méthodes de cette classe exploitent ces attributs pour déduire les similarités elles couramment utilisé dans les cas où les entités ont des définitions intentionnelles précises. Les premiers systèmes qui se basent sur ce principe sont les systèmes d'intégration et l'alignement de schémas de bases de données puis reprise dans le contexte d'alignement d'ontologie [18].

Dans le domaine des bases de données, plusieurs méthodes ont été proposées pour calculer la similarité entre deux éléments de deux schémas de base de données, en se basant sur les contraintes à propos de ces éléments [13].

### II.6.2.2. Méthodes structurelles externes

Contrairement aux méthodes structurelles internes, qui exploitent des informations des attributs d'entité, Les méthodes structurelles externes traitent la structure externe de l'entité et exploitent des relations entre elles-mêmes, donc qui sont souvent des relations spécifique-générique ou hyponyme-hyperonyme (is-a ou spécialisation), et la relation de composition meronymie-holonymie (partie-tout). L'ensemble de ces relations nous donne tout la hiérarchie de l'ontologie qu'est souvent représenté par des graphes. La comparaison de similarité entre deux entités de deux ontologies peut être basée sur la position des entités dans leurs hiérarchies.

Ces méthodes ont la possibilité d'aligner deux entités en s'appuyant sur leurs voisines. Une entité d'une ontologie peut avoir trois type de voisinage : ses super-entités, ses sous-entités, ses sœurs. L'idée de base est que deux entités sont similaires, si leurs voisines sont similaires [19].

### II.6.3 Les méthodes extensionnelles

Afin de déduire la similarité entre deux entités, les méthodes extensionnelles comparent leurs extensions c-à-d leurs ensembles des instances.

Ces méthodes déduisent la similarité entre deux entités qui sont notamment des concepts ou des classes en analysant leurs extensions.

Dans le cas où les ensembles d'instances partagent une partie commune, on peut avoir des mesures extensionnelles qui emploient des opérations de l'ensemble, telles que la distance de Hamming ou la mesure de Jaccard. Fondamentalement, Il y a la mesure de Hamming compte un nombre d'éléments différents entre deux ensembles à comparer et la

mesure de Jaccard est le rapport entre l'intersection des ensembles et leur union. Ces mesures peuvent être adaptées pour construire des mesures extensionnelles [20].

**Définition 1 :** (Distance de Hamming, version adaptée pour les ensembles des instances). Soit  $S$  et  $T$  deux ensembles. La distance de Hamming (appelée aussi la différence symétrique) entre  $S$  et  $T$  est une fonction de la dissimilarité [28].

$$DS_{Hamming}(S, T) = \frac{|S \cup T - S \cap T|}{|S \cup T|}$$

**Définition 2 :** (Distance de Jaccard, version adaptée pour les ensembles des instances). Soit  $s$  et  $t$  deux chaînes de caractères et  $S$  et  $T$  deux ensembles des caractères de  $s$  et  $t$  respectivement. Soit  $P(x)$  la probabilité d'une instance aléatoire être dans l'ensemble  $X$ . La distance de Jaccard est une fonction de la dissimilarité  $DS_{jaccard} : 2^E * 2^E \rightarrow [0, 1]$  telle que : [29]

$$DS_{jaccard}(s, t) = 1 - \frac{P(S \cap T)}{P(S \cup T)}$$

Ces mesures produisent la similarité de deux entités qui est en fait la similarité entre les deux ensembles de leurs instances en se basant sur la comparaison exacte des éléments dans deux ensembles.

#### II.6.4. Les méthodes sémantiques

Les méthodes sémantiques disposent d'un modèle théorique utilisé pour justifier leurs résultats. Ces méthodes se basent principalement sur deux approches. La première approche repose sur le raisonnement dans les logiques de descriptions tandis que la deuxième approche regroupe les méthodes de déduction afin de déduire la similarité entre deux entités. [25]

##### II.6.4. 1. Les techniques basées sur les ontologies externes

Lorsque deux ontologies doivent être alignées, il est préférable que les comparaisons se fassent selon un capital de connaissances communes. Ce type de techniques s'intéresse à l'utilisation d'ontologie formelle intermédiaire pour répondre à ce besoin. Cette ontologie va définir un contexte commun pour les deux ontologies à aligner.

L'idée est que cette ontologie, avec une couverture appréciable du domaine d'intérêt des ontologies (ou une ontologie encore plus générale comme une ontologie de haut



niveau), va permettre de lever le voile sur les ambiguïtés concernant les différentes significations possibles des termes. Des exemples d'ontologies intermédiaires: FMA<sup>2</sup> " the Foundational Model of Anatomy ", CYC ontology et SUMO<sup>3</sup> "the Suggested Upper Merged Ontology". [20]

---

<sup>2</sup> <http://sig.biostr.washington.edu/projects/fm/>

<sup>3</sup> <http://www.ontologyportal.com>

#### **II.6.4.2. Les techniques déductive:**

Les méthodes sémantiques se basent sur des modèles de logique (tels que la satisfiabilité propositionnelle (SAT), la SAT modale ou les logiques de descriptions. emploient des techniques issues de la satisfiabilité propositionnelle (SAT). Ces techniques permettent la vérification de la validité d'un ensemble de formules propositionnelles. Ce dernier est construit en traduisant des relations déjà connues et des relations à vérifier entre des entités vers des formules propositionnelles. Étendent les méthodes proposées vers le modèle de la SAT modale.

Les techniques des logiques de description (le test de subsumption) peuvent être employées. Elles permettent de vérifier les relations sémantiques entre les entités telles que l'équivalence (la similarité est égale à 1), la subsumption (la similarité est comprise entre 0 et 1) ou l'exclusion (la similarité est égale à 0). Elles assurent aussi la déduction de la similarité de deux entités. [13]

#### **II.6.5. Les méthodes d'alignement combinées :**

Il existe plusieurs vues ou aspects sous lesquels une entité peut être observée ou considérée sous plusieurs différents aspects, soit en s'appuyant sur, son nom, ses attributs, ou sur ses relations avec d'autres entités. La similarité entre deux entités peut donc être calculée en se basant sur plusieurs aspects, et Sur chaque aspect, les caractéristiques d'une entité sont comparées avec les caractéristiques correspondantes d'une autre par une des mesures de similarité de base présentées dans les méthodes de base pour

mesurer la similarité, cela retourne une valeur de la similarité (ou de la dissimilarité /distance).. Par conséquent, il faut un moyen pour combiner ces valeurs de similarité obtenues pour chaque aspect afin de produire une valeur de similarité unique pour chaque deux entité à comparer. Dans ce qui suit, on va présenter quelques approches qui existent dans la littérature. La distance de Minkowski entre deux entités est définie comme suivante : [19]

**Définition 1 (Distance de Minkowski).** Soit  $O$  l'ensemble d'objets qui peuvent être analysés dans  $n$  dimensions. Soit  $x$  et  $y$  deux objets dans  $O$ . La distance de Minkowski entre  $x$  et  $y$  est une fonction de la dissimilarité  $DS_{Minkowski} : O * O \rightarrow \mathbb{R}$  telle que : [29]

$$DS_{Minkowski}(x, y) = \sqrt[p]{\sum_{i=1}^n DS(x_i, y_i)^p}$$

Cette distance est une mesure généralisée avec différentes valeurs de  $p$ ,  $p \geq 1$ .

Quand  $p$  est égale à 1, elle devient la distance de « city block » et quand  $p = 2$  elle devient la distance euclidienne. La distance de Chebyshev (appelée aussi la distance de valeur maximum) est un cas spécial de la distance de Minkowski avec  $p = \infty$  :  $DS_{chebyshev}(X, Y) = \max_i DS(X_i, Y_i)$ .

Cette mesure n'est une fonction linéaire que quand  $p = 1$  ou  $p = \infty$ . Dans le cas où  $p = 1$ , une variante de cette mesure avec des poids est souvent utilisée. Le bon côté de cette variante est que nous pouvons contrôler l'influence (ou l'importance) de chaque dimension sur la valeur finale de la distance. Les dimensions les plus importantes seront associées aux poids les plus élevés, donc les valeurs de ces dimensions influenceront mieux à la valeur agrégée finale.

**Définition 2 (Somme pondérée).** Soit  $O$  l'ensemble d'objets qui peuvent être analysés dans  $n$  dimensions. Soit  $x$  et  $y$  deux objets dans  $O$ . Soit  $w_i$  le poids de la dimension  $i$ . Soit  $DS(X_i, Y_i)$  la dissimilarité de la paire des objets à la dimension  $i$ . La somme pondérée entre  $x$  et  $y$  est une fonction de la dissimilarité  $DS_{sp} : O * O \rightarrow \mathbb{R}$  telle que : [30]

$$DS_{sp}(x, y) = \sum_{i=1}^n \omega_i * DS(x_i, y_i)$$

En général, la somme des poids est égale à 1 :  $\sum_{i=1}^n w_i = 1$ , dans ce cas, nous avons la version normalisée de  $DS_{sp}$ .

Une autre mesure analogue à la somme pondérée est le produit pondéré. Cependant, un inconvénient de cette mesure est que le résultat sera égal à 0 si une des dimensions est égale à 0.

**Définition 3 (Produit pondéré).** Soit  $O$  l'ensemble d'objets qui peuvent être analysés dans  $n$  dimensions. Soit  $x$  et  $y$  deux objets dans  $O$ . Soit  $w_i$  le poids de la dimension  $i$ . Soit la dissimilarité de la paire des objets à la dimension  $i$ . Le produit pondéré entre  $x$  et  $y$  est une fonction de la dissimilarité  $DS_{pp} : O * O \rightarrow \mathbb{R}$ . [31]

$$DS_{sp}(x, y) = \prod_{i=1}^n DS(x_i, y_i)^{w_i}$$

Toutes les approches, qui combinent des valeurs de similarité calculées par différentes mesures, emploient la méthode de la somme pondérée (Définition 2). Cependant, certaines approches (par exemple Anchor-PROMPT) déduisent des alignements en examinant des critères heuristiques sans utiliser des méthodes de combinaison des similarités.

## II.7. Quelques approches de l'alignement d'ontologies

Les techniques d'alignement jouent un rôle capital dans la construction d'un lien sémantique entre les ontologies d'un même domaine. Donc la diversité d'approches vient du fait que chaque approche utilise les méthodes adéquates selon les types d'ontologies à aligner. Dans cette partie, nous allons analyser des approches déjà existantes dans la littérature qui concernent le problème d'alignement d'ontologies. Nous allons voir un petit résumé sur quelques approches :

### ➤ ASCO3 :

Représente les ontologies sous forme d'un graphe étiqueté, orienté et cyclique, appelé O-Graphe. Les entités de l'ontologie (classes, relations, instances) sont des nœuds du graphe, deux nœuds sont liés par un arc orienté et étiqueté par la primitive de OWL. Un O-Graphe est un 4-uplet  $og = (V, E, a, b)$ , où :

- $V$  est l'ensemble fini de sommets (également appelés les nœuds)
- $E \subseteq V \times V$  est l'ensemble d'arcs

- $\alpha : V \rightarrow L$  est une fonction affectant une étiquette à chaque sommet
- $\beta : E \rightarrow L$  est une fonction affectant une étiquette (type) à chaque arc.

Arc  $(u,v)$  est un arc orienté, commençant au nœuds  $u$  et se terminant au nœuds  $v$ . [32]

➤ **QOM** : L'approche QOM (Quick Ontology Matching) est basée sur l'approche NOM (Naive Ontology Matching) cette dernière a montré qu'elle est effective, mais inefficace, donc on a pensé à l'optimiser, la chose qui a donné naissance à l'approche QOM. Cette approche comme son prédécesseur exploite RDF-triples, et pour mesurer la similarité entre les ontologies à aligner elle utilise la similarité terminologique et structurelle. Et, après la phase de l'analyse de la similarité et ses interprétations, de nouvelles décisions doivent être prises, comme : quels alignements candidats doit-on ajouter à l'agenda pour l'itération suivante.

Cette approche a montré de très bons résultats en termes de temps d'exécution ( $n \cdot \log(n)$  au lieu de  $n^2$ ,  $n$  étant le nombre d'entités dans les ontologies) en comparaison avec d'autres approches de la même classe de complexité. [19]

➤ **Anchor-PROMPT** : Construit un graphe étiqueté orienté représentant l'ontologie à partir de la hiérarchie des concepts (appelés classes dans l'algorithme) et de la hiérarchie des relations (appelées slots dans l'algorithme), où les nœuds dans le graphe sont des concepts et les arcs dénotent des relations entre les concepts (les étiquettes des arcs sont les noms des relations). Cet algorithme décide si ces concepts sont sémantiquement similaires. Une liste initiale des paires d'ancres (des paires de concepts similaires) définies par les utilisateurs ou automatiquement identifiées par la mise en correspondance lexicologique sert d'entrée à l'algorithme.

Cependant, Anchor-PROMPT ne cherche que des correspondances des concepts, pas des correspondances des relations. [32]

➤ **SODA** : La nouvelle méthode d'alignement, SODA (Structural Ontology OWL-DL Alignment), implémente un nouvel algorithme d'alignement d'ontologies OWL-DL. La nouvelle méthode d'alignement repose sur le calcul des mesures de similarité. Il définit deux modèles de calcul de similarité (locale et globale). La méthode combine les mesures de similarité locale (terminologique et structurelle) pour l'évaluation de la similarité globale. Elle permet de générer un alignement exploitant l'aspect structurel du voisinage des entités à apparier.

La méthode SODA est performante sur les petites et moyennes ontologies et n'est pas encore très bien adaptée pour les ontologies de grande taille. [16]

➤ **TaxoMap** : TaxoMap est un outil d'alignement qui a pour objectif de permettre un accès unifié via le Web aux documents d'un même domaine d'application. Il est adapté au traitement de taxonomies dont les structures sont hétérogènes et dissymétriques. L'objectif de TaxoMap est de mettre en correspondance les concepts de la taxonomie la moins structurée, la taxonomie source. [18]

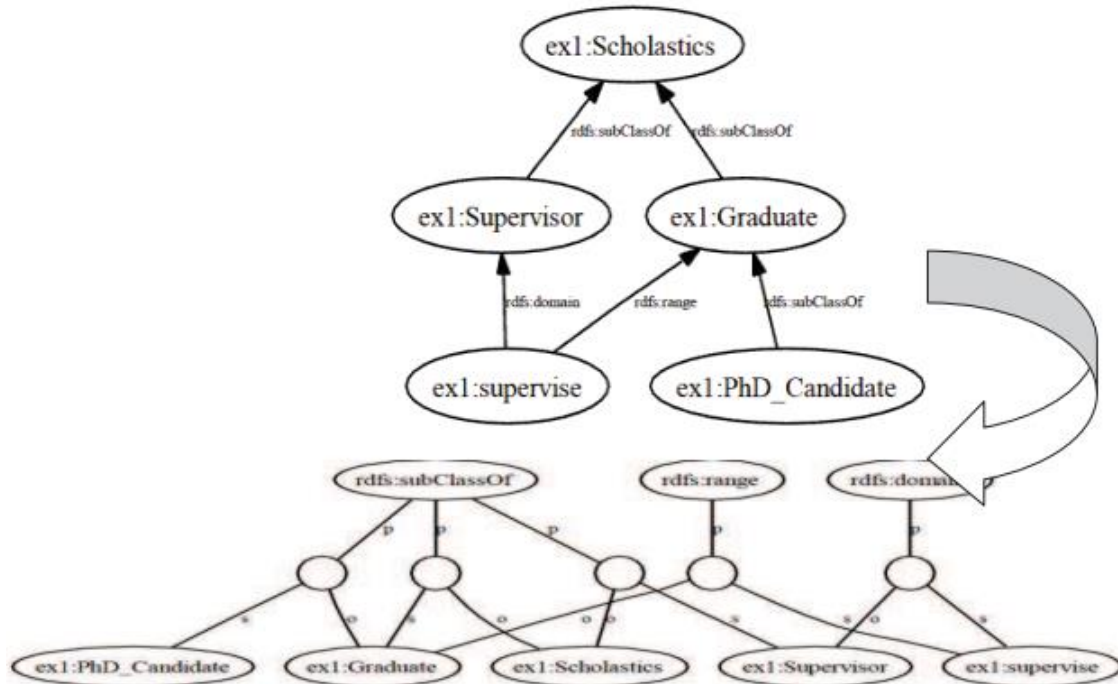
➤ **ASCO2** : L'algorithme ASCO2 propose un modèle de calcul de similarité sur deux étapes, la similarité partielle et la similarité finale. La similarité partielle entre deux entités des deux ontologies est déduite entre les composantes correspondantes aux entités en question. Ces composantes sont des pièces de connaissance contenues dans les définitions de l'entité en employant des primitives du langage OWL. Les valeurs de similarité partielle sont ensuite agrégées dans un schéma de pondération variable pour obtenir une meilleure valeur de similarité finale de ces deux entités. [13]

➤ **OLA** : OLA pour OWL-Lite Alignement est un algorithme pour aligner des ontologies représentées en OWL. Cet algorithme utilise les différentes méthodes de calcul de similarité pour trouver les correspondances entre les entités de deux ontologies en se basant sur leurs caractéristiques et leurs rapports avec d'autres entités et pour combiner ces valeurs de similarités calculées pour chaque paire d'entités il utilise la somme pondérée des valeurs de similarité de chaque caractéristique. [19]

OLA transforme les ontologies sous forme d'un graphe OL-graphs qui représente toutes les caractéristiques possibles des ontologies représentées en OWL-Lite. Les nœuds du graphe sont: classe (C), objet (O), relation (R), propriété (P), instance de propriété (A), type de donnée (D), valeur de type de donnée (V), les labels de restriction de propriété (L). Les arcs entre les nœuds sont : [18]

- rdfs: subClassOf entre deux classes ou deux propriétés (S);
- rdf: type (I) entre les objets et les classes, instance de propriété et propriété, valeurs et types de données ;
- A entre les classes et les propriétés, objets et les instance de propriétés ;
- owl: Restriction (R) exprimant la restriction sur une propriété dans une classe;
- valuation (U) d'une propriété d'un individu.

➤ **FALCON-OA** : Falcon est un outil d'alignement qui a été développé par Wei Hu et al afin de découvrir automatiquement les correspondances entre des ontologies en exploitant la structure et le langage de ces ontologies. Utilise les graphes RDF directs bipartis pour représenter les ontologies. Ces graphes sont dérivés des graphes RDF bipartis. La figure 6 montre le graphe RDF et le graphe RDF biparti d'une ontologie. [18]



**Figure II.6.** Le graphe RDF et le graphe RDF biparti d'une ontologie  $O_A$ . [18]

## II.8. Comparaison des méthodes d'alignement

La comparaison des méthodes d'alignement peut se faire sur plusieurs volets. Le premier volet permet de les comparer en se basant sur leurs caractéristiques (dimensions) externes. Ces caractéristiques regroupent principalement les entrées et les sorties des méthodes. Le second volet permet de comparer selon les caractéristiques internes.

### II.8.1. Comparaison selon les dimensions externes d'une méthode d'alignement

Les méthodes que nous avons choisies dans notre comparaison, sont souvent citées dans la littérature comme des méthodes d'alignement d'ontologie dans le domaine du web sémantique. La plupart de ces méthodes ont été conçues pour aligner des ontologies RDFS/OWL et elles produisent en sortie des alignements au format RDF/XML. La table I présente leur comparaison selon les dimensions externes.

Au niveau des relations détectées, nous remarquons que la plupart des méthodes comparées se limite à l'équivalence. En effet, seule la méthode TaxoMap détecte la

subsumption. Seule la méthode ASCO2 laisse un choix quant à la cardinalité des alignements qu'elle produit [18].

Méthode	Données d'entrées	Données sorties	
	Types de schémas	format	Relation
Falcon-AO [ 7 ]	RDFS / OWL	RDF/ XML	↔
Anchor-Prompt [21]	RDF(S), OWL	RDF(S), XML	↔
TaxoMap [7]	OWL/ RDF(taxonomies)	RDF/ XML	↔
OLA [7]	RDFS / OWL	RDF/ XML	↔
ASCO3 [21]	OWL DL/ Lite	RDF/ XML	↔
QOM [8]	RDFS/ OWL	RDF/XML	↔

**Tableau II.1.** Comparaison des méthodes par rapport à leurs dimensions externes [8].

### II.8.2. Comparaison selon les dimensions internes d'une méthode d'alignement :

Le tableau (II.8) montre, pour chaque méthode d'alignement:

- sa représentation interne : généralement les méthodes transforment les ontologies à aligner en graphes.
- sa technique utilisée : Ces méthodes sont des méthodes structurelles et terminologiques ;
- sa mesure de similarité : distance d'édition, TF/IDF, Jaro-Winkler...etc.;
- son pattern d'exploitation des ontologies: l'ontologie est un ensemble de triplet RDF comme ASCO2, où chaque entité est vue comme une seule structure de plusieurs composants comme QOM ;

➤ son paramétrage et sa composition : nous remarquons que toutes les méthodes présentées se composent de différentes techniques, et la plupart d'entre elles a une combinaison parallèle avec des moyennes pondérées. Quelques méthodes s'appuient également sur la fonction sigmoïde. La méthode TaxoMap utilise une combinaison séquentielle. Ces méthodes s'appuient, en générale, sur un seuil de similarité pour sélectionner les correspondances [18].

Techniques utilisées	Mesure de similarité	Pattern d'exploitation	Paramétrage et composition
Terminologique / structurelle	Edition distance, TF, IDF	RDF triples	Composition (intégration d'alignement), critère d'arrêt de l'algorithme itératif de propagation de similarité, seuil de similarité, la langue utilisée dans Is
Terminologique / structurelle	Techniques basés sur la mesure de similarité de Lin (SimLinLike)	Caractéristiques des entités (la richesse des étiquettes des entités)	Composition (séquentielle), Les catégories de mots considérées par Treetagger, les différents seuils de similarité (Equiv.threshold, HiddenInc.t hresholdSim,...), la langue utilisé
Terminologique / structurelle	Hamming entre les synsets, système d'équation-s	Caractéristiques des entités (descriptive knowledge about a couple of entités )	Composition (moyenne pondérée, parallèle), pondération (poids), seuil de similarité, un
Terminologique / structurelle	Jaro-Winkler (id, labels, synset), IF, I DF	RDF triples	Composition (la somme pondérée avec les poids variables, parallèle), pondération variable (poids), seuil de
Terminologique / structurelle	Distance d'édition	Caractéristiques des entités	composition (moyenne pondérée, fonction sigmoïde), critère d'arrêt de l'algorithme itératif de propagation de



Représentatif interne	Graphe bipartite direct	Format TaxoMap (seulement les étiquettes et les sous classes sont prends en	OL -Graphes	Format RDF	Format RDF(S)
Approches	Falcon -AO [7]	TaxoMap [7]	OLA [7]	ASCO2 [2]	QOOM [8]

**Tableau II.2.** Comparaison des méthodes par rapport à leurs dimensions internes [18].

## II.9. Conclusion

Dans ce chapitre, nous avons présenté un état de l'art sur l'alignement des ontologies; quelques domaines d'application de l'alignement des ontologies et les dimensions de L'alignement, puis nous avons présenté les techniques et les méthodes utilisées dans la littérature qui attaque le problème de recherche de la similarité, de la dissimilarité ou de la correspondance entre deux entités en général, qui apparaissent dans des ontologies. L'alignement d'ontologies revêt toute son importance dans des applications nécessitant la prise en compte d'une interopérabilité sémantique. Plusieurs approches d'alignement d'ontologies existent dans la littérature. Elles sont basées sur les mesures de similarités.

Les techniques d'alignement présentées sont les outils dont disposent les concepteurs de systèmes d'alignement pour trouver les solutions aux problèmes de l'hétérogénéité des ontologies.

Dans le chapitre suivant, nous présentons la conception de notre system, nous allons présenter une architecture générale pour le fonctionnement de notre application ainsi que la méthode utilisée pour calculer la distance entre les entités d'ontologies et en fin nous allons citer quelques étapes principales de notre algorithme d'application effectué et son déroulement pour mesurer les valeurs des similarités.

### III.1. Introduction

Le domaine biomédical est par ailleurs caractérisé par l'existence de nombreux standards terminologiques, thesaurus, et langages, partagés par les communautés biomédicales, qu'ils soient généralistes ou dédiés à un domaine de spécialité. Les ontologies doivent fournir les concepts et les relations utilisés pour le marquage sémantique des données en vue du Web Sémantique avec une signification partagée et réutilisable pour différentes applications et différents usagers.

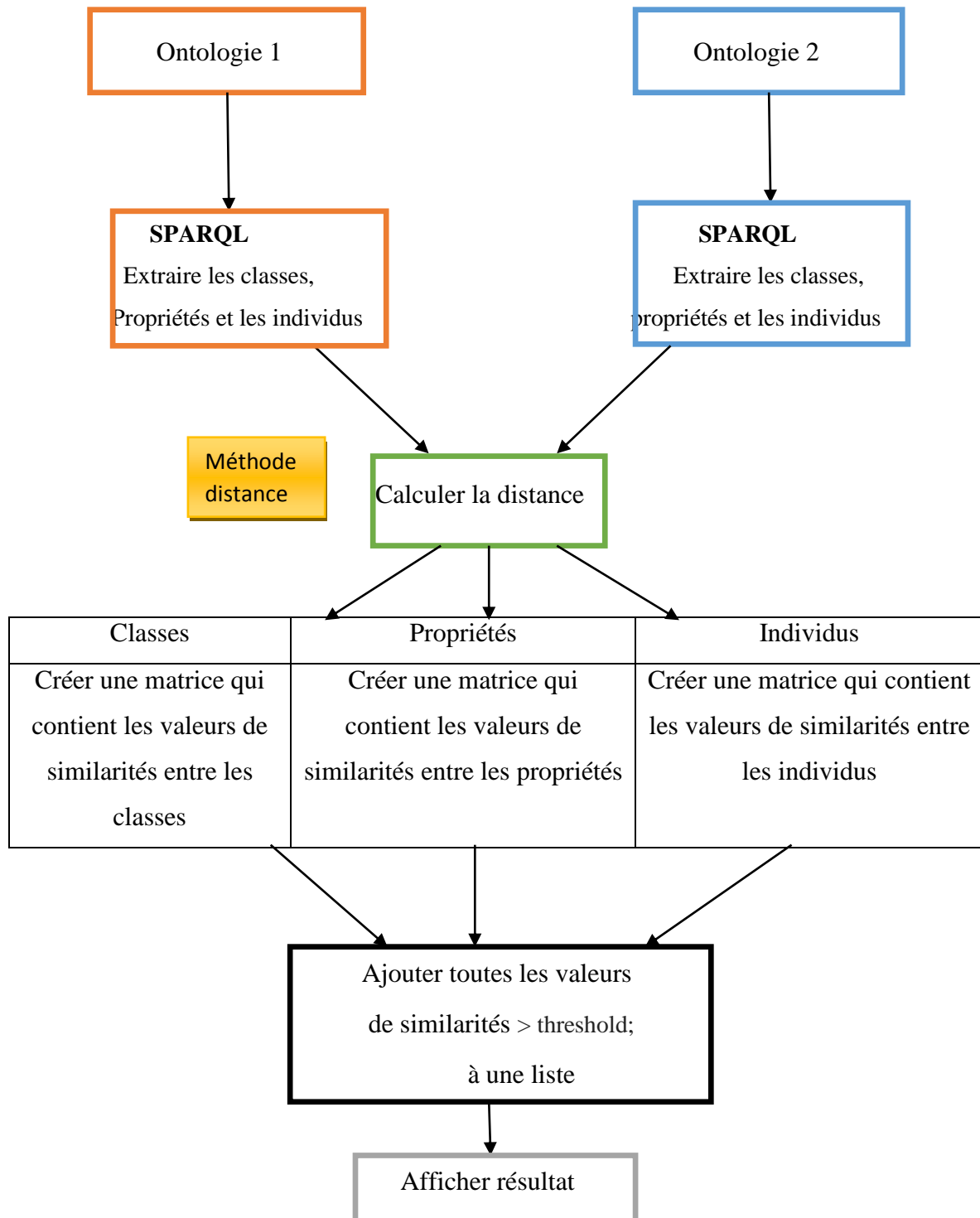
Dans ce chapitre, nous présentons notre approche d'alignement d'ontologies suivi, qui exploite les caractéristiques du langage OWL ; ce langage décrit et structure les deux ontologies existante dans le domaine biomédicale que nous avons choisi pour tester notre système ainsi que déduire les valeurs de similarités entre les entités de deux ontologies avec quelque d'exemples de calcul.

### III.2. Approche suivi :

Pour tester l'alignement on a utilisé deux ontologies existante l'une sur l'anatomie des humains et l'autre sur l'anatomie des rats.

L'approche s'articule sur trois grandes lignes :

- Proposer une représentation de l'ontologie décrite en OWL sous forme d'un graphe O-Graph. Ce graphe contient tous les entités et les relations d'ontologies.
- Définir une mesure de calcul de similarité entre les entités d'ontologies par catégorie d'entité (classe ; propriété ; individu).
- Génération d'un fichier nommé « Resultats.txt » qui contient les concepts similaires de deux ontologies OWL et leurs mesures de similarité correspondant.



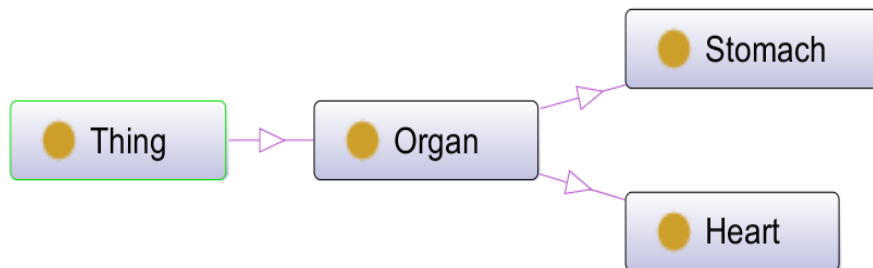
**Figure III.1.**architecture généraliste de fonctionnement l’application.

### III.2.1 Modèle du graphe O-Graph

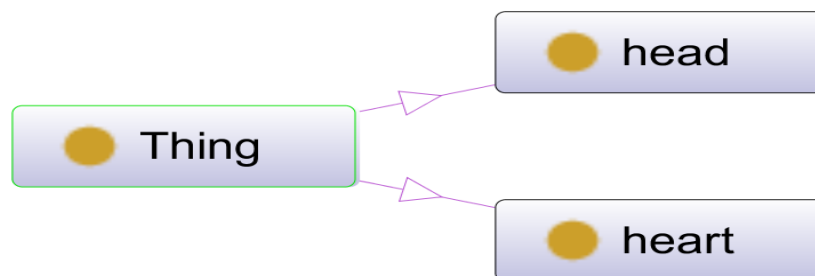
Il fallait tout d’abord penser à un modèle de graphe générique et adéquat pour représenter une ontologie OWL. Le rôle d’un tel graphe est de refléter toutes les entités, les relations et les individus de l’ontologie décrite sous le langage OWL.

L'ontologie humaine contient 10221 entités et l'ontologie rat contient 16682 entités.

Voici un extrait de deux ontologies humaine et rat.



**Figure III.2.** Représentation O-Grappe d'ontologie « humain ».



**Figure III.3.** Représentation O-Grappe d'ontologie « rat »

### III.2.2. Le problème de calcul des similarités

Le calcul de la similarité sémantique entre les concepts et entre les phrases est un problème de longue durée dans le domaine du traitement du langage naturel.

Il existe plusieurs moyens pour évaluer la similarité entre deux entités. La technique la plus répandue consiste à définir une formulation mathématique pour évaluer le degré de similarité. Le champ d'analyse sémantique a un rôle crucial à jouer dans la recherche liée à l'analytique du texte. La similarité sémantique diffère lorsque le domaine d'opération diffère.

Une méthode Pour calculer la similarité sémantique entre les mots et aussi entre les phrases peut être appliquée dans une variété de domaines. Chaque domaine nécessite des algorithmes différents et spécifiques bien que l'idée de base du calcul de la similarité sémantique reste la même. Pour déterminer la proximité entre les concepts en comparaison, nous avons besoin de mesure standard prédéfinie qui décrit aisément ces connexités de la signification. Les bases de données lexicales entrent dans l'image à ce stade de traitement. Les

bases de données lexicales ont des liens entre les mots qui peuvent être utilisés pour déterminer la similarité sémantique des mots.

Les valeurs de similarité calculées sont stockées dans une base de similarités (matrice). Elles sont calculées par des mesures normalisées, elles sont donc comprises dans l'intervalle de 0 à 1. Ces valeurs de similarité sont ensuite agrégées par différentes stratégies de combinaison pour atteindre une seule valeur de similarité finale entre deux entités, qui est aussi comprise entre 0 et 1. En examinant cette valeur de similarité finale par rapport à un seuil prédéfini [0.7], deux entités sont considérées comme similaires (équivalentes) ou différentes.

Tout ceci est développé dans la partie suivante. Dans la section suivante nous allons présenter la méthode d'alignement d'ontologies qui se base essentiellement sur les bases de données lexicales.

### **III.2.3. La méthode utilisée pour le calcul de la distance**

Le système que nous traitons appartient à un domaine médical (anatomie). La méthode d'alignement permet d'aligner deux ontologies décrites en format OWL. Elle produit un alignement représenté sous la forme d'un fichier texte. Ce fichier comporte les paires d'entités ontologiques appariées ainsi qu'une mesure de similarité.

En utilisant la méthodologie de base de données lexicales, la similarité est calculée en utilisant une hiérarchie prédéfinie, le sens et la relation avec d'autres mots qui sont stockés dans une structure arborescente. Lors de comparaison de deux mots, il prend en compte la distance de chemin entre les mots.

La méthode utilisée pour le calcul de la distance entre les concepts nommée : «distance» est une implémentation de la nouvelle version de la méthode «Levenshtein Distance». On appelle «distance de Levenshtein» entre deux mots M et P le coût minimal pour transformer M en P en effectuant les seules opérations élémentaires suivantes :

- substitution d'un caractère de M par un caractère différent de P ;
- insertion (ou ajout) dans M d'un caractère de P ;
- suppression (ou effacement) d'un caractère de M.

Cette méthode prend comme arguments deux phrases en langage naturel qui représentent les concepts de deux ontologies humaine et rat respectivement. Ensuite ; elle considère le texte comme une séquence de mots donc est une méthode terminologique qui traite de tous les mots

des phrases appartenant à base lexicale « Word Net » séparément selon leur structure sémantique et syntaxique.

Voici au dessous le principe de cette méthode:

1- La fonction **distance** prend comme des arguments deux chaînes de caractères.

- **Si** chaîne caractères 1 ou chaîne de caractères 2 n'existent pas **alors** return **1**.
- Calculer la longueur de deux chaînes de caractères 1 et 2 et réserve ses valeurs dans deux variables `len1` et `len2` respectivement.
- **Si** la longueur `len1` et `len2` égales a 0 **alors** return **0**.
- **Si** la longueur `len1` ou `len2` égales a 0 **alors** return **1**.
- Créer une variable nommé « best » qui représente le nombre des concepts similaires et initialiser à 0.
- Lancer le parcours des concepts lettre par lettre et il compare entre eux pour calculer la distance entre ces derniers par la formule suivante :

$$1 - (\text{best} * 2 / \text{len1} + \text{len2})$$

2- Return la valeur de la distance.

### III.3. Algorithme d'alignement proposé

Notre algorithme commence tout d'abord par la lecture de l'ontologie ensuite il fait l'extraction des classes/sous classes, propriétés, individus de chaque ontologie, il le stock dans une matrice en gardant la sémantique et le sens cela est fait pour chaque ontologie (human et mouse). L'algorithme procède à faire un calcul de distance entre les classes/ sous classes, propriétés et individus en vérifiant à chaque fois les synonymes de ces éléments avec wordnet.

La valeur obtenu est entre [0,1] en la finalise en mettant 1- valeur obtenu, ensuite la valeur va être comparé avec un seuil qui dans notre algorithme égale à 0.4 seules les éléments qui on une distance entre eux supérieure ou égale à 0.4 vont être pris en considération.

#### III.3.1 Les grandes lignes du déroulement de l'algorithme

- La lecture de deux ontologies « humaine » et « rat ».
- On utilisant des requêtes SPARQL pour extraire les classes et les propriétés et les individus pour chaque ontologie et retourne les résultats sous forme des listes:

Prop\_list1=[ ObsoleteProperty, Part of]

Prop\_list2=[ is a, ObsoleteProperty, Part of]

Class\_list1=[ Organ, Stomach , Heart ]

Class\_list2=[ head , heart]

- Calculer les distances entre ces concepts et mettre les résultats dans des matrices :

### III.3.2 Calculer les distances entre les classes et les sous classes :

- Calculer la distance entre « Organ » et « head » :

- Len1=5 len2=4

- Best=0 i=0

- Tant que  $i < 5$  et  $Len1-1 > 0$

J=0

- Tant que  $Len2-0 > 0$

K=0

- $0 < 4$  et  $organ[0] \neq head [0]$

J=1

$1 < 4$  et  $organ[0] \neq head [1]$

J=2

$2 < 4$  et  $organ[0] \neq head [2]$

J=3

$3 < 4$  et  $organ[0] \neq head [3]$

J=4

- $4 \neq 4$  non verifier

- Incrémentation  $i=1$

- Tant que  $i < 5$  et  $Len1-1 > 0$

J=0

- Tant que  $len2-0 > 0$

K=1

- $1 < 4$  et  $organ[1] \neq head [0]$

J=1

$1 < 4$  et  $organ[1] \neq head [1]$

J=2

2 < 4 et organ[1] != head [2]

J=3

3 < 4 et organ[1] != head [3]

J=4

- 4 != 4 non verifier

- Incrémentation i=2
- Tant que i < 5 et Len1-1 > 0

J=0

- Tant que Len2-0 > 0

K=2

- 0 < 4 et organ[2] != head [0]

J=1

1 < 4 et organ[2] != head [1]

J=2

2 < 4 et organ[2] != head [2]

J=3

3 < 4 et organ[2] != head [3]

J=4

- 4 != 4 non verifier

- Incrémentation i=3
- Tant que i < 5 et Len1-1 > 0

J=0

- Tant que Len2-0 > 0

K=3

- 0 < 4 et organ[3] != head [0]

J=1

- 1 < 4 et organ[3] != head [1]

J=2

- 2 != 4 verifier

J=3 k=4

Best = max(0, 4-3) = max(0, 1) = 1

- Tant que Len2-3 > 1 non verifier



- Incrémentation  $i=4$  rien ne change car il n'est pas des caractères similaires.
- Retour  $1 - (2*1)/(5+4)$

Donc: distance (Organ; head) = 7/9

### III.3.3. Calculer les distances entre les propriétés :

- **Calculer la distance entre « ObsoleteProperty » et « is a » :**

Best = 1      Retour  $1 - (2*1)/(16+3)$

Donc: distance (ObsoleteProperty ; is a) = 17/19

- **Calculer la distance entre « ObsoleteProperty » et « ObsoleteProperty » :**

Best = 16      Retour  $1 - (2*16)/(16+16)$

Donc: distance (ObsoleteProperty ; ObsoleteProperty) = 0

- **Calculer la distance entre « ObsoleteProperty » et « PartOf » :**

Best = 4      Retour  $1 - (2*4)/(16+6)$

Donc: distance (ObsoleteProperty ; part of) = 14 /22

- **Calculer la distance entre « PartOf » et « is a » :**

Best = 1      Retour  $1 - (2*1)/(6+3)$

Donc: distance (PartOf ; is a) = 7 /9

- **Calculer la distance entre « PartOf » et « ObsoleteProperty » :**

Best = 4      Retour  $1 - (2*4)/(16+6)$

Donc: distance (part of ; ObsoleteProperty) = 14 /22

- **Calculer la distance entre « PartOf » et « PartOf » :**

Best = 6      Retour  $1 - (2*6)/(6+6)$

Donc: distance (PartOf ; PartOf) = 0

**III.3.4. Création des matrices :**

- Initialiser les matrices de classes et les matrices des propriétés à 0.
- Ensuite nous avons rempli ces matrices en utilisant deux boucles « for » imbriquées.
  - Calculer la valeur de similarité : valeur = 1- distance entre les concepts.
  - Affecter la valeur obtenue à la matrice
  - Si la valeur de similarité supérieur à threshold (seuil proposé) alors ajouter ces concepts et la valeur de similarité à liste des objets.
- Afficher resultats.

**III.3.5 La matrice des classes « Class\_matrix »**

- **Initialiser la matrice :**

	Stomach	Heart	Organ
head	0	0	0
heart	0	0	0

Class\_matrix=[ [0,0,0 ] , [0,0,0 ] ]

Similarité (Organ,head) = 1- distance (Organ,head)  
 = 1- 7/9  
 = 2/9

Similarité (Organ,heart) = 1- distance (Organ,heart)  
 = 1- 6/10  
 = 4/10

Similarité (Stomach,head) = 1- distance (Stomach,head)  
 = 1- 7/11  
 = 4/11

Similarité (Stomach,heart) = 1- distance (Stomach,heart)  
 = 1- 6/12  
 = 6/12

Similarité (Heart,head) = 1- distance (Heart,head)  
 = 1- 3/9  
 = 6/9

Similarité (Heart,heart) = 1- distance (Heart,heart)  
 = 1- 0  
 = 1

**Finalement, on obtient la matrice suivante :**

	Stomach	Heart	Organ
head	<b>2/9</b>	<b>4/11</b>	<b>6/9</b>
heart	<b>4/10</b>	<b>6/12</b>	1

Class\_matrix = [ [ 2/9,4/11,6/9 ],[ 4/10 ,6/12,1] ]

### III.3.6. La matrice des propriétés « Prop\_matrix »

- Initialiser la matrice :

	ObsoleteProperty	PartOf
Is a	<b>0</b>	<b>0</b>
ObsoleteProperty	<b>0</b>	<b>0</b>
PartOf	<b>0</b>	<b>0</b>

Prop\_matrix=[ [0,0] , [0,0] , [0,0] ]

$$\begin{aligned} \text{Similarité (ObsoleteProperty, is a)} &= 1- \text{distance (ObsoleteProperty, is a)} \\ &= 1- 17/19 \\ &= 2/19 \end{aligned}$$

$$\begin{aligned} \text{Similarité (ObsoleteProperty, ObsoleteProperty)} &= 1- (\text{ObsoleteProperty, ObsoleteProperty}) \\ &= 1- 0 \\ &= 1 \end{aligned}$$

$$\begin{aligned} \text{Similarité (ObsoleteProperty,PartOf)} &= 1- \text{distance (ObsoleteProperty,PartOf)} \\ &= 1- 14/22 \\ &= 8/22 \end{aligned}$$

$$\begin{aligned} \text{Similarité (PartOf,is a)} &= 1- \text{distance (PartOf,is a)} \\ &= 1- 7/9 \\ &= 2/9 \end{aligned}$$

$$\begin{aligned} \text{Similarité (PartOf,ObsoleteProperty)} &= 1- \text{distance (PartOf,ObsoleteProperty)} \\ &= 1- 14/22 \\ &= 8/22 \end{aligned}$$

$$\begin{aligned} \text{Similarité (PartOf,PartOf)} &= 1- \text{distance (PartOf,PartOf)} \\ &= 1- 0 \end{aligned}$$

= 1

**Finalement, on obtient la matrice suivante :**

	ObsoleteProperty	PartOf
Is a	<b>2/19</b>	<b>2/9</b>
ObsoleteProperty	<b>1</b>	<b>8/22</b>
PartOf	<b>8/22</b>	<b>1</b>

Class\_matrix = [ [ 2/19,2/9 ],[ 1,8/22],[8/22,1]]

### III.4 Conclusion

Dans ce chapitre, nous avons vu les différentes étapes de l'algorithme proposée, qu'est utilisé pour aligner les ontologies représentées en OWL, en donnant comme résultats la liste des correspondances entre les entités des deux ontologies à aligner. Cet algorithme fait l'extraction de tous les concepts pour chaque ontologie et les stocks dans des listes ensuite fait un calcul de distance entre ces entités en vérifiant à chaque fois les synonymes de ces éléments avec wordnet. Pour le calcul la valeur de similarité en mettant (1- distance obtenu), le résultat va être comparé avec un seuil donné ; finalement seules les éléments qui on une valeur de similarité supérieure ou égale à ce seuil vont être affichés.

## IV.1 Introduction :

Dans ce chapitre, nous présentons l'implémentation et l'évaluation de l'algorithme d'alignement d'ontologies sur deux ontologies existante que nous avons présenté dans le chapitre précédent. L'implémentation de l'algorithme a été effectuée avec python sur l'environnement de développement Anaconda, ainsi qu'avec l'aide de la bibliothèque Wordnet.

## IV.2 Outils de développement :

Avant de commencer l'implémentation de notre système, nous allons tout d'abord spécifier les outils utilisés.

### ➤ Le vocabulaire Wordnet

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. 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.

### ➤ Langage Python

Nous avons utilisé comme langage de scriptage ; le langage « python », il est développé depuis 1989 par Guido van Rossum et de nombreux contributeurs bénévoles. Python peut être considéré comme un langage de haut niveau généraliste et peut entre autre être utilisé pour la création de scripts destinés à l'administration système ou à la manipulation de fichiers textuels. Le langage étant très complet et ne nécessitant aucune phase de compilation. Le choix de ce langage présente les avantages suivants :

- Python est entièrement gratuit.
- C'est un langage complet et puissant dans de nombreux domaines.
- Il est orienté objet mais n'impose pas ce type de programmation.
- Sa syntaxe reste très simple et le code peut être très lisible.
- Ses performances semblent être très honorables pour un langage dit de script.

- Raccourcit le cycle de développement par rapport aux langages compilés et permet un prototypage rapide des projets.

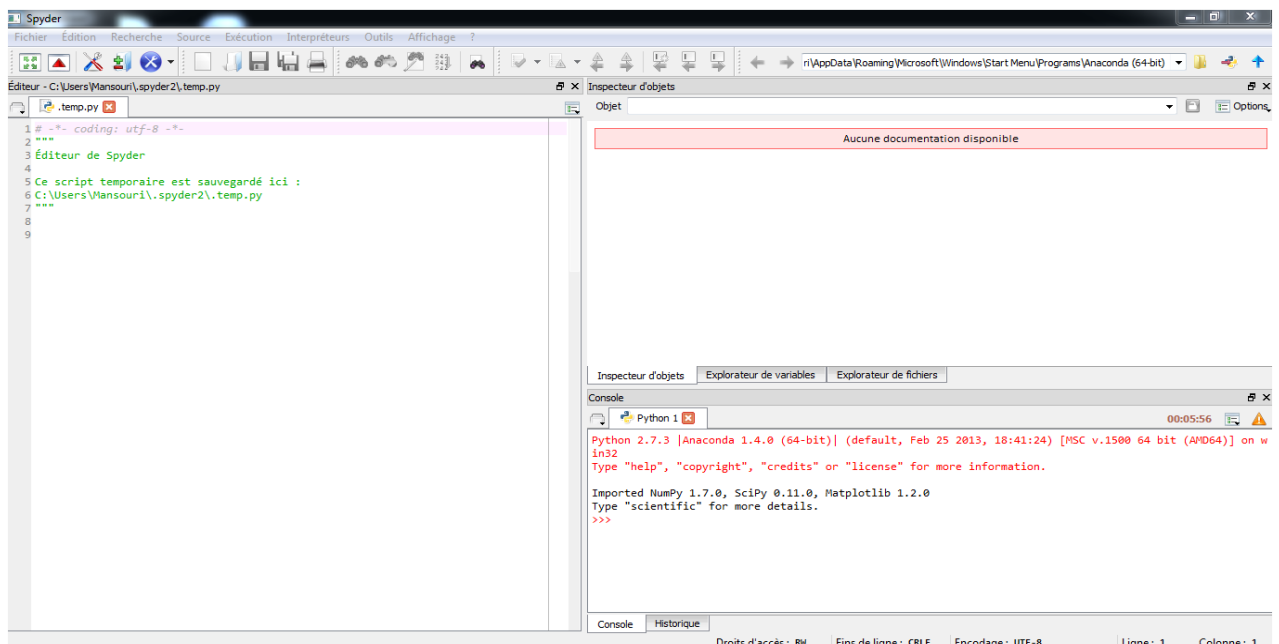
### ➤ Environnement Anaconda

Anaconda est un installateur pour les distributions GNU/Linux dérivées de Red Hat (Red Hat Linux, CentOS, Fedora, Oracle Linux et Scientific Linux) et certaines des distributions dérivées de Gentoo (Sabayon Linux et Vida Linux).

Ce logiciel est écrit en Python et en C. Son nom fait référence au serpent anaconda et est un clin d'œil au serpent python, le nom du langage de programmation utilisé (bien que le nom du langage Python ne vienne pas directement du nom du serpent).

Anaconda propose une installation au choix en mode texte ou en mode interface graphique. Il fonctionne sur de nombreuses plates-formes (x86, Itanium, DEC Alpha, IBM ESA/390, PowerPC). Il permet l'installation à partir de périphériques locaux (disque dur ou CD-ROM) ou du réseau (FTP, HTTP ou NFS).

À la fin de l'installation, un fichier de configuration *anaconda-ks.cfg* est généré.



**Figure IV.1.** Environnement de développement (Anaconda)

### ➤ **Spyder (nommé Pydee dans ses premières versions)**

Est un environnement de développement pour Python. Libre (Licence MIT) et multiplateforme (Windows, Mac OS, GNU/Linux), il intègre de nombreuses bibliothèques d'usage scientifique : Matplotlib, NumPy, SciPy et IPython.

En comparaison avec d'autres IDE pour le développement scientifique, Spyder a un ensemble unique de fonctionnalités - multiplateforme, open-source, écrit en Python et disponible sous une licence non-copyleft. Spyder comprend le support d'outils interactifs pour l'inspection des données et incorpore des instruments d'assurance de la qualité et d'introspection spécifiques au code Python, tels que Pyflakes, Pylint et Rope.

### ➤ **Protégé 4.3**

Protégé est un système auteur pour la création d'ontologies. Il a été créé à l'université Stanford et est très populaire dans le domaine du Web sémantique et au niveau de la recherche en informatique. Protégé est développé en Java. Il est gratuit et son code source est publié sous une licence libre (la *Mozilla Public License*). Protégé peut lire et sauvegarder des ontologies dans la plupart des formats d'ontologies : RDF, RDFS, OWL, OWL-S etc. Il possède plusieurs concurrents tels que Hozo, Onto Edit et Swoop. Il est reconnu pour sa capacité à travailler sur des ontologies de grandes dimensions.

## **IV.3 Les ontologies de tests**

Pour appliquer l'alignement on a utilisé deux ontologies existantes l'une sur l'anatomie des humains et l'autre sur l'anatomie des rats.

### **IV.3.1 L'ontologie « humain » et L'ontologie « rat »**

Pour tester l'algorithme implémenté, on a utilisé une paire d'ontologies qui soient très grandes, pour tester notre application. Ces ontologies se composent d'un nombre très énorme d'entités, nous avons réduit la taille de ses derniers pour rendre le temps d'exécution de l'application assez rapide.

***Pour les tests nous n'avons pas utilisé les ontologies proposées tout entières, on a pris une partie de quelques entités de chaque ontologie, vu que le temps d'exécution est très important.***

## IV.4 Implémentation de notre système

Cette partie décrit les détails d'implémentation de notre système. Nous commençons d'abord par la description des classes des tâches du système ensuite :

### IV.4.1. Description des classes de tâche du système

#### ➤ «Objet\_Alignement » :

C'est la classe qui permet l'initialisation des variables qu'on va afficher comme résultat ; concept 1 et concept 2 et la mesure de similarité.

#### ➤ « Alignement » :

C'est la classe qui permet d'initialiser les variables : onto1 et onto2 qui sont les deux NULL et la liste des objets.

Les méthodes principales de la classe « Alignement » :

- **Init** : une méthode qui fait l'appel à la méthode « ontologie » qui permet de lire l'ontologie et retourne sous forme de graphe
- **Propriétés (ontologie)** : cette méthode retourne une liste des propriétés par une requête SPARQL.
- **Classes (ontologie)** : cette méthode retourne une liste des classes par une requête SPARQL.
- **individuels (ontologie)** : cette méthode créer une liste et fait appel a requête SPARQL et ajouter ces individuels a la liste des individuels.

#### ➤ «methode\_alignement » :

C'est la classe qui prend en entrée la classe « Alignement ».

Les méthodes principales de cette classe :

- **calcule\_de\_distance (string1, string2 )** : cette méthode tout d'abord transforme les deux chaines de caractères en minuscule ensuite fait appel à la fonction qui calcule la distance « calculdistance » et fait un test si les chaines de caractères n'ont pas des synonymes dans Wordnet ; fragmente cette chaine en des fragments et teste si les fragments existent dans Wordnet et retourne la distance calculée.
- **mesure1(classe1,classe2)** : cette méthode calcule la distance entre deux classes, si les deux classes sont NULL retourne 1 sinon calculer la distance.
- **mesure(entité1,entité2)** : cette méthode calcule la distance entre deux propriétés et entre deux individus , si les deux entités sont NULL retourne 1 sinon calculer la distance entre eux.
- **Alignier()** : cette méthode permet de créer des matrices contenant les valeurs des distances entre les propriétés et les classes et les individus.



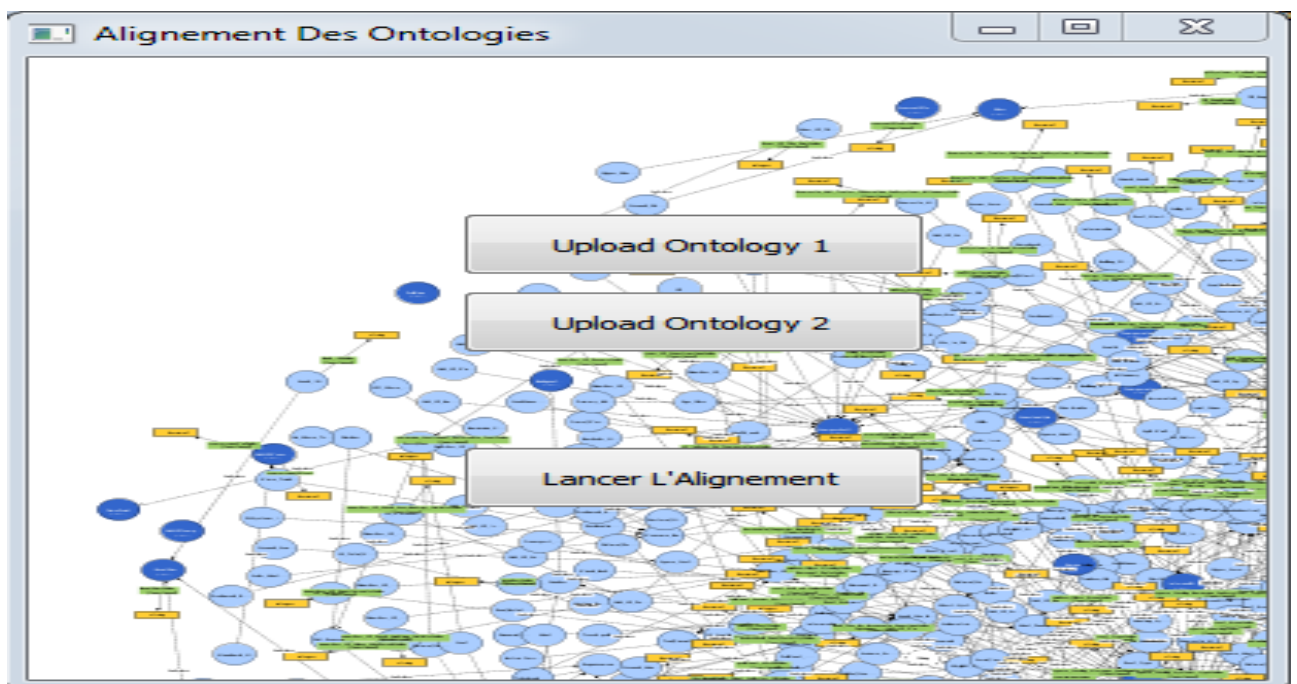
## IV.4.2 Présentation de notre système

### IV.4.2.1 Page d'accueil :

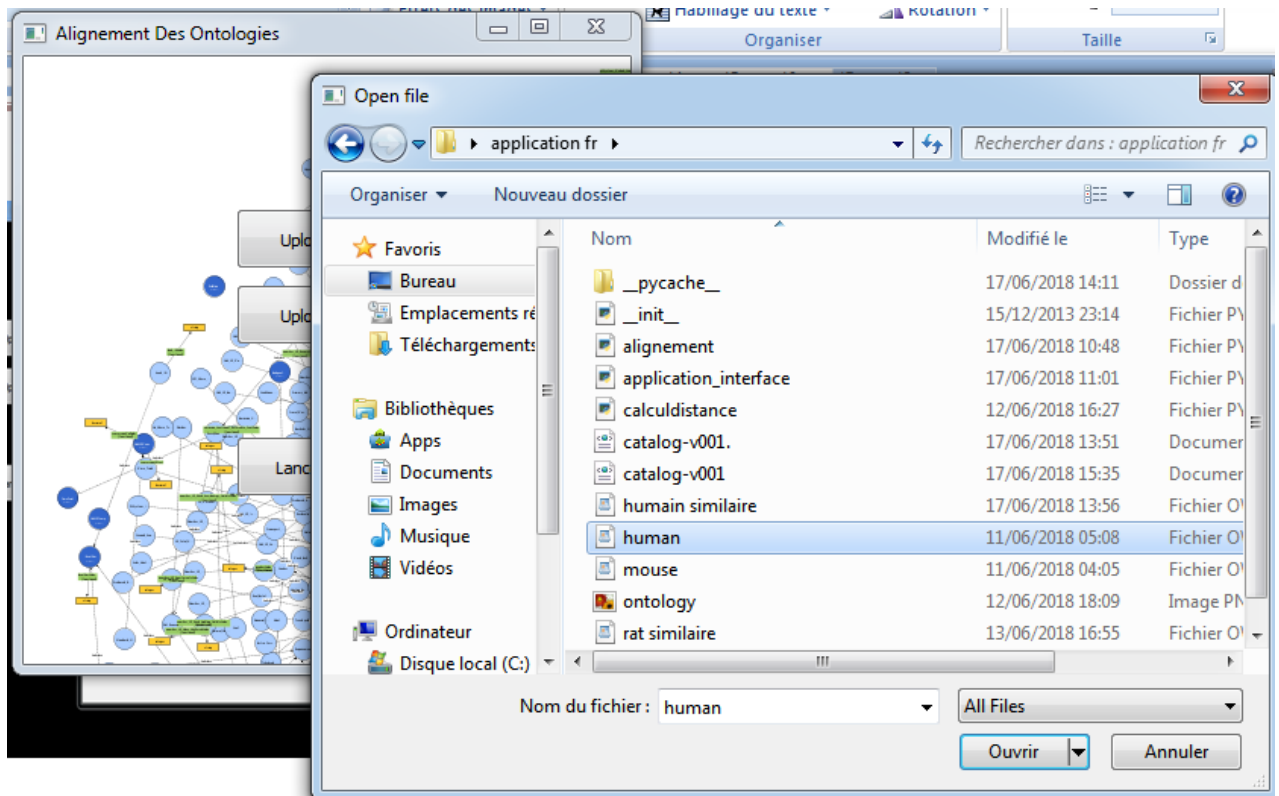
Maintenant on fera des tests sur l'application et nous allons les effectuer dans les prises d'écrans qui nous montrent les différentes étapes nécessaires à la réalisation de l'alignement d'ontologie et avoir les résultats dans un fichier qui permet à l'utilisateur de bien exploiter les similarités entre les concepts.

La figure suivante est l'interface d'accueil de notre application. Elle contient deux boutons «Upload Ontology 1» et «Upload Ontology 2» qui permettent à l'utilisateur de choisir les fichiers OWL des deux ontologies à aligner, et cela à travers un explorateur de fichiers (Figure IV.3).

Après qu'on charge les fichiers des ontologies à aligner, on appuie sur le bouton « Lancer L'Alignement », pour lancer l'algorithme.



*Figure IV.2. L'interface d'accueil de l'application "Alignement Ontologie"*



*Figure IV.3. La fenêtre de l'explorateur pour choisir le fichier d'ontologie*

Après que l'algorithme finisse son travail, un fichier nommé « Resultats.txt » sera créé, dans lequel le résultat des correspondances est annoncé.

#### IV.5 Résultats expérimentaux:

Le tableau suivant représente un exemple de la liste des correspondances trouvées par l'algorithme :

Les concepts de l'ontologie « humain »	Les concepts de l'ontologie « rat »	Mesure de similarité
Heart	head	0.666666667
Organ	heart	0.4
Stomach	heart	0.545454545
Heart	heart	1
ObsoleteProperty	ObsoleteProperty	1
PartOf	PartOf	1

**Tableau IV.1.** Résultats des correspondances obtenus par l'algorithme proposé

**VI.5.1. Les Concepts ayant une valeur de similarité inférieure au seuil proposé (0.4)**

Les concepts de l'ontologie « humain »	Les concepts de l'ontologie « rat »	Mesure de similarité
Organ	head	0.2222222222
Stomach	head	0.0.3636363636
Stomach	heart	0.545454545
Heart	heart	1
ObsoleteProperty	Is a	0.1052631579
PartOf	Is a	0.2222222222
PartOf	ObsoleteProperty	0.3636363636
ObsoleteProperty	PartOf	0.3636363636

**Tableau IV.2.** Résultats des correspondances inférieures à seuil proposé**IV.6 Conclusion**

Dans ce chapitre, nous avons vu les outils qu'on a utilisés pour l'implémentation de l'algorithme, qui sont Le vocabulaire Wordnet qui nous a permis de trouver tout les synonymes d'un mot ou d'un terme en anglais, Langage Python qu'on a utilisé comme un langage de scriptage (Scripting langage). Ensuite l'environnement Anaconda qu'on considère comme une plate forme de programmation ,Spyder qu'on a utilisé comme un éditeur de textes (IDE) et protégé pour l'édition d'ontologie. On a fait un petit aperçu sur notre interface. Pour les ontologies de teste on n'avait pas beaucoup de choix. On a utilisé deux ontologies existantes et énormes pour appliquer l'alignement.

### Conclusion Générale

Les recherches qui ont été menés et décrits dans ce mémoire se situent dans le cadre de l'alignement des ontologies. Nous avons commencé d'abord par l'explication de la notion d'ontologie, ses composants, comment on peut la développer et les outils importants pour faire ça. Ensuite nous avons défini le processus d'alignement, domaine d'application de l'alignement, les méthodes de base utilisées dans l'alignement, on a présenté aussi une comparaison entre les méthodes d'alignement et quelques approches de l'alignement d'ontologies. Enfin nous avons présenté la conception de notre application qui décrit la méthode proposé pour aligner nos ontologies et explique le fonctionnement de l'algorithme exploité en suivant une procédure très précis pour mesurer les similarités entre les entités des ontologies avec un scénario de déploiement dans la section d'implémentation.

Dans notre travail nous avons développé un outil informatique qui permet d'aligner deux ontologies. L'application proposée est axée notamment sur les deux fonctionnalités suivantes. La première consiste sur l'utilisation d'une base de données lexical « wordnet » pour découvrir si le concept concerné possède au moins un synonyme dans cette dernière.

La seconde fonctionnalité, qui représente notre travail de développement de cette application, est basée sur les chaînes de caractères en particulier la mesure de distance qui est une méthode amélioré de Levenstein distance de calcul des valeurs de distances entre les entités de deux ontologies choisi pour le test. A partir de ces valeurs de distances, notre algorithme calcule les mesures de similarités ainsi que les correspondances sémantiques sont identifiées.

### Les Perspectives :

Comme perspective, on doit essayer de trouver des solutions pour ces limites, afin d'améliorer cet algorithme. Surtout pour le problème du temps d'exécution de l'algorithme, on doit trouver une manière pour optimiser le temps. Ce qui va nous permettre d'exécuter cet algorithme sur des ontologies de tailles normales.

**BIBLIOGRAPHIE**

[1] : Brigitte Safar, Chantal Reynaud, François Calvier ; Techniques d'alignement d'ontologies basées sur la structure d'une ressource complémentaire ; Université Paris-Sud, CNRS (LRI, UMR 8623) & INRIA (Futurs),91405 Orsay, France ;

[2] : Bounaceur mohamed salah , Kahloul hafiane (2016) ; Mémoire Présenté en vue d'obtenir le diplôme de Master en informatique ;  
Elaboration D'un Environnement Virtuel Pour la Gestion Des Travaux Pratiques à distance ; mai2016.

[3] : Chaour Moundir , Kermani Mohamed Hachem ; Développement d'une ontologie floue et classification d'instances : Application au domaine des analyses médicales ; Projet de fin d'études pour l'obtention du diplôme de Master Recherche en Informatique ; Juin 2014

[4] : hadjui Fatima Zohra ; Mémoire de fin d'études pour l'obtention du diplôme de Master en Informatique ; Gestion des Ontologies Médicales ;  
Septembre 2012.

[5]: HACINE GHERBI Ahcine ; Mémoire Présenté pour l'obtention du Diplôme de Magister ; Construction d'une Ontologie pour le WEB Sémantique ;  
École doctorale d'informatique.

[6] : Aouachria Moufida ; Une approche basée ontologie pour la réutilisation des connaissances dans le processus d'affaires (Business Process) ;MÉMOIRE Présenté en vue de l'obtention du diplôme de Magistère ; 2012.

[7] : Abdeltif Elbyed ; « ROMIE, une approche d'alignement d'ontologies à base d'instances Institut National des Télécommunications », 2009.

[8]: Michel Tétréault ; Modélisation d'une ontologie et conceptualisation d'une application sémantique dédiée au e-recrutement dans le domaine des technologies de l'information ; Mémoire présenté à la Faculté des arts et des sciences de la communication en vue de l'obtention du grade de M.Sc ; Université de Montréal ; 2011.

- [9] : M.Hemam. Thèse pour l'obtention du diplôme de magister en informatique « Un processus de développement d'ontologies dans le cadre du Web sémantique ». Université Mentouri Constantine, Algérie 2005.
- [10] : Z. Boufaïda. «Cours Web sémantique ».Université Constantine 2 Algérie 2014.
- [11] : O.Djama. Thèse pour l'obtention du diplôme de magister en informatique. « Une approche d'annotation sémantique à partir d'une ontologie multi-points de vue ». Université Mentouri Constantine, Algérie 2010.
- [12] :SamiZghal,KarimKamoun,SadokBenYahia,EngelbertMephuNguifo,YahyaSlimani .« EDOLA:Unenouvelle méthode d'alignement d'ontologies OWL-Lite ». Saint-Etienne, 28–30 mars 2007
- [13] : [http://thesis.univ-biskra.dz/843/1/Inf\\_m1\\_2011](http://thesis.univ-biskra.dz/843/1/Inf_m1_2011), pdf.
- [14] : Hecht, Thomas and Buche, Patrice and DibieBarthélemy, Juliette and Ibanescu, Liliana and Trojahn, Cassia. « Open Archive TOULOUSE Archive Ouverte (OATAO) ». 28 January 2014 - 31 January 2014 (Rennes, France).
- [15] : Jean-Marc Mercantini, Nicole Tourigny, Eugène Chouraqui. « Élaboration d'ontologies à partir de corpus en utilisant la méthode d'ingénierie des connaissances KOD ». LSIS – UMR CNRS 6168, LSI-ERICA Département d'informatique et de génie logiciel.
- [16] : Sami Zghal,Sadok Ben Yahia, Engelbert Mephu Nguifo, Yahya Slimani. « S ODA : Une approche structurelle pourl'alignement d'ontologies OWL-DL ». CRIL CNRS FRE 2499, Université d'Artois, IUT de Lens, Département des Sciences de l'Informatique, Faculté de Sciences de Tunis.
- [17] : Sami Zghal, « Contributions à l'alignement d'ontologies OWL par agrégation de Similarités ».Thèse Doctorat en Informatique université d'ARTOIS 2010.
- [18] : Soumaya Kasri, « Etude des techniques d'alignement des ontologies: proposition d'un algorithme de passage à l'échelle ». Thèse Magister en Informatique université 20 Août 1955 Skikda.2010
- [19] : Yassine namir,Achraf ragui. « Alignement d'ontologies du web sémantique ».Master en Informatique université Sidi Mohamed Ben Abdellah 2016

- [20] : Warith-Eddine DJEDDI. « Alignement sémantique des ontologies de grande Taille ». Thèse Doctorat en Informatique Université Badji Mokhtar –Annaba 2013
- [21] : Zohra Bellahsene, « Rôle et techniques de l’alignement d’ontologies : un survol de l’état de l’art » ; INFORSID 2017.
- [22] : Jérôme Euzenat : Towards a principled approach to semantic interoperability in OIS@IJCAI; 2001
- [23] : Michel Klein Vrije : Combining and relating ontologies: an analysis of problems and solutions, Universiteit Amsterdam.
- [24]:[https://www.google.dz/search?rlz=1C1RLNS\\_frDZ686DZ694&ei=rvYqW5nXJIO7UeOqgMAO&q=ww;These+Thomas+cerque.com&oq=www.These+Thomas+cerque](https://www.google.dz/search?rlz=1C1RLNS_frDZ686DZ694&ei=rvYqW5nXJIO7UeOqgMAO&q=ww;These+Thomas+cerque.com&oq=www.These+Thomas+cerque)
- [25] :Moharned TOUZANI« Alignement des ontologies OWL-Lite », Faculté des arts et des sciences ,Avril,200.5
- [26] :MinaZiani ;Danielle Boulanger ;Guilaine Talens : « Système d’aide à l’alignement d’ontologies métier »,Marseille ,mai 2010.
- [27] : Aissa Fellah, Mimoun Malki, Ahmed ZAHAF, « Alignement des ontologies : utilisation de WordNet et une nouvelle mesure structurelle ».2008.
- [28]: Booch, 1994] Booch, G., “Object-Oriented Analysis and Design with Applications”, Addison-Wesley 2nd edition, 1994.
- [29] : Bouquet et al., 2003 Bouquet P., Giunchiglia F., Van Harmelen F., Serafini L. and Stuckenschmidt H, “C-OWL – contextualizing ontologies”.
- [30] : Bodenreider O<sup>1</sup>, Hayamizu TF, Ringwald M, De Coronado S, Zhang S : « Of mice and men: aligning mouse and human anatomies »,2005.
- [31]: « An Overview of the KL-ONE Knowledge Representation System »,27 juin 2014. <https://doi.org/10.1016/B978-0-934613-53-8.50019-4>
- [32] : Bach, T.L.: Construction d’un Web Sémantique Multi-Points de Vue. These de doctorat Informatique, Ecole des Mines de Paris a Sophia Antipolis, (2006).

[33] : J.Euzenat « Dynamique et sémantique: l'alignement d'ontologies dans le web sémantique est bien plus difficile que vous n'osiez même le penser » INRIA Grenoble, France 19 octobre 2008.