REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA
Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie
Département d'informatique

N° d'ordre : IA8/M2/2019

Mémoire

présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : **Intelligence Artificielle**

Alignement d'ontologies de domaine

Par :

RAMDANE AHMED MEROUANE

Soutenu le 07 juillet 2019, devant le jury composé de :

BEN SGHIR Nadia	MAB	Président
MEKLID Abdessalam	MAA	Rapporteur
MOUSSAOUI Manel	MAA	Examinateur

Dédicaces

Tous les mots ne sauraient exprimer la gratitude, l'amour, le respect, la reconnaissance, c'est tous simplement que : Je dédie ce projet à :

A Ma tendre Mère Samia : Tu représentes pour moi la source de tendresse et l'exemple de dévouement qui n'a pas cessé de m'encourager. Tu as fait plus qu'une mère puisse faire pour que ses enfants suivent le bon chemin dans leur vie et leurs études.

A Mon très cher Père Sami : Aucune dédicace ne saurait exprimer l'amour, l'estime, le dévouement et le respect que j'ai toujours pour vous. Rien au monde ne vaut les efforts fournis jour et nuit pour mon éducation et mon bien être. Ce travail et le fruit de tes sacrifices que tu as consentis pour mon éducation et ma formation le long de ces années.

A mes sœurs : Dania, Farah et son mari et leur petite princesse Djinane.

A mes cousin : Ali , Takieddine, Aymen, Okba, Sidali.

A toute ma très chère famille.

A mes très chers amis : BOUHALI Chihab-eddine, , ABDELHAK Djalal-eddine, YOUKANA Hamza, BEY Haroun , SAHNOUN Younes ,Hakki Salah-eddine , AOUBID Abderahman, MHENNAOUI Fouad , BOUZID Med Amine, BELLIL Taha ,AMRATE AYMEN, HAFIDHI Med Salah, ABDEDDAIM Ramdhan, BOUBAKEUR Houssem, BRAINIS Farid .

A monsieur l'encadreur MEKLID Abdessalam : qui ne cesse pas de m'encourager et me conseiller. Cette humble dédicace ne saurait exprimer mon grand respect et ma profonde estime.

A tous mes enseignants depuis mes premières années d'études.

A tous les membres de ma promotion.

A tous ceux qui ont contribué de près ou de loin.

A tous ceux qui me sens chers et que j'ai omis de citer.

Remerciements

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.

Je remercie mes très chers parents, Sami et Samia, qui ont toujours été là pour moi. Je remercie mes sœurs Farah et Dania, pour leurs encouragements.

Je tiens à exprimer toute ma reconnaissance à mon encadreur de mémoire, Monsieur MEKLID Abdessalam. Je le remercie de m'avoir encadré, orienté, aidé et conseillé.

J'adresse mes sincères remerciements à tous les professeurs, intervenants et toutes les personnes qui par leurs paroles, leurs écrits, leurs conseils et leurs critiques ont guidé mes réflexions et ont accepté de me rencontrer et de répondre à mes questions durant mes recherches.

Enfin, je remercie mes cousins Ali et Takieddine, mes amis Chihab-eddine, Djalal-eddine, Hamza et Haroun qui ont toujours été là pour moi. Leur soutien inconditionnel et leurs encouragements ont été d'une grande aide.

À tous ces intervenants, je présente mes remerciements, mon respect et ma gratitude.

Liste des tableaux

Liste des figures

Liste des abréviations

Abréviation	Mot entier
Rdf	Ressource Description Framework.
Rdfs	Ressource Description Framework Schéma (schéma léger).
Onto-Edit	l'éditeur d'ontologie.
IA	Intelligence Artificielle.
OWL	Ontology Web Language.
XML	eXtensible Markup Language.
O-Graphe	Ontology Graphe.
DS	Distance.
E-commerce	Electronic commerce.
SMA	Système Multi-Agents.
QOM	Quick Ontology Matching.
NOM	Naive Ontology Matching.
SODA	Structural Ontology OWL -DL Alignement.

RESUME. *L'alignement d'ontologies est une tâche complexe reposant sur la définition de mesures de similarité. De nombreuses étude sont été effectuées. Un seul type de mesure se révélant la plupart du temps insuffisant pour détecter une similarité, différents outils combinant plusieurs mesures de façon séquentielle ont été proposés: cette combinaison est effectuée a priori et n'est pas modifiable. Notre objectif est de concevoir un système d'alignement d'ontologies de domaine afin d'arriver au but d'aligner les ontologies du même domaine pour permettre aux autres systèmes de communiquer entre eux en utilisant des termes partagés.*

L'alignement entre deux ontologies (appariement ou mise en correspondance) consiste en la production d'un ensemble de correspondances entre les entités. Ces entités peuvent être des concepts, des propriétés ou encore des instances. Cependant la génération automatique des correspondances entre deux ontologies est d'une extrême difficulté qui est due aux divergences (conceptuelle, habitudes, etc.) entre communautés différentes de développement des ontologies. De plus, la problématique d'alignement se pose avec acuité lorsque le nombre et le volume des schémas de données sont importants. En effet, dans les domaines d'applications réelles où les ontologies sont volumineuses et complexes, les exigences de l'exécution du temps et de l'espace mémoire sont les deux facteurs significatifs qui influencent directement la performance d'un algorithme d'alignement.

ABSTRACT. *Ontology alignment is a complex task based on the definition of similarity measures. Many studies have been done. Since only one type of measurement is often insufficient to detect a similarity, different tools that combine several measurements in a sequential way have been proposed: this combination is performed a priori and can not be modified. Our goal is to design a domain ontology alignment system to achieve the goal of aligning ontologies of the same domain to allow other systems to communicate with each other using shared terms.*

The alignment between two ontologies (matching) consists of producing a set of correspondences between the entities. These entities can be concepts, properties or even instances. However, the automatic generation of correspondences between two ontologies is of an extreme difficulty which is due to divergences (conceptual, habits, etc.) between different communities of development of ontologies. In addition, the problem of alignment is acute when the number and volume of data patterns are important. Indeed, in real application domains where ontologies are large and complex, the requirements of the

execution of time and memory space are the two significant factors that directly influence the performance of an alignment algorithm.

Mots clés : *alignement d'ontologies, ontology mapping, hétérogénéité, médiation de context, web sémantique.*

Sommaire

INTRODUCTION GENERALE

Nous assistons ces dernières années à l'émergence de nouvelles applications ayant besoin de partager de l'information entre différents systèmes comme c'est le cas pour l'apprentissage enrichi par les technologies (Technology Enhanced Learning, TEL) et les applications biomédicales. L'enjeu est de développer des techniques facilitant l'interopérabilité sémantique entre ces systèmes d'informations, qui constituent généralement des sources de données autonomes et hétérogènes.

L'interopérabilité est une question importante, largement identifiée dans plusieurs domaines comme par exemple dans la communauté des systèmes d'information (SI). La dépendance et le partage d'information entre des organismes ont créé un besoin de coopération et de coordination qui facilite aussi bien l'échange et l'accès aux informations distantes qu'aux informations locales. La large adoption de l'internet (WWW : World Wide Web), pour accéder et distribuer l'information, engendre un besoin crucial de l'interopérabilité des systèmes.

Actuellement le Web contient plus de 4.2 milliards de pages, mais la grande majorité d'entre elles sont dans un format lisible et compréhensible uniquement pour l'homme. Par conséquent les machines ou plutôt les logiciels ne peuvent ni comprendre ni traiter cette information. De plus, une grande partie du Web reste jusqu'à présent inexploitée.

Pour pallier cette insuffisance du Web, les chercheurs ont créé la vision du Web sémantique, où les ontologies vont décrire la structure et la sémantique des données. L'idée est que les ontologies permettent à des utilisateurs d'organiser l'information en taxonomie des concepts, chacune avec leurs attributs, et décrivent des relations entre ces concepts. Quand des données sont présentées ou annotées par des ontologies, les logiciels peuvent mieux comprendre leurs sémantiques, ce qui facilite la localisation et l'intégration des données pour des objectifs divers.

Le terme d'ontologie dans la philosophie est la science de ce qui est, c'est-à-dire les natures et les structures d'objets, de propriétés, d'événements, de processus et de relations suivant le contexte palpable de la représentation. L'ontologie cherche une classification approfondie, dans le sens que tous les types d'entités sont inclus dans cette classification. Dans le domaine de l'informatique, une position plus pragmatique aux ontologies est

adoptée, où l'ontologie est considérée comme un accord sur une représentation de domaine.

L'ontologie est un facteur clé qui facilite l'interopérabilité dans le web sémantique. Les ontologies sont le noyau du Web sémantique parce qu'elles permettent aux applications de communiquer en utilisant des termes partagés. L'ontologie facilite donc la communication en fournissant des notions précises qui peuvent être employées pour composer et échanger des messages (questions, réponses, etc.). 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 est souvent réalisée par l'intégration manuelle ou semi-automatisée des ontologies. Elle consiste à identifier les liens de correspondance entre les ontologies, on parle alors de mapping d'ontologies.

Objectif de notre travail

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 travail est dans le cadre de l'alignement d'ontologies et consiste à développer un système d'alignement d'ontologies de domaine. Notre système dépend de la combinaison de plusieurs mésures de similarité comme des similarités partielles pour déduire la similarité globale et trouver les correspondances entre les entités des ontologies alignées. La combinaison des mésures de similarité est composée de trois types de similarité, la similarité terminologique, la similarité sémantique et la similarité structurelle. La première est basée sur la similarité des chaines de caractères calculée par une implémentation de la méthode Jaro Winkler. La deuxième est basée sur l'utilisation d'une ressource externe « WordNet » pour calculer la similarité sémantique par une implémentation de la méthode Wu&Palmer. La dernière est basée sur la structure des deux ontologies alignées.

Plan général du mémoire

Ce mémoire est composé de cinq chapitres :

Chapitre I : ce chapitre présente une introduction sur les ontologies, en donnant les différentes définitions, le rôle des ontologies, les différents composants, les types, cycle de vie, les langages de description et de représentations et aussi les domaines d'application.

Chapitre II : ce chapitre décrit l'alignement des ontologies, la naissance du besoin à l'alignement d'ontologies, les différentes méthodes d'alignement et quelques domaines d'application de l'alignement des ontologies.

Chapitre III : ce chapitre contient les approches existantes dans le domaine de l'alignement d'ontologies et la synthèse comparative entre ces différentes approches.

Chapitre IV : ce chapitre contient la conception de notre travail, nous allons présenter notre approche avec une architecture générale pour le fonctionnement de notre système, le problème de calcule des similarités , une architecture détaillée de l'algorithme d'alignement proposé et les méthodes principale utilisées dans ce système.

Chapitre V : ce chapitre contient les outils de développement utilisés pour la réalisation de notre application , les ontologies de test, l'implémentation de notre système, l'évaluation de ce système et l'analyse des résultats expérimentaux.

Chapitre I : Les ontologies

1. Introduction

L'exploitation de connaissances en informatique a pour objectif de ne plus faire manipuler en aveugle des informations à la machine mais de permettre un dialogue (une coopération) entre le système et les utilisateurs. Alors, le système doit avoir accès non seulement aux termes utilisés par l'être humain mais aussi à la sémantique qui leur est associée, afin qu'une communication efficace soit possible. Actuellement, la connaissance visée par ces ontologies est un sujet de recherche populaire dans diverses communautés (l'ingénierie des connaissances, la recherche d'information, le traitement du langage naturel, les systèmes d'information coopératifs, l'intégration intelligente d'information et la gestion des connaissances). Elles offrent une connaissance partagée sur un domaine qui peut être échangée entre des personnes et des systèmes hétérogènes. Elles ont été définies en intelligence artificielle afin de faciliter le partage des connaissances et leur réutilisation. La définition explicite du concept ontologie soulève un questionnement qui est tout à la fois d'ordre philosophique, épistémiologique, cognitif et technique.

2. Définition

Ontologie est 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. En effet, ce terme est construit à partir des racines grecques « ontos » 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.

Dans la philosophie classique, l'ontologie correspond à ce qu'Aristote appelait la Philosophie première (protè philosopha), c'est-à-dire la science de l'être en tant qu'être, par opposition aux philosophies secondes qui s'intéressaient, elles, à l'étude des manifestations de l'être (les existants).

Ontologie : partie de la métaphysique qui s'attache à l'étude ou à la théorie de l'être dans son essence, indépendamment des phénomènes de son existence.

- Dans le cadre de l'intelligence artificielle, Neeches et ses collègues furent les premiers à proposer une définition à savoir : « Une ontologie définit les termes et les relations de base du vocabulaire d'un domaine ainsi que les règles qui indiquent

comment combiner les termes et les relations de façon à pouvoir étendre le vocabulaire »

- En 1993, Gruber propose la définition suivante : « Spécification explicite d'une conceptualisation ».
- Cette définition a été modifiée légèrement par Borst comme « spécification formelle d'une conceptualisation partagée ».
- Ces deux dernières définitions sont regroupées dans celle de Studer comme «spécification formelle et explicite d'une conceptualisation partagée ».
  - Formelle : l'ontologie doit être lisible par une machine, ce qui exclut le langage naturel.
  - Explicite : la définition explicite des concepts utilisés et des contraintes de leurs utilisations.
  - Conceptualisation : le modèle abstrait d'un phénomène du monde réel par identification des concepts clefs de ce phénomène.
  - Partagée : l'ontologie n'est pas la propriété d'un individu, mais elle représente un consensus accepté par une communauté d'utilisateurs.
- Pour Guarino & Giaretta « une ontologie est une spécification rendant partiellement compte d'une conceptualisation ».
- Swartout et ses collègues la définissent comme suit: « une ontologie est un ensemble de termes structurés de façon hiérarchique, conçue afin de décrire un domaine et qui peut servir de charpente à une base de connaissances».
- La même notion est également développée par Gomez comme : « une ontologie fournit les moyens de décrire de façon explicite la conceptualisation des connaissances représentées dans une base de connaissances ». [1]

3. Rôle des ontologies

Nées des besoins de représentation des connaissances, les ontologies sont a l'heure actuelle au coeur des travaux mènes dans le Web sémantique. Visant a 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 des 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. Emergeant 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 base 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 engage est de rendre les machines suffisamment sophistiquées pour qu'elles puissent intégrer le sens des informations, qu'a 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 [5].

4. 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 **: [5]**

4.1. 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.

4.2. 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.

4.3. Fontions

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).

4.4. 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, …etc.

4.5. Instances

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

5. Les différents types d'ontologies

On distingue six types d'ontologie : [1]

5.1. Ontologie de représentation de connaissances

Modélise les représentations primitives utilisées pour la formalisation des connaissances sous un paradigme donné. Par exemple, une ontologie sur le formalisme des Topic Maps comportera les concepts : Topic, Type de Topic, Association, Occurrence, Type Occurrence,..

5.2. Ontologie de haut niveau / supérieure (Top-level / Upper-model)

Elle exprime des conceptualisations valables dans différents domaines. Elle décrit des concepts très généraux comme l'espace, le temps, la matière, les objets, les évènements, les actions, etc. ces concepts ne dépendent pas d'un problème ou d'un domaine particulier, et doivent être, du moins en théorie, consensuels à de grandes communautés d'utilisateurs. Ce type d'ontologies est fondé sur la théorie de la dépendance. 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 évènements, les états, les processus, les actions, le temps, l'espace, les relations, les propriétés…

5.3. Ontologie Générique

Elle est appelée également noyau ontologique, modélise des connaissances moins abstraites que celles véhiculées par l'ontologie de haut niveau mais assez générales néanmoins pour être réutilisées à travers différents domaines. Cette ontologie inclut un vocabulaire relatif aux choses, évènements, temps, espace, causalité, comportement, fonction, etc.

5.4. Ontologie du domaine

Cette ontologie exprime des conceptualisations spécifiques à un domaine, elle est pour plusieurs applications de ce domaine. Elle fournit les concepts et les relations permettant de couvrir les vocabulaires, activités et théories de ces domaines. Selon

Mzoguchi, l'ontologie du domaine caractérise la connaissance du domaine *où* la tache est réalisée. Par exemple, dans le contexte du e-Learning, le domaine peut être celui de formation.

5.5. Ontologie de Taches

L'ontologie de taches fournit un vocabulaire systématisé des termes employés pour résoudre des problèmes liés aux taches qui peuvent être ou non du même domaine. Elle fournit un ensemble de termes au moyen desquelles nous pouvons décrire généralement comment résoudre un type de problèmes. Elle inclut des noms, des verbes et des adjectifs génériques dans les descriptions de taches.

5.6. Ontologie d'application

C'est l'ontologie la plus spécifique, elle contient des concepts dépendants d'un domaine et d'une tache particuliers, elle est spécifique et non réutilisable. Ces concepts correspondent souvent aux rôles joués par les entités du domaine lors de l'exécution d'une certaine activité. Il s'agit donc ici de mettre en relation les concepts liés à une tache particulière de manière à en décrire l'exécution.

6. Cycle de vie

Puisque les ontologies sont destinées à être utilisées comme des composants logiciels dans des systèmes répondant à des objectifs opérationnels différents, leur développement doit s'appuyer sur les mêmes principes que ceux appliqués en génie logiciel. Ainsi, les ontologies doivent être considérées comme des objets techniques évolutifs et possédants un cycle de vie qui nécessite d'être précisé. Dans ce contexte, les activités liées aux ontologies sont, d'une part, des activités de gestion de projet (planification, contrôle, assurance qualité), et d'autre part, des activités de développement (spécification, conceptualisation, formalisation) ; s'y ajoutent des activités transversales de support telles que l'évaluation, la documentation, la gestion de la configuration.

Nous avons adapté ce cycle à nos besoins et proposons notre vision du cycle de vie d'une ontologie (Figure I.1). Il comprend une étape initiale de détection et de spécification des besoins qui permet notamment de circonscrire précisément le domaine de connaissances, une étape de conception qui se subdivise en trois phases, une étape de déploiement et de diffusion, une étape d'utilisation, une étape incontournable, d'évaluation, et enfin, une sixième étape consacrée à l'évolution et à la maintenance du

modèle. Après chaque utilisation significative, l'ontologie et les besoins doivent être réévalués et 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.

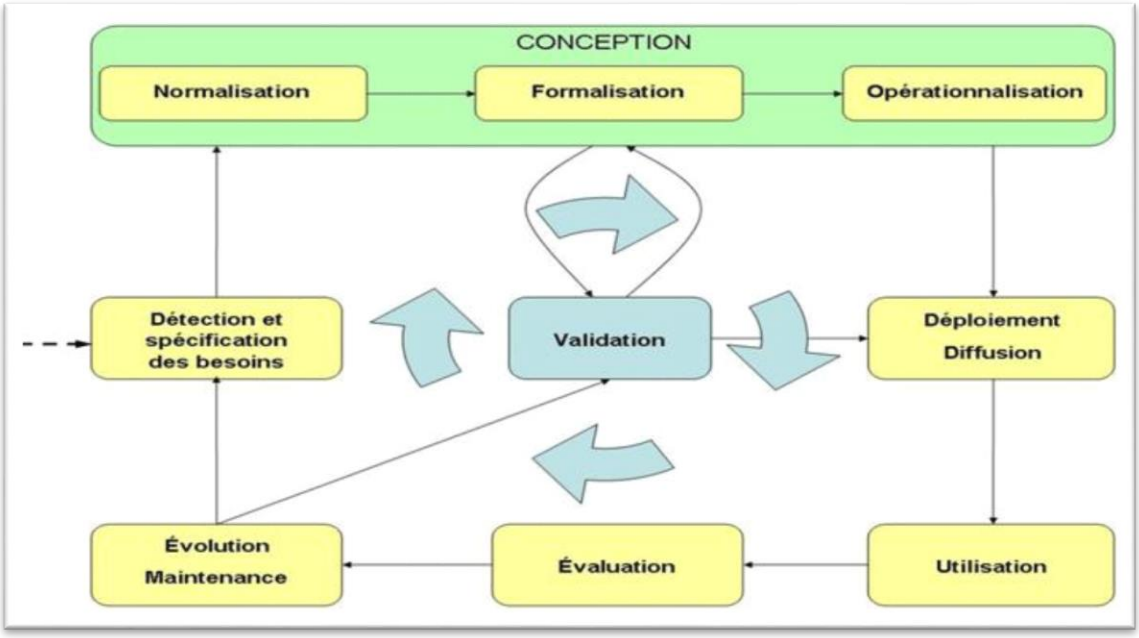

Figure I.1. Cycle de vie d'une ontologie. [1]

Les activités de documentation et d'évaluation sont nécessaires à chaque étape du processus de construction, l'évaluation précoce permettant de limiter la propagation d'erreurs. [1]

7. Langages de description et de représentation d'ontologies

Dans cette section, nous définissons quelques langages de représentation des ontologies les plus connus et les plus utilisés.

7.1. KIF

KIF est un langage basé sur les prédicats du premier ordre avec des extensions pour représenter des définitions et des méta-connaissances, la logique du premier ordre étant un langage de bas niveau pour l'expression d'ontologies. Une extension du langage KIF, ONTOLINGUA, est utilisée dans le serveur d'édition d'ontologies, Ontolingua du même nom. [2]

7.2. RDF et RDF Schéma

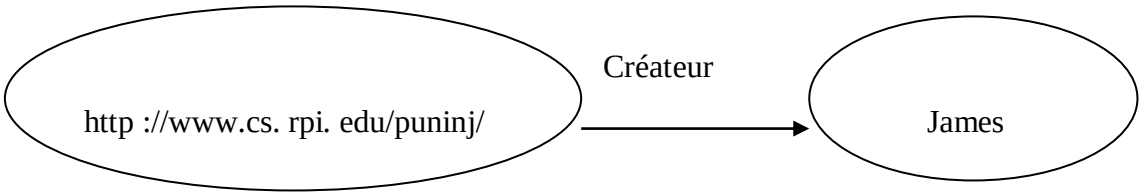

Figure I.2. Description d'une ressource selon le modèle RDF. [4]

Le RDF (Ressources Description FrameWork) est un formalisme standard développé par le W3C pour décrire les ressources sur le Web. Le RDF ne représente pas un langage mais un méta-modèle pour décrire les ressources sur le Web en utilisant la syntaxe d'XML [9]. Il est équivalent au formalisme des réseaux sémantiques. Ainsi, et suivant ce modèle, chaque ressource sur le Web peut être décrite sous forme d'un triplet composé d'une ressource, d'une propriété et d'une valeur. [4]

1. Ressource : décrite par des expressions de RDF et nommée souvent par des URIs. Elle peut être une page Web ou une partie d'une page définie par une balise XML (la page Web identifiée par I'URI « http ://www.cs. rpi. edu/puninj/ » dans la FigureI.2).
2. Propriété : représente un aspect particulier, une caractéristique, un attribut ou une relation qui décrit la ressource (la propriété Créateur dans la FigureI.2).
3. Valeur : peut être à son tour une ressource, assignant une valeur à une propriété d'une ressource spécifique (la valeur James dans la FigureI.2).

7.3. DAML + OIL

DAML+OIL est le fruit d'un projet commun de la fusion de deux langages l'OIL (Ontology Interchange Language) développé par l'union européenne et DAML (DARPA Agent Modeling Languaage) développé par les États Unis dans le cadre du projet DARPA (The Defense Advanced Research Projects Agency). DAML+OIL est construit sur le modèle RDFs et basé sur le modèle théorique des logiques de descriptions. Il est considéré le premier langage à fournir des mécanismes d'inférence sur les concepts d'ontologies. [4]

7.4. OWL

La combinaison de RDF/RDF-S et de DAML+OIL a permis l'émergence de OWL (Web Ontology Language), un langage standard de représentation de connaissances pour le Web.

Développé par le groupe de travail sur le Web Sémantique du W3C, OWL peut être utilisé pour représenter explicitement les sens des termes des vocabulaires et les relations entre ces termes. OWL vise également à rendre les ressources sur le Web aisément accessibles aux processus automatisés, d'une part en les structurant d'une façon compréhensible et standardisée, et d'autre part en leur ajoutant des méta-informations. Pour cela, OWL a des moyens plus puissants pour exprimer la signification et la sémantique que XML, RDF, et RDF-S. De plus, OWL tient compte de l'aspect diffus des sources de connaissances et permet à l'information d'être recueillie à partir de sources distribuées, notamment en permettant la mise en relation des ontologies et l'importation des informations provenant explicitement d'autres ontologies.

OWL a trois sous langages de plus en plus expressifs : OWL Lite, OWL DL, et OWL Full:

- **OWL Lite**

  Il supporte les utilisateurs ayant besoin principalement d'une hiérarchie de classification et des contraintes simples (un ensemble est limité à 0 ou 1 élément, par exemple). Il a une complexité formelle inférieure à celle d'OWL DL. OWL Lite supporte seulement un sous ensemble de constructions du langage OWL. [7]

- **OWL DL**

  D'après son nom, OWL DL utilise la logique de description DL. Il a été défini pour les utilisateurs qui réclament une expressivité maximale tout en retenant la complétude informatique (toutes les conclusions sont garanties être calculables), et la possibilité de décision (les calculs finiront en un temps fini). Il inclut toutes les constructions du langage OWL, qui ne peuvent être utilisées que sous certaines restrictions. [7]

- **OWL Full**

  Il a été défini pour les utilisateurs qui veulent une expressivité maximale et une liberté syntaxique de RDF mais sans les garanties informatiques. OWL Full

permet à une ontologie d'augmenter la signification du vocabulaire prédéfini (RDF ou OWL). Il est peu probable que n'importe quel logiciel de raisonnement soit capable de supporter le raisonnement complet de chaque caractéristique d'OWL Full. Autrement dit, en utilisant OWL Full en comparaison avec OWL DL, le support de raisonnement est moins prévisible puisque l'implémentation complète d'OWL Full n'existe pas actuellement.

OWL Full et OWL DL maintiennent le même ensemble de constructions d'OWL. La différence se situe dans les restrictions sur l'utilisation de certaines de ses caractéristiques et sur l'utilisation des caractéristiques de RDF. OWL Lite permet le mélange libre d'OWL avec RDFS et, comme RDFS, n'impose pas une séparation stricte des classes, des propriétés, des individus, et des valeurs de données. [2]

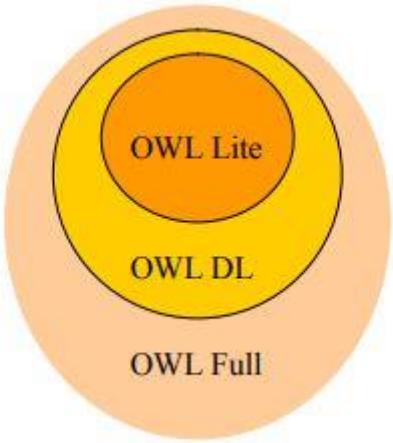

Figure I.3. Les couches de L'OWL. [8]

7.5. F-Logic

F-Logic (La logique de Frame) est un langage de bases de données orienté objet qui combine l'expressivité des langages de bases de données déductives et la richesse de modélisation des modèles orientés objet. [2]

Nous présentons ci-dessous quelques exemples de la création d'ontologie en F-Logic :

C1 :: C2 signifie que le concept C1 est un sous-concept (direct ou indirect) de C2 ;
I : C1 signifie que I est une instance du concept C2 ;
C1 [P=>> C2] signifie que les deux concepts C1 et C2e sont reliés par la propriété P ;
C[A->>V] signifie que le concept C dispose de l'attribut A ayant comme valeur V ;
I1< :I2 signifie que, l'instance I1 fait partie de l'instance I2 ;
etc.

7.6. Protégé

Protégé est un éditeur d'ontologies distribué en open source par l'université en informatique médicale de Stanford. Protégé n'est un outil spécialement dédié à OWL, mais un éditeur hautement extensible, capable de manipuler des formats très divers.

Le support d'OWL, comme de nombreux autres formats, est possible dans protégé grâce à un plugin dédié. Protégé est un outil employé par les développeurs et des experts de domaine pour développer des systèmes basés sur les connaissances (Ontologies).

Des applications développées avec Protégé sont employées dans la résolution des problèmes et la prise de décision dans un domaine particulier. Protégé est aussi une plate-forme extensible, grâce au système de plug-ins, qui permet de gérer des contenus multimédias, interroger, évaluer et fusionner des ontologies, etc. L'outil Protégé possède une interface utilisateur graphique (GUI) lui permettant de manipuler aisément tous les éléments d'une ontologie : classe, méta-classe, propriété, instance,…etc. Protégé peut être utilisé dans n'importe quel domaine où les concepts peuvent être modélisés en une hiérarchie des classes.

Protégé permet aussi de créer ou d'importer des ontologies écrites dans les différents langages d'ontologies tel que : RDF-Schéma, OWL, DAML, OIL, …etc. Cela est rendu possible grâce à l'utilisation de plugins qui sont disponibles en téléchargement pour la plupart de ces langages. [3]

8. Domaines d'applications d'ontologies

8.1. Système d'information

L'intégration d'une ontologie dans un système d'information vise à réduire, voire éliminer, la confusion conceptuelle et terminologique à des points clefs du système, et à tendre vers une compréhension partagée pour améliorer la communication, le partage, l'interopérabilité et le degré de réutilisation possible, ce qui permet de déclarer formellement un certain nombre de connaissances utilisées pour caractériser les informations gérées par le système, et de se baser sur ces caractérisations et la formalisation de leur signification pour automatiser des tâches de traitement de l'information. [1]

L'ontologie retrouve maintenant dans une large famille de systèmes d'information. Elle est utilisée pour :

- Décrire et traiter des ressources multimédia ;
- Assurer l'interopérabilité d'applications en réseaux ;
- Piloter des traitements automatiques de la langue naturelle ;
- Construire des solutions multilingues et interculturelles ;
- Permettre l'intégration des ressources hétérogènes d'information ;
- Vérifier la cohérence de modèles ;
- Permettre les raisonnements temporel et spatial ;
- Faire des approximations logiques ; etc.

Ces utilisations des ontologies se retrouvent dans de nombreux domaines d'applications tel que :

- Intégration d'information géographique ;
- Gestion de ressource humaine ;
- Aide à l'analyse en biologie, suivi médicale informatisé ;
- Commerce électronique ;
- Enseignement assisté par ordinateur ;
- Bibliothèque numériques ;
- Recherche d'informations.

8.2. Web sémantique

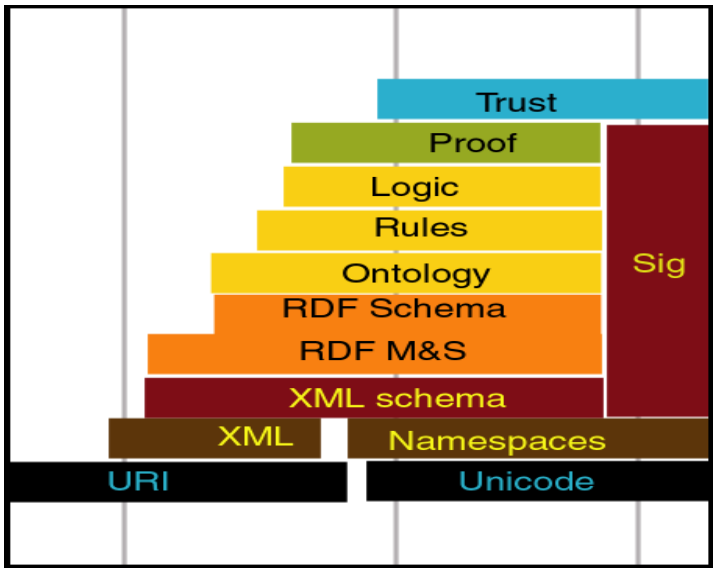

Figure I.4. Pyramide du Web sémantique [1]

Un courant particulièrement prometteur pour l'expansion des systèmes à base d'ontologies est celui du Web sémantique. Il s'agit d'une extension du Web actuel, dans laquelle l'information se voit associée à un sens bien défini, améliorant la capacité des logiciels à traiter l'information disponible sur le Web. L'annotation des ressources d'information du Web repose sur des ontologies, elles sont aussi disponibles et échangées sur le Web.

Grace au Web sémantique, l'ontologie a trouvé un jeu de formalismes standards à l'échelle mondiale, et s'intègre dans de plus en plus d'applications Web, sans même que les utilisateurs ne le sachent. Cela se fait au profit des logiciels qui à travers les ontologies et les descriptions qu'elles permettent, peuvent proposer de nouvelles fonctionnalités exploitant les effets d'échelles du Web pour en améliorer les effets. [1]

9. Conclusion

Dans ce chapitre, nous avons rappelé la notion des ontologies,leur rôle ,leurs composants, leurs différents types, leur cycle de vie, leurs langages de représentation et les différents domaines d'utilisation de ces ontologies.

Partant du fait que plusieurs connaissances peuvent prendre des représentations différentes, on trouve de nos jours plusieurs ontologies de domaine pour un même champ

d'application. Les techniques d'alignement représentent un cadre général, dans lequel plusieurs ontologies hétérogènes peuvent être exploitées et interopérables.

A cet effet, nous allons présenter dans le chapitre suivant les techniques d'alignement d'ontologies existées et les domaines d'application les plus dispersés.

Chapitre II : Alignement d’ontologies

1. Introduction

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. En effet, l'intégration de données est motivée par la forte hétérogénéité des données issues de sources multiples et l'absence de sémantique suffisante pour bien comprendre la signification des données. Citons le domaine biomédical où l'alignement d'ontologies joue un rôle clé dans le développement de la recherche biomédicale en facilitant le développement d'entrepôts de données articulés autour d'ontologies communes.

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. L'alignement d'ontologie est un domaine de recherche actif en raison de son large éventail d'applications.

2. Définition d'alignement

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.

Cependant. trouver les concepts en commun manuellement est une tache fastidieuse, pratiquement infaisable dans le cas des ontologies volumineuses. Par conséquent, trouver des moyens pour automatiser ce processus aura un apport important dans le domaine d'ingénierie d'ontologies plus particulièrement celui du Web sémantique. [4]

3. 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 [12].

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 [13].

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 [10]. 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.

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 [13].

3.1. 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 [14].

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 [12].Dans la littérature, Il existe plusieurs classifications d'hétérogénéité entre les ontologies, se basent sur l'étude du décalage sémantique et structurel qui peut exister entre les ontologies. Ce qui suit, nous présentons les principales formes d'hétérogénéité en quatre niveaux : [13]

- **a) Le niveau syntaxique**

Dans ce niveau, l'hétérogénéité se produit quand deux ontologies sont décrites avec deux langages ontologiques différents [12], 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 [14], 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 [12].

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 [14].

- **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 [14].

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 [12].Dans ce qui suit, nous citons des exemples de ce type d'hétérogénéité [5]:

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 probleme 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.).

- **c) Le niveau conceptuel**

est appelée aussi hétérogénéité sémantique ou la différence logique .Chaque ontologiste, selon son domaine de prédilection pour représenter les concepts de manière

propre et dans une hiérarchie spécifique. 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 [13].

- **La granularité** : des ontologies peuvent représenter les mêmes connaissances [14], 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… [11].

- **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 [13].

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

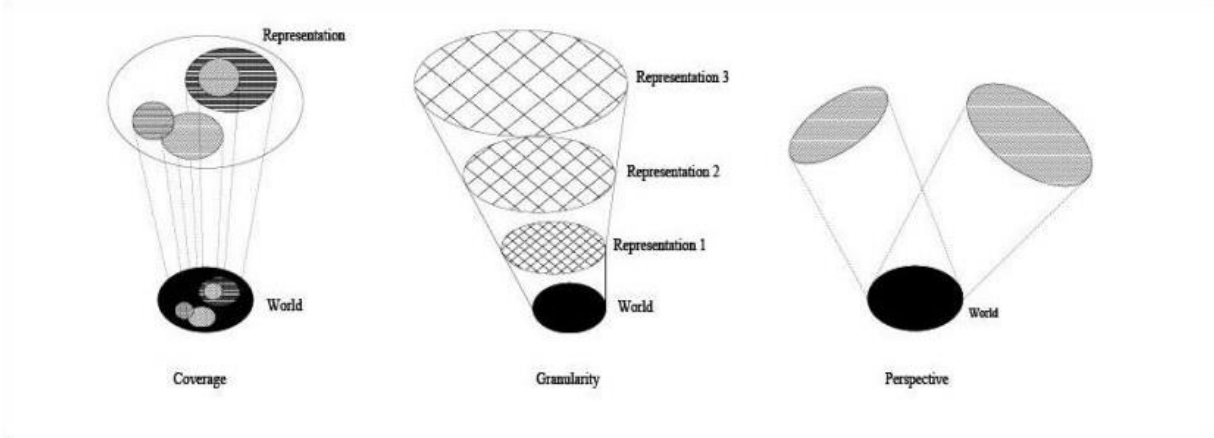

Figure II.1. Les trois dimensions de l'hétérogénéité au niveau conceptuel [13].

- **d) 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 [11].

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 oeuvre les entités ontologiques influence leurs interprétations. De plus, ce type d'hétérogénéité reste difficile à détecter par la machine [12].

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 . 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… [12].

4. Méthodes 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 [10] .

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 [11].

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 : [12]

- $\forall a,b \in O, S(a,b) \geq 0$ *(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 : $O \times O \rightarrow R$ est une fonction d'une paire d'entités à un nombre réel exprimant la dissimilarité entre ces deux entités telle que: [12]

- $\forall a, b \in O, DS(a, b) \geq 0$ *(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. [12]

- $\forall a, b \in O, D(a, b) = 0 \Leftrightarrow a = b$ *(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. [5]

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 **[12].**

4.1. 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…etc. Ces méthodes peuvent encore être divisées en deux sous-catégories: méthodes syntaxiques et linguistiques.

4.1.1. Méthodes syntaxiques

Ces méthodes analysent la structure des chaines 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 chaines à comparer avant de les fournir aux fonctions calculant la similarité. 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»[5].

4.1.2. 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 dernier 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) [15].

4.1.2.1. 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 [13]. 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 infléchie 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[12].

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 [13].

4.1.2.2. 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 [11].

4.2. 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) [15].

4.2.1. 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 [14].

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 [11].

4.2.2. 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 soeurs. L'idée de base est que deux entités sont similaires, si leurs voisines sont similaires [12].

4.3. 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 [13].

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 [5].

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

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 : $2^E * 2^E$ [0,1] telle que : [5]

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

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.

4.4. 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 [4].

4.4.1. 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 ambigüités concernant les différentes significations possibles des termes. Des examples d'ontologies

intermediaries: FMA « the Foundational Model of Anatomy » , CYC ontology et SUMO « the Suggested Upper Merged Ontology ». [5]

4.4.2. 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 subsomption) peuvent être employées. Elles permettent de vérifier les relations sémantiques entre les entités telles que l'équivalence (la similarité est égale à 1), la subsomption (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[11].

4.5. 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 [12].

5. Domaines d'application d'alignement d'ontologies

L'expansion du Web sémantique et l'adoption des ontologies comme moyen de décrire les ressources sur le Web a accentué davantage ce besoin. Ainsi, les domaines d'applications manifestant le besoin aux outils d'alignement sont multiples comme I'E-commerce, la fouille d'information. Le but par ces cas d'application est de montrer l'intérêt de l'alignement en général -celui des ontologies en particulie.

5.1. Communication entre les agents

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

Suivant le modèle courant, dans le cas des agents mobiles dits cognitifs, le comportement de ces agents est considéré comme un ensemble de croyances, désirs et attentions exprimés symboliquement dans des langage inspérés de "speech-act" pour intéragir entre eux. Ces langages déterminent "l'enveloppe" des messages interchangés entre les agents et leur permettent de situer ces messages dans un contexte d'intéraction. Cependant, ces langages ne speciflent pas le contenu des messages qui peut être exprimé dans des langages de représentation de connaissances différents.. Par conséquent, pour que les agents se comprennent mutuellement, il faudrait qu'ils soient en mesure de traduire leurs messages et de les aligner via l'intégration des ponts d'axiomes dans leur propres modèles.

Une première solution à ce problème consiste à adopter un protocole d'alignement d'ontologies, qui se déclenche à la réception d'un message d'un autre agent faisant référence à une ontologie hétérogène. [4]

5.2. Les services Web sémantiques

Les services Web sont des processus qui exposent leurs interfaces pour offrir des fonctionalités à d'autres applications pour des raisons de complémentarité. Les services Web sémantiques offrent un moyen plus riche et plus précis pour décrire les services via l'utilisation des langages de représentation de connaissances et d'ontologies. 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. Toutefois, la description de tels services fait référence à des ontologies

multiples généralement exprimées dans des modèles et des langages hétérogènes. Par conséquent, l'opération d'intégration des services nécessite l'établissement de correspondances entre les termes d'ontologies.

A tire d'illustration, pennons le cas de deux services Web A et B dans le contexte du E-commerce.En effet, les deux services pouvent ne pas partager le même modèle (ontologie) pour décrire les fonctionalités qu'ils supportent. Par exemple, le service B fourni la possibilté d'annuler une commande alors que le A ne la supporte pas, ou le cas inverse, le service A implémente la fonctionalité de changer une commande alors que B ne la fourni pas. Dans ce genre de situation, pour que les deux services Web s'engagent eventuellement dans des transactions de vente-achat, une vérification des fonctionalités suportées exige une reconciliation entre les deux ontologies à l'avance pour déterminer les fonctionalités en commun. Ceci revient à aligner les deux ontologies, en temps réel à l'aide d'outils d'alignement. [4]

5.3. 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 [5].

La technologie des réseaux Peer-to-Peer a été déployée pour le partage des resources entre les applications (exemple KaZaA, Napster). Ces réseaux se caractérisent par leur dynamisme et leur flexibilité. Toutefois, les noeuds des réseaux 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 ou faire usage à des ontologies. Pour assurer un échange efficace d'information entre les noeuds, il faudrait identifier et caractériser les relations de leurs schérnas/ontologiesd'une manièreinstantanée lors de l'échange de données. [4]

5.4. La recherche dans les catalogues d'E-commerce

Les applications B2B (le cas d'Amazon, e-Bay etc) 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 UNSPSC (http://www.unspsc.org) est largement utilisé pour représenter les

besions du vendeur, alors que le standard ecl@ss (http://www.eclass-online.corn) est utilisé pour représenter les besions 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.[4]

5.5. Le Web Biomédical

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, ainsi que de riches banques de documents généralistes ou plus spécifiques (par exemple, sur les maladies rares ORPHANET, RAREDISEASE), 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 [5].

6. Conclusion

Dans ce chapitre, nous avons rappelé la définition d'alignement d'ontologies, la naissance du besoin à l'alignement des ontologies, les différentes méthodes principales de l'alignement et les domaines d'application les plus dispersés.

A cet effet, nous allons présenter dans le chapitre suivant les approches existants déjà dans ce domaine qu'on est en train de traiter et évoluer et la représentation d'une synthèse comparative de ces approches.

Chapitre III : Les approches existantes

1. Introduction

L'alignement d'ontologies représente un grand intérêt dans le domaine de la gestion des connaissances hétérogènes. L'alignement d'ontologies repose sur le calcul des mesures de similarité.

La littérature du domaine propose plusieurs méthodes d'alignement. Ces méthodes exploitent des différents formats de représentations des ontologies et des différentes méthodes de mesures de similarité.

Dans ce chapitre, nous allons présenter quelques approches existantes dans la littérature qui concernent le problème d'alignement d'ontologies .

2. Les approches existantes

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 :

2.1. 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 noeuds du graphe, deux noeuds 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 noeuds)
- $E : V \times V$ 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 noeuds u et se terminant au noeuds v [16].

2.2. 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.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 [12].

2.3. 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 noeuds 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. [16]

2.4. SODA

La nouvelle méthode d'alignement, SODA (Structural Ontology OWL-DL Alignement), 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. [17]

2.5. TaxoMap

TaxoMap est un outil d'alignement qui a pour objectif de permettre un accès unifie 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 [14].

2.6. 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[11].

2.7. 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 [12].

2.8. FALCON-OA

Falcon est un outil d'alignement qui a été développé par Wei Hu pour 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 III.1 montre le graphe RDF et le graphe RDF biparti d'une ontologie [14].

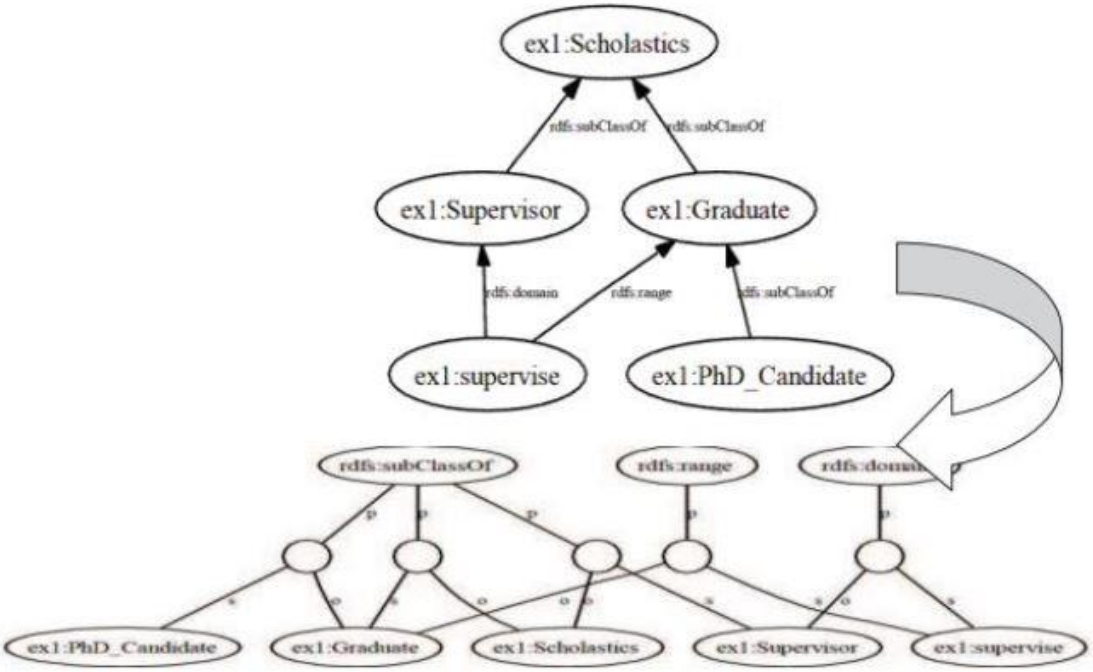

Figure III.1. Le graphe RDF et le graphe RDF biparti d'une ontologie OA. [5]

3. Synthèse comparative entre les approches existantes

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.

3.1. 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. Le Tableau III.1 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 subsomption. Seule la méthode ASCO2 laisse un choix quant à la cardinalité des alignements qu'elle produit [14].

Méthode	Données d'entrée	Données de sortie	
	Types de schémas	**Format**	**Relation**
Flacon-AO	RDFS / OWL	RDF/ XML	↔
Anchor-Prompt	RDF(S), OWL	RDF(S), XML	↔
TaxoMap	OWL/ RDF(taxonomies)	RDF/ XML	↔
OLA	RDFS / OWL	RDF/ XML	↔
ASCO3	OWL DL/ Lite	RDF/ XML	↔
QOM	RDFS/ OWL	RDF/XML	↔

Tableau III.1. Tableau de comparaison des approches par rapport à leurs dimensions externes [5].

3.2. Comparaison selon les dimensions internes d'une méthode d'alignement

Le Tableau III.2 montre, pour chaque méthode d'alignement:

- sa représentation interne : généralement les méthodes transforment les ontologie 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 [14].

Approches	**Représentation interne**	**Techniques utilisées**	**Mesure de similarité**	**Pattern d'exploitation**
Falcon – AO	Graphe direct bipartie	Terminologique /structurelle	Edition distance,TF,IDF	RDF triples
TaxoMap	Format TaxoMap (seulement les étiquettes et les sous classes sont prends en considération	Terminologique /structurelle	Techniques basés sur la mesure de similarité de Lin (SimLinLike)	Caractéristiques des entités (la richesse des étiquettes des entités)
OLA	OL-Graphes	Terminologique /structurelle	Hamming entre les synsets,systéme d'équation-s interdependentes	Caractéristiques des entités (descriptive knowledge about a couple of entités)
ASCO2	Format RDF	Terminologique	Jaro-Winkler (id,labels,synset	RDF triples

		/structurelle	),TF, IDF	
QOM	Format RDF(S)	Terminologique /structurelle	Distance d'édition	Caractéristiques des entités

Approches	**Paramétrage et composition**
Falcon – AO	Composition (intégration d'alignement), critère d'arrêt de l'algorithme itératif de propagation de similarité,seuil de similarité,la langue utilisée dans les ontologies (stemmer)
TaxoMap	Composition(séquentielle), Les catégories de mots considérées par Treetagger,les différents seuils de similarité (Equiv.threshold,HiddenInc.thresholdSim,..),la langue utilisée dans les ontologies.
OLA	Composition (moyenne pondérée,paralléle),pondération (poids),seuil de similarité,un alignement en entrée,WordNet.
ASCO2	Composition(la somme pondérée avec les poids variables,paralléles),pondération variable (poids),seuil de similarité,Wordnet.
QOM	composition(moyenne pondérée,fonctionsigmoide),critére d'arrét de l'algorithme itératifde propagation de similarité,seil de similarité.

Tableau III.2. Tableau de comparaison des approches par rapport à leurs dimension interne [14].

4. Conclusion

Dans ce chapitre, nous avons présenté les différentes approches existantes et utilisées dans la littérature qui attaque le problème de l'alignement et de la correspondance entre deux entités en général, qui apparaissent dans des ontologies. Puis nous avons fait une synthèse comparative entre les différentes approches que nous avons cité selon leurs dimensions internes et leurs dimensions externe.

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. En fin nous allons citer quelques étapes principales de notre algorithme d'application effectué et son déroulement pour arriver aux résultats souhaités.

Chapitre IV : Conception

1. Introduction

Dans ce chapitre, nous présentons la conception de notre system, nous allons présenter une architecture générale pour le fonctionnement de notre application ainsi que les méthode utilisées pour calculer la similarité entre les entités d'ontologies et en fin nous allons citer quelques étapes principales de notre algorithme d'application effectué et son déroulement pour déterminer la similarité entres les différentes entités des deux ontologies du domaine.

Les valeurs de similarité calculées par les méthodes linguistiques,sémantiques ou structurelles sont appelées les valeurs de similarité partielles entre deux entités, et elles sont stockées dans une variable de similarité. Les valeurs de similarité partielles sont calculées par des mesures normalisées, elles sont donc comprises dans l'intervalle de 0 à 1.

Ces valeurs de similarité partielles 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, deux entités sont considérées comme similaires (équivalentes)ou différentes. Tout ceci est développé dans les sections suivantes.

2. Approche proposée :

Pour arriver au but de notre application, nous avons utilisé trois paires d'ontologies déjà existantes.

Notre système realisé dépend sur quelques grandes lignes que nous allons présenter comme suite :

- La normalisation des entités des deux ontologies utilisées.
- Le calcul de la similarité entre les entités des deux ontologies utilisées en basant sur plusieurs méthodes (méthode terminologique, méthode sémantique, méthodes structurelle).
- La génération des résultats représentés par les concepts similaires ou les concepts proches entre les deux ontologies utilisées dans l'application de notre système.

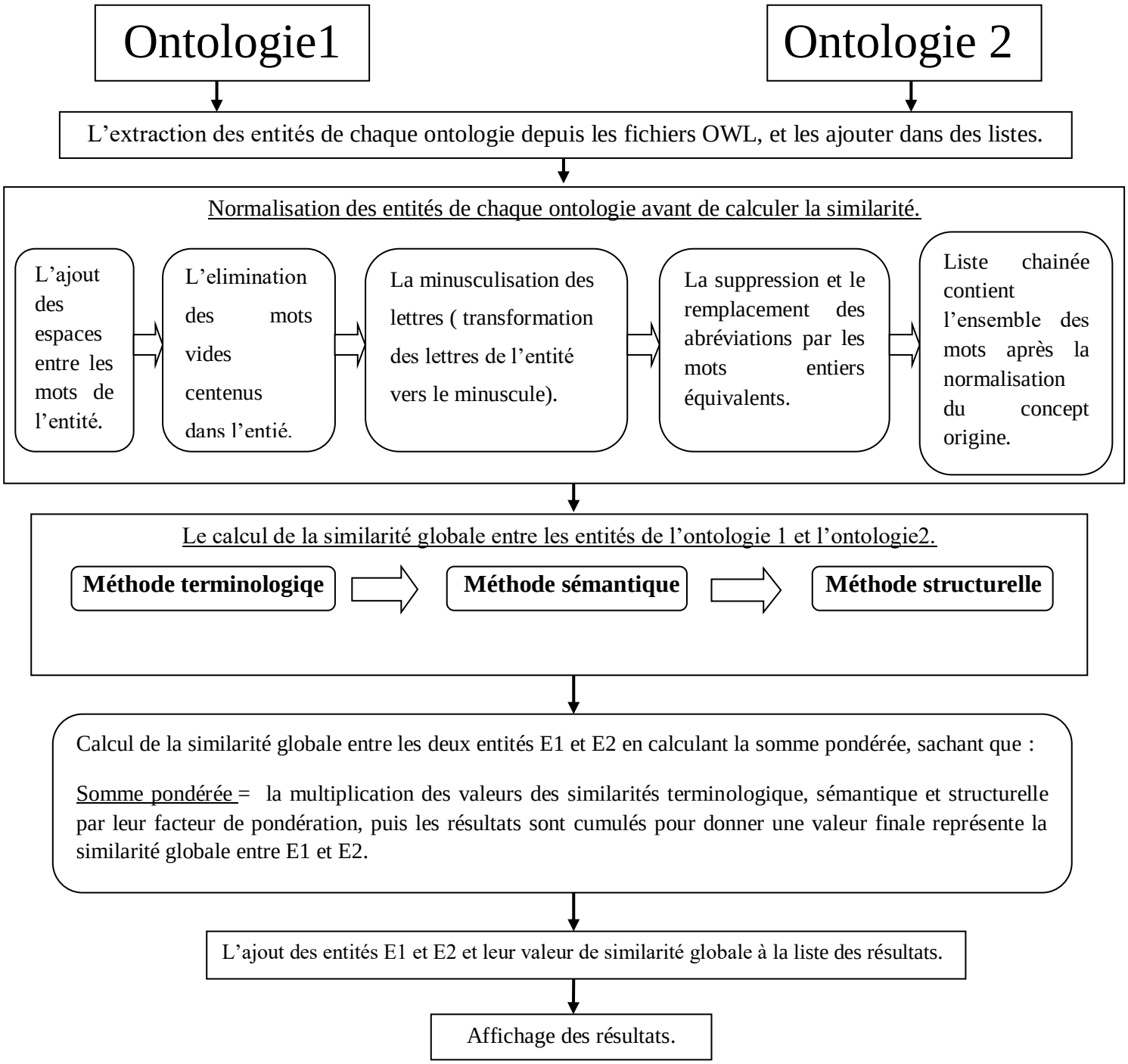

Figure IV.1. Architecture générale du fonctionnement de l'application.

2.1. Problème de calcul de la 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.

Une méthode pour calculer la similarité sémantique entre les mots et aussi entre les phrases peut être appliquée dans une variété de domaines. 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 valeurs de similarité calculées sont stockées dans une variable de similarité. Elles sont calculées par des mesures normalisées, elles sont donc comprises dans l'intervalle de 0 à 1 En examinant cette valeur de similarité finale par rapport à un seuil, deux entités sont considérées comme similaires (équivalentes),proches ou différentes.

2.2. Architecture détaillée de l'algorithme d'alignement proposé

Le principe de fonctionnement de l'algorithme proposé dépend sur des grandes lignes, qui sont :

- La lecture des deux ontologies.
- L'extraction des concepts de chaque ontologie, et les stocker dans des listes.
- La normalisation des concepts.
- Le calcul de la similarité terminologique,sémantique et structurelle.
- La combinaison des valeurs des similarités précedentes grâce à une somme pondérée.
- L'obtention de la similarité globale entre chaque deux concepts des deux ontologies.

La valeur de la similarité globale est entre [0,1], cette valeur va être comparé avec un seuil, seul les concepts qui ont une similarité globale entre eux supérieure ou égale au seuil vont être pris en considération.

2.3. Méthodes utilisées

Dans cette partie, on va présenter les méthodes essentielles utlisées par notre système pour réaliser l'alignement d'ontologies.

2.3.1. Méthode de normalisation

Le système que nous traitons nécéssite une méthode de normalisation des concepts avant de commancer l'alignement entre deux ontologies décrites en format OWL.

Cette méthode permet de normaliser les concepts pour extraire le sens et la relation entre ces derniers.

La méthode utilisée pour la normalisation des concepts est nommée : «concept_normalisation», le principe de fonctionnement de cette méthode est :

Concept_normalisation(chaine)

```
string= " " ;
Pour chaque caractère dans chaine :
    Si caractère est minuscule :
        string=string+caractère ;
    Fin Si
    Si caractère est majuscule ou caractère est le dernier dans la chaine :
        Si contain(string,stopwords)=false et string.equal(“”)=false :
            Ajouter string à la liste des mot ;
        Fin Si
        string=“” ;
        string+=caractère;
    Fin Si
Fin Pour

Pour chaque mot dans la liste concept :
```

Transformer le mot en minuscule ;

Fin Pour

Pour chaque mot dans la liste concept :

Pour chaque abréviation dans la liste des abréviations :

Si mot est trouvé dans la liste des abréviation :

Remplacer l'abréviation par le mot entier ;

Fin Si

Fin Pour

Fin Pour

Retourner concept : la liste des mots.

2.3.2. Méthode de calcul de la similarité terminologique

Cette méthode permet de calculer la similarité entre les deux concepts des deux ontologies en basant sur les chaines de caractères seulement, cette méthode est l'implémentation de la méthode «JaroWinkler**».**

La méthode utilisée pour le calcul de la similarité terminologique est nommée : «apply» de la classe « JaroWinkler ». Le principe de fonctionnement de cette méthode est :

apply (left , right)

Si (left =null) ou (right=null) :

Les chaînes ne doivent pas être vides ;

Fin Si

match = max(longeur_première chaine, la longeur_deuxième chaine) / 2 – 1 ;

```
jaroComparaisonSim(left,right)
    = ( matches(left,right) / left.length()
       + matches(left,right) /right.length()
       + (matches(left,right) - transposes(left,right))  / matches(left,right) )
       / 3 ;

  jaroDistance(left,right) = 1 – jaroComparaisonSim(left,right) ;

jaroWinklerComparaisonSim(left,right,boostThreshold,prefixSize)
    = jaroMeasure(left,right) < = boostThreshold
    ? jaroMeasure(left,right)
    : jaroMeasure(left,right)
      + 0.1 * prefixMatch(left,right,prefixSize)
            * (1.0 - jaroDistance(left,right)) ;

  jaroWinklerDistance(left,right,boostThreshold,prefixSize)
    = 1 - jaroWinklerComparaisonSim(left,right,boostThreshold,prefixSize) ;
```

Retourner JaroWinklerComparaisonSim ;

2.3.3. Méthode de calcul de la similarité sémantique

Cette méthode permet de calculer la similarité entre les deux concepts des deux ontologies en basant sur le sens.

Cette méthode est l'implémentation de la méthode «Wu & Palmer**»,** elle calcule la connexité en considérant les profondeurs des deux synsets dans les taxonomies WordNet, ainsi que la profondeur du plus petit sous-client (LCS) sous la formule :

$$SimWuPalmer = 2 * \frac{profondeur(lcs)}{profondeur(S1) + profondeur(S2)}.$$

Cela signifie que $0 < SimWuPalmer <= 1$. $SimWuPalmer$ ne peut jamais être nul car la profondeur du LCS n'est jamais nulle (la profondeur de la racine d'une taxonomie est égale à un). $SimWuPalmer$ est égal à 1 si les deux concepts d'entrée sont identiques.

Le principe de fonctionnement de cette méthode est comme suite :

Compute(mot1,mot2)
Si (mot1=null) ou (mot2=null) : Les chaînes ne doivent pas être vides ; **Fin Si** Wu_palmer_Sim(mot1,mot2)=2*profondeur(lcs) /(profondeur(mot1)+profondeur(mot2)); **Retourner** Wu_palmer_Sim;

2.3.4. Méthodes de calcul de la similarité structurelle

Ces méthodes permettent de calculer la similarité entre les deux concepts des deux ontologies en basant leurs positions dans une hiérarchie et en fonction des informations structurelles.

- La méthode «structural_interne» calcule la similarité terminologique et sémantique entre ces 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.

  Le pricipe de fonctionnement de cette méthode est :

structural_interne(concept1,concept2)

```
simstructurelleinterneglob=0 ;
compteur_global=0 ;
SimStructurelle=0 ;
poids_terminologique ;
poids_sémantique ;
Si (la taille de la liste propertiesOnto1(concept1) !=0  et la taille de la liste propertiesOnto2(concept2) :
Pour chaque propriété1 de la liste des propriétés du concept1 :
   Pour chaque propriété2 de  la liste des propriétés du concept2 :
      Pour chaque mot1 de la liste des mot de la propriété1 :
         Pour chaque mot2 de la liste des mots de la propriété2 :
      Similarité_terminologique = JaroWinkler.apply(mot1, mot2);
      result_bcc+= Similarité_terminologique ;
      Similarité_sémantique= SimilarityCalculationDemo.compute(mot1, mot2);
      result_sem+= Similarité_sémantique ;
      compteur1++ ;
         Fin Pour
resultatbcc+=resultbcc/compteur1;
resultatsem+=resultsem/compteur1;
```

```
 compteur2++;
Fin Pour
finalBCC= resultatbcc/compteur2 ;
finalSEM=resultatsem/compteur2 ;
  SimStructurelle+= (finalBCC *poids_terminologique)+
                    (finalSEM * poids_sémantique) ;
        Compteur_global++ ;
  Fin Pour
Fin Pour
Simstructurelleinterneglob= SimStructurelle/ compteur_global ;
Sinon  Simstructurelleinterneglob=0 ;
Fin Si
Retourner Simstructurelleinterneglob ;
```

- La méthode «structural_externe» calcule la similarité terminologique et sémantique entre ces deux concepts en basant sur la structure externe de l'entité et exploite des relations entre elles-mêmes.

  Le pricipe de fonctionnement de cette méthode est :

```
structural_externe(concept1,concept2)
compteur_global=0 ;
SimStructurelle=0 ;
simstructurelleexterneglob=0 ;
poids_terminologique ;
poids_sémantique ;
Si (concept1 a des superclasses et concept2 a des superclasses) :
  Pour chaque superclasse1 dans la liste des superclasse du concept1 :
      Pour chaque superclasse2 dans la liste des superclasses du concept2 :
        Pour chaque mot1 dans la liste des mots de la superclasse1 :
            Pour chaque mot2 dans la liste des mots de la superclasse2 :
```

```
Similarité_terminologique = JaroWinkler.apply(mot1,mot2);
Similarité sémantique= SimilarityCalculationDemo.compute(mot1,mot2);
resultbcc+= Similarité_terminologique;
resultsem+= Similarité sémantique ;
compteur1++ ;
    Fin Pour
resultatbcc+=resultbcc/compteur1;
resultatsem+=resultsem/compteur1;
compteur2++;
  Fin Pour
finalBCC=resultatbcc/compteur2;
finalSEM=resultatsem/compteur2;
   SimStructurelle+= (finalBCC *poids_terminologique)+
                     (finalSEM *poids_sémantique) ;
  compteur_global++ ;
      Fin Pour
  Fin Pour
Simstructurelleexterneglob=SimStructurelle/compteur_global ;
Sinon   Simstructurelleexterneglob=0 ;
Fin Si
Retourner Simstructurelleexterneglob ;
```

2.3.5. Méthode de calcul de similarité globale

Cette méthode est nommée « calcul » dans la classe « ontologya » , elle permet de calculer la similarité globale entre deux concepts en faisant la combinaison entre les trois (03) méthodes de similarité précédentes, la méthode de similarité terminologique, la méthode de similarité sémantique et la méthode de similarité structurelle.

Le principe de fonctionnement de cette méthode est comme suite :

Calculate()

```
resultat ← créer une liste vide pour stocker les resultats de la methode.
Pour chaque classe1 dans la liste des classes de l'ontologie 1 :
    Pour chaque classe2 dans la liste des classes de l'ontologie 2 :
        Pour chaque mot1 de la liste des mots de la classe1 :
            Pour chaque mot2 de la liste des mots de la classe2 :
                d= JaroWinkler.apply(mot1,mot2);
                result+=d ;
                compteur1++ ;
            Fin Pour
            res+=result ;
            resultat_syntax+=result/compteur1 ;
            compteur2++ ;
        Fin Pour
    resultat_syntax=resultat_syntax/compteur2 ;
    Si (resultat_syntax==1) :
    Si ( les deux classes ne sont pas déjà dans la liste des résultats) :
    Ajouter les deux classe à la liste des résultats avec leurs valeur de similarité ;
    Fin Si
Sinon :
resultat _syntax=resultat_syntax*poids_syntaxique ;
resultat_globale+=resultat_syntax ;
Pour chaque mot1 de la liste des mots de la classe1 :
  Pour chaque mot2 de la liste des mots de la classe2 :
```

```
d=SimilarityCalculationDemo.compute(mot1,mot2);
result+=d ;
compteur1++ ;
    Fin Pour
res+=result ;
resultat_sémantique+=result/compteur1 ;
compteur2 ++;
Fin Pour
resultat_sémantique=resultat_sémantique/compteur2 ;
  Si (resultat_sémantique=1) :
     Si (les deux classes ne sont pas déjà dans la liste des résultats) :
       Ajouter les deux classes à la liste des résultats avec leur valeur de similarité ;
     Fin Si
  Sinon :
resultat _sémantique=resultat_sémantique*poids_sémantique ;
resultat_globale+=resultat_sémantique ;

   resultat_structurelle_interne=structural_interne(classe1,classe2) ;
   resultat_structurelle_externe=structural_externe(classe1,classe2) ;
  structurelle_globale=resultat_structurelle_interne*poids_interne+
resultat_structurelle_externe*poids_externe ;
   Si (structurelle_globale=1) :
      Si (les deux classes ne sont pas déjà dans la liste des résultats) :
      Ajouter les deux classes à la liste des résultats avec keur valeur de similarité ;
      Fin Si

  Sinon :
  structurelle_globale=structurelle_globale*poids_structurel ;
  resultat_globale+=structurelle_globale ;
  Si (les deux classes ne sont pas déjà dans la liste des résultats) :
  Ajouter les deux classes à la liste des resultats avec leur valeur de similarité ;
```

Fin Si
Fin Si
Fin Si
Fin Si
Fin Pour
Fin Pour

Pour chaque propriete1 de la liste des proprietes de l'ontologie1 :
Pour chaque propriete2 de la liste des proprietes de l'ontologie2 :
Pour chaque mot1 de la liste des mots de la propriete1 :
Pour chaque mot2 de la liste des mots de la propriere2 :
d=JaroWinkler.apply(mot1,mot2);
result+=d ;
compteur1++ ;
Fin Pour
res+=result ;
resultat_syntax+=result/compteur1 ;
compteur2++ ;
Fin Pour
resultat_syntax= resultat_syntax/compteur2 ;
Si (resultat_syntax=1) :
Si (les deux proprietes ne sont pas déjà dans la liste des résultats) :
Ajouter les deux proprietes à la liste des resultats avec leur valeur de similarité ;
Fin Si

Sinon :
resultat_syntax=resultat_syntax*poids_syntaxique ;
resultat_globale+=resultat_syntax
Pour chaque mot1 de la liste des mots de la propriete1 :
Pour chaque mot2 de la liste des mots de la propriete2 :
d=SimilarityCalculationDemo.compute(mot1,mot2);

```
      result+=d ;
      compteur1++ ;
    Fin Pour
  res+=result ;
  resultat_sémantique+=result/compteur1 ;
  compteur2++ ;
 Fin Pour
resultat_sémantique=resultat_sémantique/compteur2 ;
 Si ( resultat_sémantique=1) :
   Si (les deux proprietes ne sont pas déjà dans la liste des résultats) :
   Ajouter les deux proprietes à la liste des resultats avec leur valeur de similarité ;
   Fin Si
 Sinon :
  resultat_sémantique=resultat_sémantique*poids_sémantique ;
  resultat_globale+=resultat_sémantique ;
  Si (les deux proprietes ne sont pas déjà dans la liste des résultats) :
  Ajouter les deux proprietes à la liste des resultats avec leur valeur de similarité ;
  Fin Si
Fin Si
Fin Si
Fin Pour
Fin Pour

Pour chaque individu1 dans la liste des individus :
  Pour chaque individu2 dans la liste des individus :
    Pour chaque mot1 de la liste des mots de l'individu1 :
      Pour chaque mot2 de la liste des mots de l'individus 2 :
        d=JaroWinkler.apply(mot1,mot2);
        result+=d ;
       compteur1++ ;
      Fin Pour
    res+=result ;
    resultat_syntax+=result/compteur2 ;
    compteur2 ;
  Fin Pour
 resultat_syntax=resultat_syntax/compteur2;
Si (resultat_syntax=1) :
  Si ( les deux individus ne sont pas déjà dans la liste des resultats) :
   Ajouter les deux individus à la liste des resultats avec  leur valeur de similarité ;
  Fin Si

Sinon :
 resultat_syntax=resultat_syntax*poids_syntaxique ;
 resultat_globale+=resultat_syntax ;
```

```
   Pour chaque mot1 de la liste des mot de l'individu1 :
      Pour chaque mot2 de la liste des mots de l'individus2 :
         d=SimilarityCalculationDemo.compute(mot1,mot2);
         result+=d ;
         compteur1++ ;
      Fin Pour
    res+=result ;
    resultat_sémantique+=result/compteur1 ;
    compteur2++ ;
 Fin Pour
resultat_sémantique=resultat_sémantique/compteur2;
Si (resultat_sémantique=1) :
  Si (les deux individus ne sont pas déjà dans la liste des resultats) :
   Ajouter les deux individus à la liste des resultat avec leur valeur de similarité ;
  Fin Si
Sinon :
 resultat_sémantique=resultat_sémantique*poids_sémantique ;
 resultat_globale+=resultat_sémantique ;
 Si (les deux individus ne sont pas déjà dans la liste des resultats) :
   Ajouter les deux individus à la liste des resultats avec leur valeur de similarité ;
Fin Si
Fin Si
Fin Si
Fin Pour
Fin Pour

Retourner resultat ;
```

3. Conclusion

Dans ce chapitre, nous avons présenté notre algorithme proposé et réalisé pour arriver au but de faire l'alignement des ontologies du domaine, en expliquant son principe de fonctionnement et les méthodes principales utilisées pour arriver aux résultats qui sont représentés par une liste des correspondances entre les concepts des deux ontologies à aligner.

Chapitre V : Implémentation

1. Introduction

Dans ce chapitre, nous présentons l'implémentation et l'évaluation de l'algorithme d'alignement d'ontologies proposé sur trois paires d'ontologies existantes que nous allons présenté.

Le développement de notre application se fait en Java car ce langage permet l'utilisation de plusieurs API permettant de manipuler les ontologies: ex, Jena.

2. Outils de développement

Avant de commencer l'implémentation de notre système, nous allons présenter et expliquer les outils utilisés et exploités pour réaliser ce système.

- **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 JAVA**

  Java est un langage de programmation orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au SunWorld.

  La société Sun a été ensuite rachetée en 2009 par la société Oracle qui détient et maintient désormais Java.

  La particularité et l'objectif central de Java est que les logiciels écrits dans ce langage doivent être très facilement portables sur plusieurs systèmes d'exploitation tels que Unix, Windows, Mac OS ou GNU/Linux, avec peu ou pas de modifications, mais qui ont l'inconvénient d'être plus lourd à l'exécution (en mémoire et en temps processeur) à

cause de sa machine virtuelle. Pour cela, divers plateformes et frameworks associés visent à guider, sinon garantir, cette portabilité des applications développées en Java.

Le langage Java reprend en grande partie la syntaxe du langage C++. Néanmoins, Java a été épuré des concepts les plus subtils du C++ et à la fois les plus déroutants, tels que les pointeurs et références, ou l'héritage multiple contourné par l'implémentation des interfaces, et même depuis la version 8, l'arrivée des interfaces fonctionnelles introduit l'héritage multiple (sans la gestion des attributs) avec ses avantages et inconvénients tels que l'héritage en diamant. Les concepteurs ont privilégié l'approche orientée objet de sorte qu'en Java, tout est objet à l'exception des types primitifs (nombres entiers, nombres à virgule flottante, etc.) qui ont cependant leurs variantes qui héritent de l'objet Object (Integer, Float, ...).

Java permet de développer des applications client-serveur. Côté client, les applets sont à l'origine de la notoriété du langage. C'est surtout côté serveur que Java s'est imposé dans le milieu de l'entreprise grâce aux servlets, le pendant serveur des applets, et plus récemment les JSP (JavaServer Pages) qui peuvent se substituer à PHP, ASP et ASP.NET.

Java a donné naissance à un système d'exploitation (JavaOS), à des environnements de développement (eclipse/JDK), des machines virtuelles (MSJVM (en), JRE) applicatives multiplate-forme (JVM), une déclinaison pour les périphériques mobiles/embarqués (J2ME), une bibliothèque de conception d'interface graphique (AWT/Swing), des applications lourdes (Jude, Oracle SQL Worksheet, etc.), des technologies web (servlets, applets) et une déclinaison pour l'entreprise (J2EE). La portabilité du bytecode Java est assurée par la machine virtuelle Java, et éventuellement par des bibliothèques standard incluses dans un JRE. Cette machine virtuelle peut interpréter le bytecode ou le compiler à la volée en langage machine. La portabilité est dépendante de la qualité de portage des JVM sur chaque OS.

- **Environnement Eclipse**

Eclipse est un projet, décliné et organisé en un ensemble de sous-projets de développements logiciels, de la fondation Eclipse visant à développer un environnement de production de logiciels libre qui soit extensible, universel et polyvalent, en s'appuyant principalement sur Java.

Son objectif est de produire et fournir des outils pour la réalisation de logiciels, englobant les activités de programmation (notamment environnement de développement intégré et frameworks) mais aussi d'AGL recouvrant modélisation, conception, test, gestion de configuration, reporting… Son EDI, partie intégrante du projet, vise notamment à supporter tout langage de programmation à l'instar de Microsoft Visual Studio.

Bien qu'Eclipse ait d'abord été conçu uniquement pour produire des environnements de développement, les utilisateurs et contributeurs se sont rapidement mis à réutiliser ses briques logicielles pour des applications clientes classiques. Cela a conduit à une extension du périmètre initial d'Eclipse à toute production de logiciel : c'est l'apparition du framework Eclipse RCP en 2004.

Figurant parmi les grandes réussites de l'open source, Eclipse est devenu un standard du marché des logiciels de développement, intégré par de grands éditeurs logiciels et sociétés de services. Les logiciels commerciaux Lotus Notes 8, IBM Lotus Symphony ou WebSphere Studio Application Developer sont notamment basés sur Eclipse.

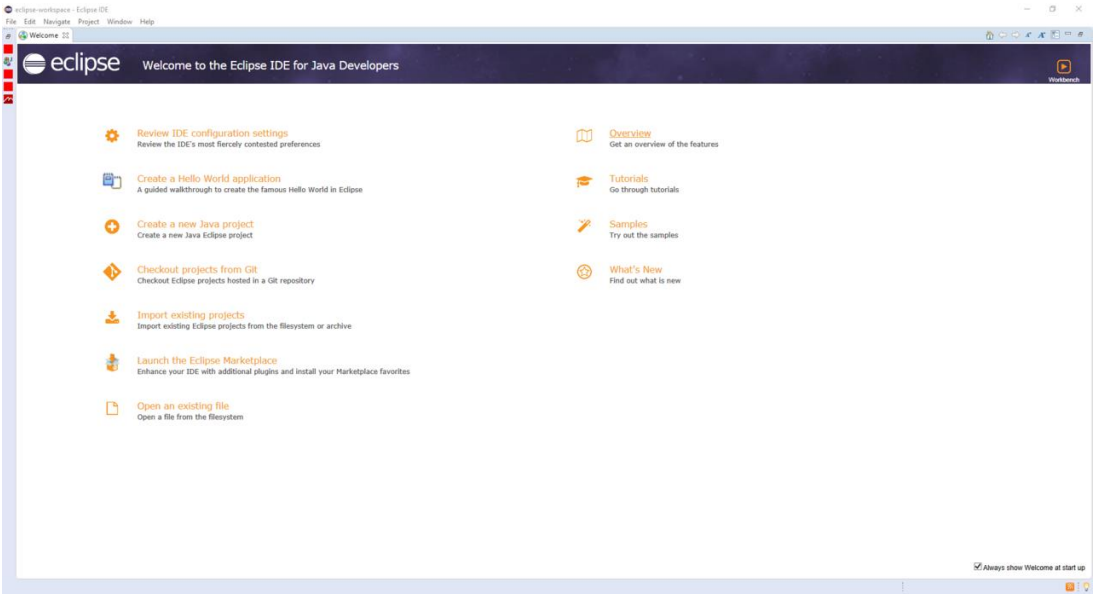

Figure V.1. Environnement de développement (Eclipse).

- **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.

3. Les ontologies de test

Pour appliquer l'alignement, on a utilisé 3 paires d'ontologies déjà existantes sur plusieurs domaines. Ces paires qu'on a utilisé pour tester notre application sont très grandes, elles se composent d'un nombre très énormes 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 avons utilisé les ontologies proposées, qui sont juste une partie de chaque ontologie pour réduire le temps d'exécution, vu qu'il est très important.

3.1. Les ontologies utilisées et pourquoi ?

3.1.1. Ontologie « Person »

Nous avons délibérément utilisé une paire d'ontologies qui soient petites, pour tester notre application en un temps raisonnable. Ces ontologies se composent d'un nombre très limité d'entités, ce qui rend le temps d'exécution de l'application assez rapide en comparaison avec d'autres ontologies plus grandes. La structure des ontologies de cette paire est assez similaire, mais les entités sont nommées différemment (par ex. Human et Person), ce qui montre la puissance de l'algorithme.

Les figure IV.2, IV.3 montrent les représentations des O-Graphes comme suite :

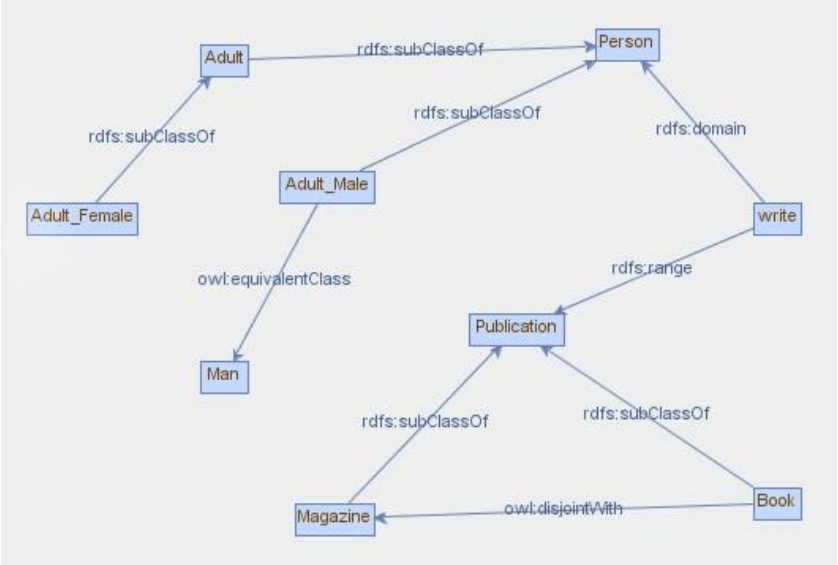

Figure IV.2. Représentation OGraphe de l'ontologie Person1.owl

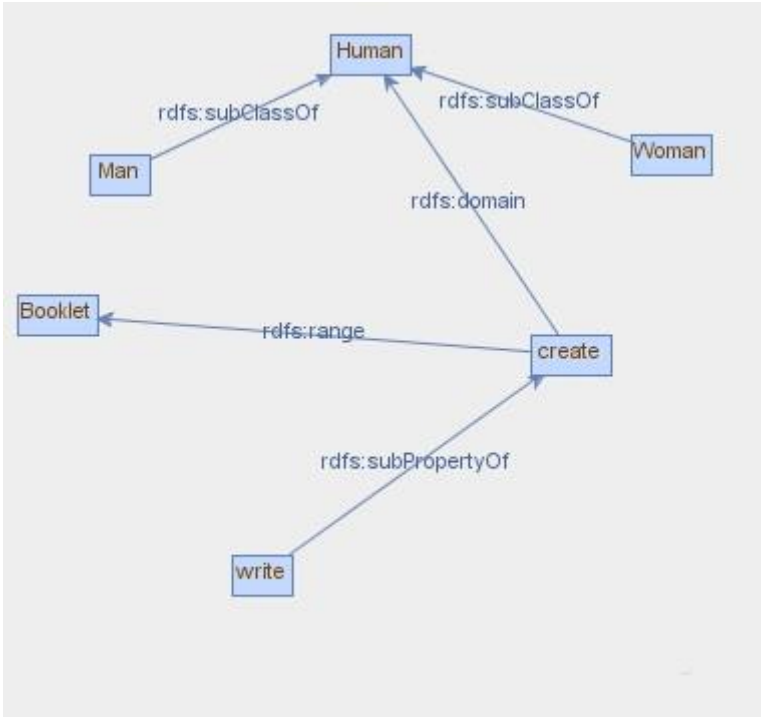

Figure IV.3. Représentation OGraphe de l'ontologie Person2.owl

3.1.2. Ontologie « Hotel »

La paire d'ontologies Hotel, décrit les caractéristiques des chambres d'hôtel. Les deux ontologies sont équivalentes, mais elles décrivent les mêmes concepts de différentes façons. Les deux ontologies de cette paire ont une structure assez différente et les entités sont nommées différemment. Par exemple, la classe OnFloor de la première ontologie correspond à la classe FloorAttribute de la deuxième ontologie.

La Figure IV.4 et IV.5 montrent les représentations des O-Graphes.

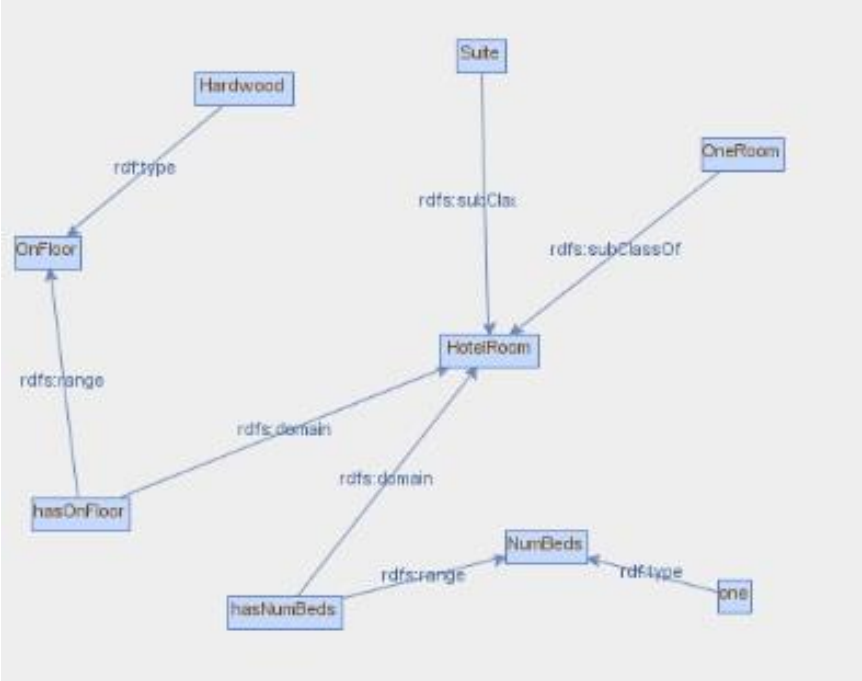

Figure IV.4. Représentation OGraphe de l'ontologies Hotel1.

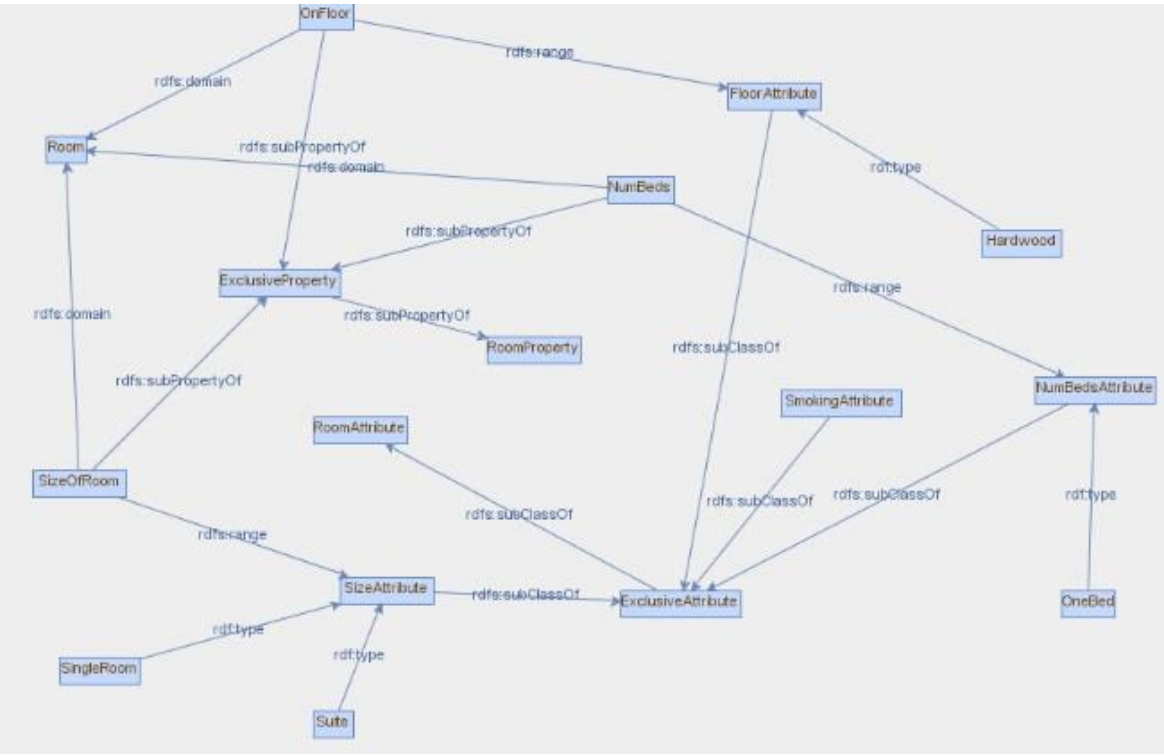

Figure IV.5. Représentation OGraphe de l'ontologies Hotel2.

3.1.3. Ontologie « Network »

La paire d'ontologies Network décrit les noeuds et les connexions dans un réseau local. Network1.owl se concentre davantage sur les noeuds eux-mêmes, tandis que network2.owl est plus englobant des connexions. Les deux ontologies dans cette paire ont une structure similaire et les termes nommant les entités sont aussi similaires (Equipment - Equipment, Cable - Cable, NetworkNode - NetworkNode). Par contre, la structure d'ontologie est assez complexe. Donc, on a ajouté cette paire d'ontologies à nos tests pour avoir une diversité de type d'ontologies pour mieux tester les performances de l'algorithme.

La Figure IV.6 et IV.7 montrent les représentations des O-Graphes comme suite :

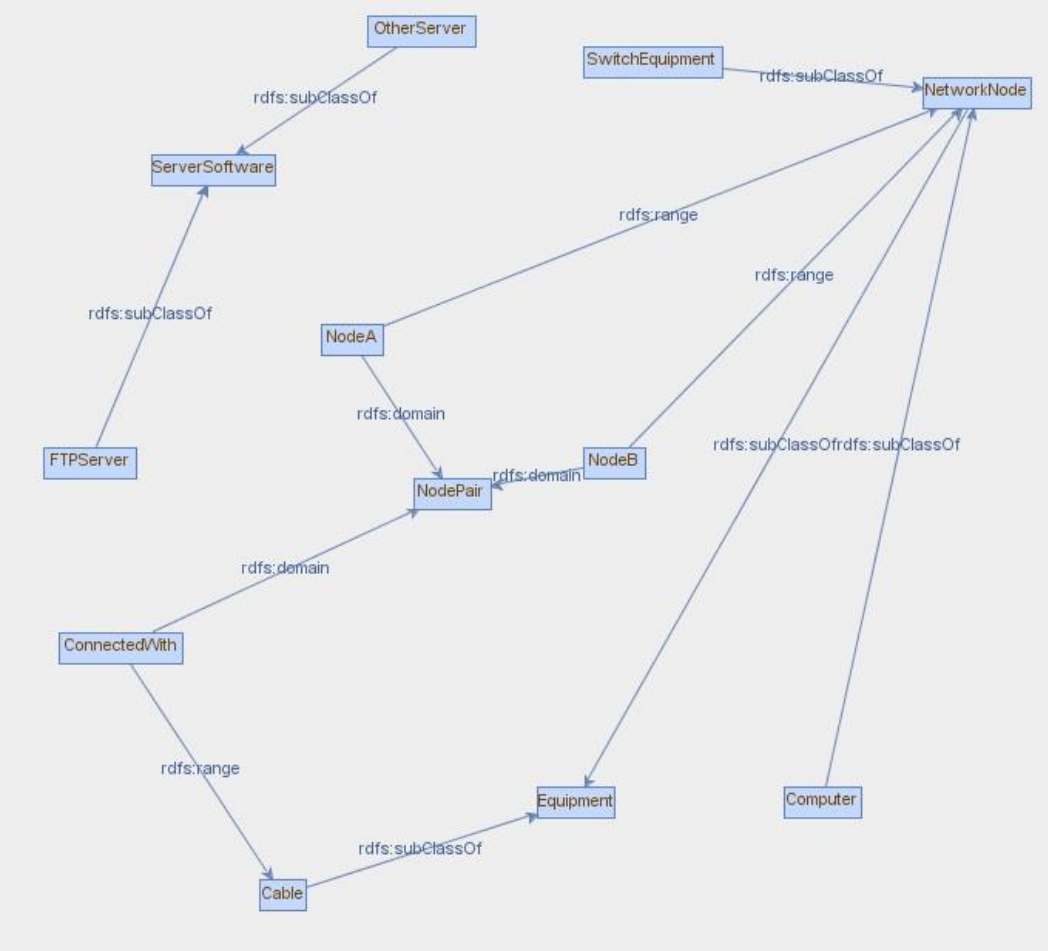

Figure IV.6. Représentation O Graphe de l'ontologies Network1.

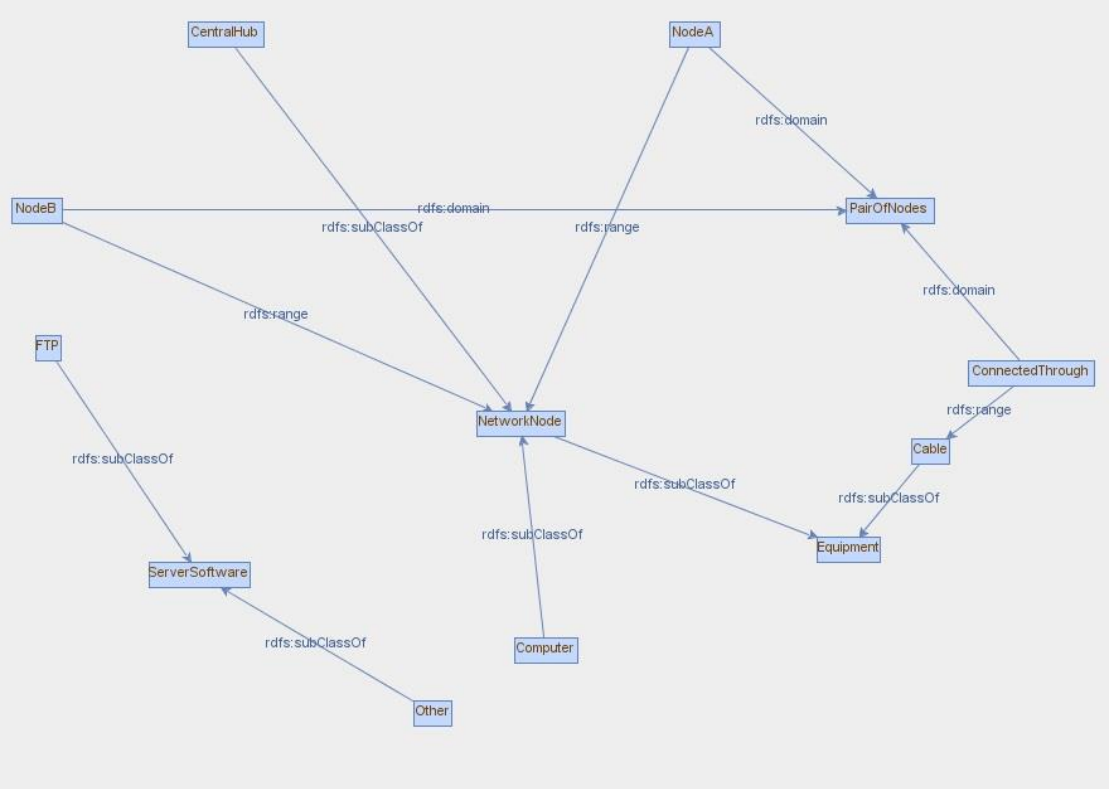

Figure IV.7. Représentation O Graphe de l'ontologies Network2.

4. Implémentation de notre système

Dans cette partie, on va décrire les détails d'implémentation de notre système. Nous commençons d'abord par la description des classes principales :

4.1. Description des classes

- **« ontologya »**

C'est la classe qui permet de normaliser les concepts et calculer la similarité globale entre eux. Les méthodes principales de cette classe sont :

- **« getClassiter » :** La méthode qui permet de lire les deux ontologies à aligner et extracter les classes de ces ontologies et les stocker dans des listes.
- **« properties » :** La méthode qui permet de lire les deux ontologies à aligner et extracter les propriétés de ces ontologies et les stocker dans des listes.
- **« individuals » :** La méthode qui permet de lire les deux ontologies et extracter les individus de ces ontologies et les stocker dans des listes.
- **« concept_normalisation » :** La méthode qui permet de normaliser les concepts des ontologies à aligner.
- **« structural_interne » :** La méthode qui permet de calculer la valeur de similarité structurelle interne entre deux concepts.
- **« structural_externe » :** La méthode qui permet de calculer la valeur de similarité structurelle externe entre deux concepts.
- **« calculate » :** La méthode qui permet de calculer la similarité globale entre deux concepts.

- **« JaroWinkler »**

C'est la classe qui contient la méthode principale pour calculer la similarité terminologique entre deux concepts.

- **« SimilarityCalculationDemo »**

C'est la classe qui contient la méthode principale pour calculer la similarité sémantique entre deux concepts.

4.2. Présentation de notre système

Dans cette partie on va tester notre système que nous avons réaliser pour effectuer l'alignement entre les ontologies du domaine. On va tester le système en faisant des captures d'écran pour montrer les différentes étapes nécessaires pour effectuer cet alignement entre les différentes paires d'ontologies de test que nous avons proposé dans le chapitre précédent.

La présentation de notre système commence par une figure de l'interface d'acceuil de notre application (Figure V.8). Cette interface d'acceuil contient 2 boutons « Importer l'ontologie 1 » et « Importer l'ontologie 2 » qui permettent à l'utilisateur l'importer les deux ontologies à aligner, un bouton « Aligner » pour lancer l'alignement entre les deux ontologies choisies, et un autre bouton « Fermer » pour fermer l'application et annuler l'exécution.

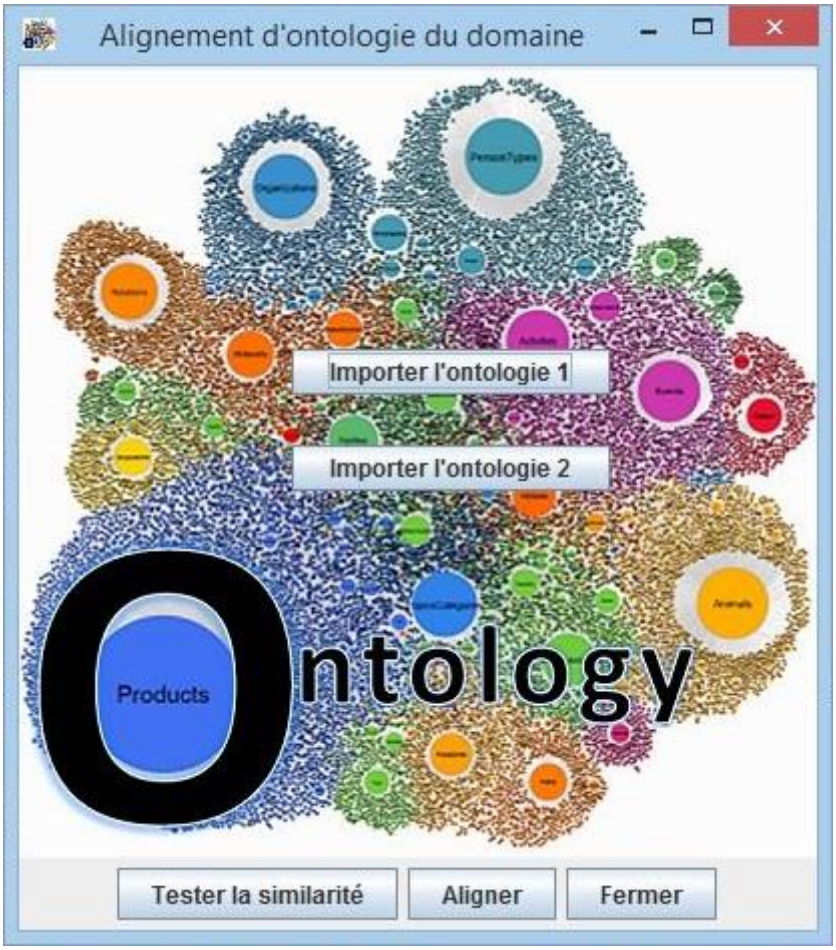

Figure V.8. L'interface d'accueil de l'application « Alignement d'ontologie du domaine ».

En appuyant sur le bouton « Tester la similarité », une fenêtre qui permet de tester la similarité terminologique et sémantique entre deux concept s'affiche (FigureV.9).

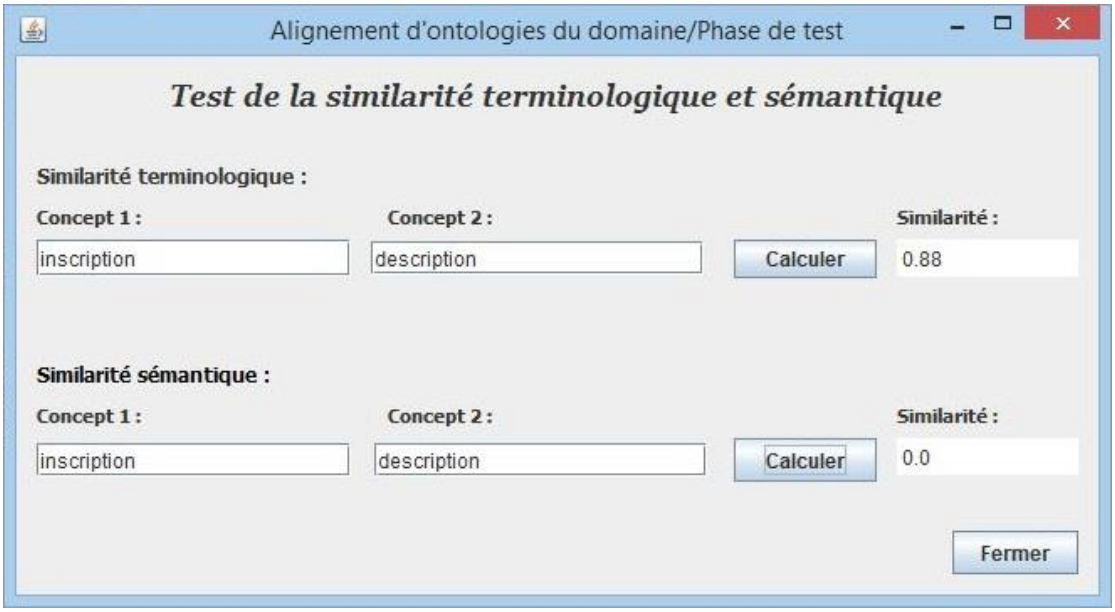

Figure V.9. Une fenêtre de test de la similarité terminologique et sémantique.

En appuyant sur l'un des deux boutons « Importer l'ontologie 1 » et « Importer l'ontologie 2 » une fenêtre s'affiche pour choisir le fichier d'ontologie sous le format owl, et çela grace à un explorateur de fichiers (Figure V.10).

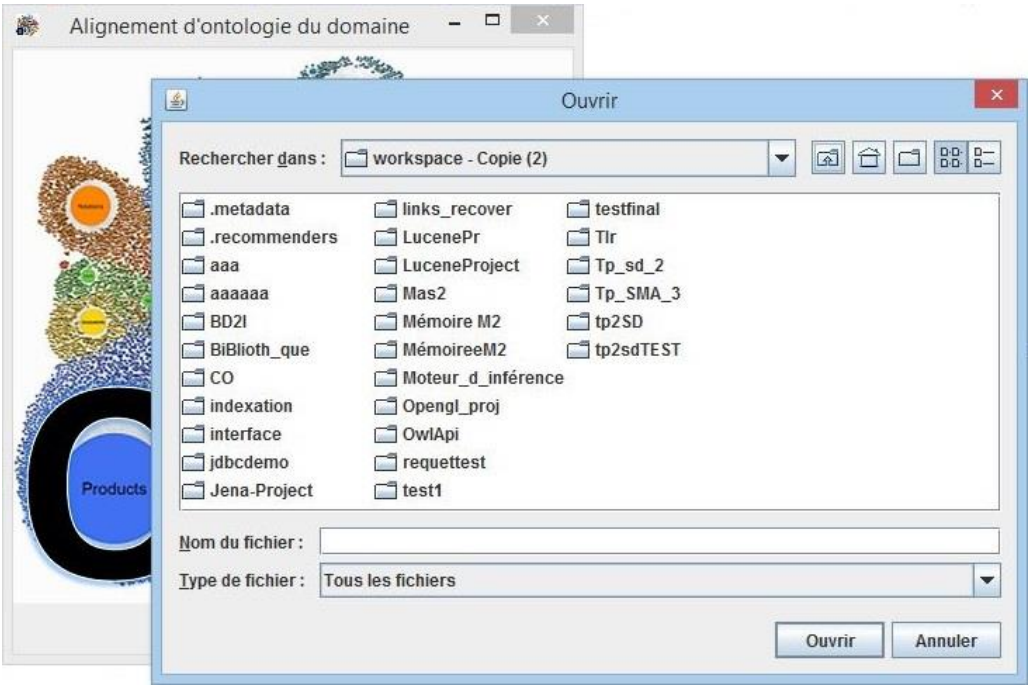

Figure V.10. Une fenêtre de l'explorateur pour choisir le fichier d'ontologie.

Après qu'on charge les deux fichiers d'ontologies à aligner, on appuie sur le bouton «Aligner» pour lancer l'algorithme.

Après que l'algorithme finisse son calcul, une autre fenêtre s'affiche, cette dernière contient un tableau remplis par les concepts des deux ontologies, cette interface est représentée dans la figure (V.11) .

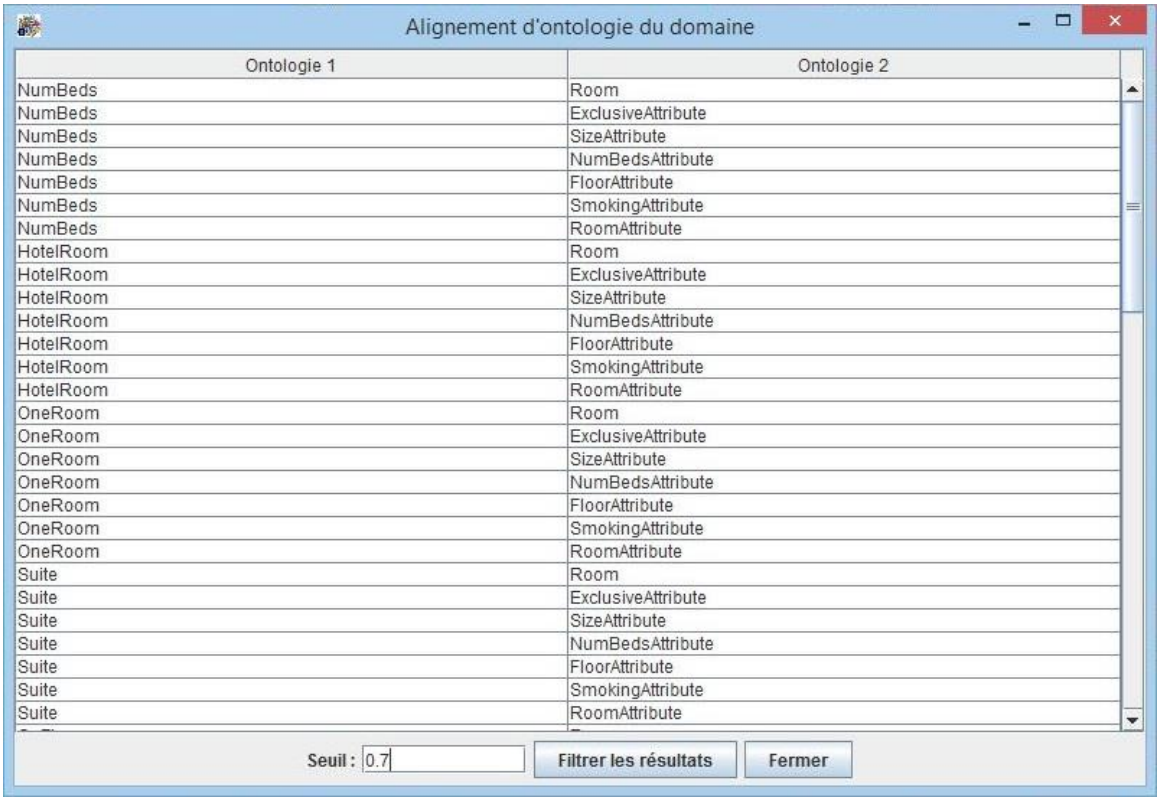

Figure V.11. Fenêtre d'affichage des résultats.

Cette fenêtre contient 2 boutons, « Filtrer les résultats » pour afficher seulement les résultats supérieurs ou égales au seuil entré, « Fermer » pour fermer l'application.

En appuyant sur le bouton « Filtrer les résultats » une autre fenêtre s'affiche (Figure V.12).

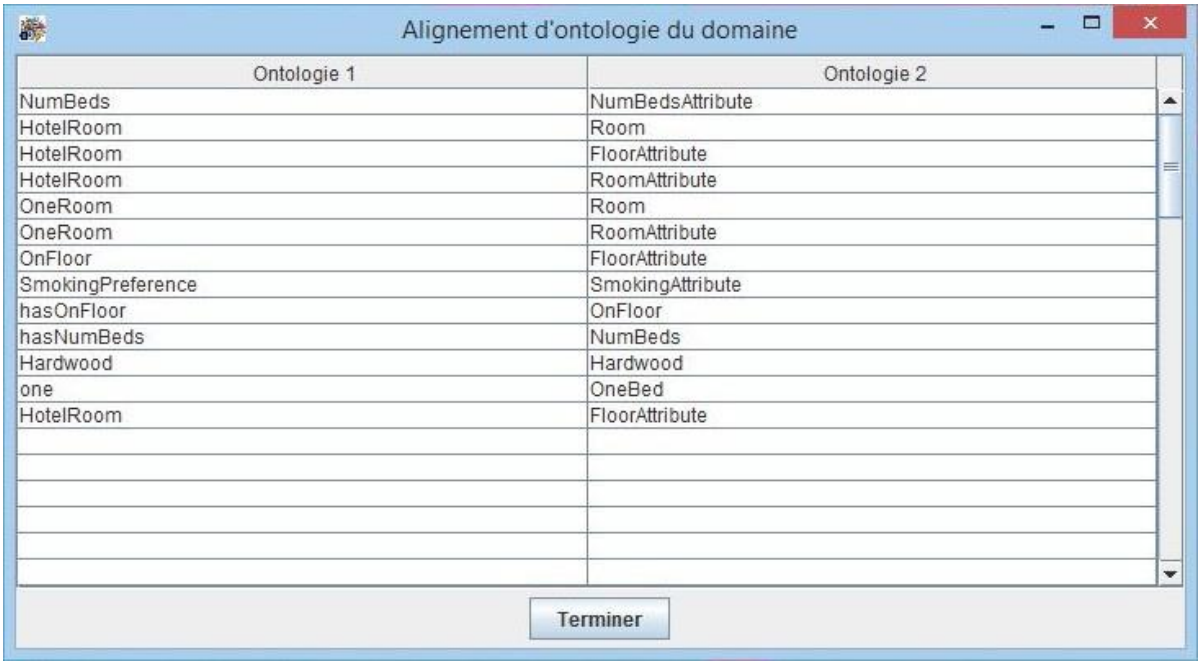

Figure V.12. Fenêtre de filtrage des résultats après le filtrage.

5. Evaluation du système et analyse des résultats expérimentaux

5.1. Mesures d'évaluation

Pour évaluer la précision, l'efficacité et la performance de l'algorithme, nous employons les mesures de précision, de rappel et de f-mesure. Ce sont les mesures bien utilisées dans le domaine de recherche d'information et ensuite appliquées dans le domaine d'alignement d'ontologies pour permettre une analyse fine des performances de système.

Le rappel est la proportion de correspondances correctes renvoyées par l'algorithme parmi toutes celles qui sont correctes (en incluant aussi des correspondances correctes que l'algorithme n'a pas détectées). Le rappel mesure l'efficacité d'un algorithme. Plus la valeur de rappel est élevée, plus le résultat de l'algorithme couvre toutes les correspondances correctes.

La précision est la proportion des correspondances correctes parmi l'ensemble de celles renvoyées par l'algorithme (ce sont les correspondances dans la liste des quadruplet). Cette mesure reflète la précision d'un algorithme. Plus la valeur de précision est élevée, plus le bruit dans le résultat de l'algorithme est réduit, et donc plus la qualité de résultat est imposante.

Enfin, la f-mesure est un compromis entre le rappel et la précision. Elle permet de comparer les performances des algorithmes par une seule mesure. La f-mesure est définie par :

$$f - mesure = \frac{2 * rappel * précision}{rappel + précision}$$

Nous utilisons aussi la mesure globale (overall measure). Cette mesure, appelée l'exactitude, correspond à l'effort exigé pour corriger le résultat renvoyé par l'algorithme afin d'obtenir le résultat correct. La mesure globale est toujours inférieure à la f-mesure. Elle n'a un sens que dans le cas où la précision n'est pas inférieure à 0,5, c'est-à-dire si au moins la moitié des correspondances renvoyées par l'algorithme sont correctes. En effet, si plus d'une moitié des correspondances sont erronées, il faudrait à l'utilisateur plus d'effort d'enlever les correspondances incorrectes et d'ajouter les correspondances correctes mais absentes, que pour mettre en correspondance manuellement les deux ontologies à partir de zéro. [16]

$$overall = rappel * \left(2 - \frac{1}{précision}\right)$$

Nous notons :

- F : le nombre des correspondances renvoyées par l'algorithme.
- T : le nombre des correspondances correctes déterminées manuellement par des experts.
- C : le nombre des correspondances correctes trouvées (appelé aussi vrais positifs).
- I = F - C : nombre des correspondances incorrectes trouvées (appelé aussi faux positifs).
- M = T - C : nombre des correspondances correctes mais pas trouvées (appelé aussi faux négatifs).
- Ainsi, précision = C / F ; rappel = C / T ; f-mesure = 2*P*R/(P+R) et overall = R*(2-1/P).

5.2. Evaluation et analyse des résultats expérimentaux

5.2.1. L'ontologie « network »

La Figure suivante représente un tableau qui contient les résultats de l'alignement trouvés par notre application en les filtrant par un seuil égale à 0.8 parceque ce domaine de réseau a besoin d'une grande précision, donc il nécessite une valeur de seuil elevée :

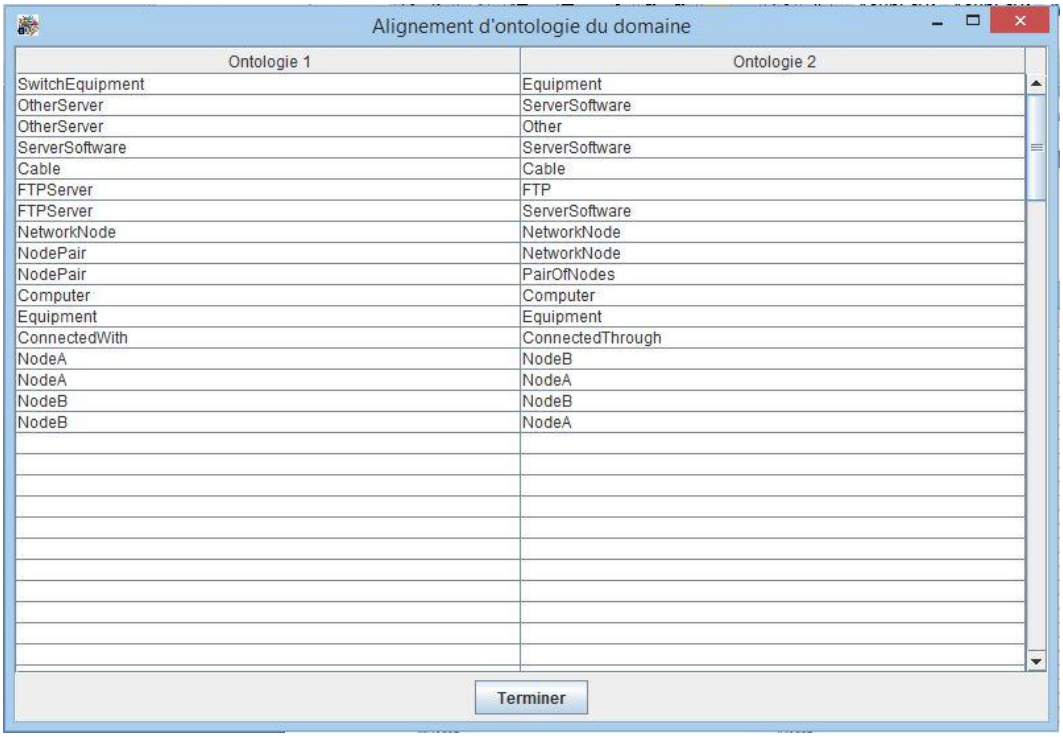

Figure V.13. Les résultats de l'alignement trouvés par notre application dans le domaine « network ».

On a les résultats renvoyés par l'approche ASCO3 pour la même paire d'ontologie « network » représentés dans la figure suivante :

Entité de l'Ontologie A	Entité de l'Ontologie B
NetworkNode	NetworkNode
Equipment	Equipment
NodeB	NodeA
NodePair	PairOfNodes
Computer	Computer
SwitchEquipment	CentralHub
OtherServer	Other
ServerSoftware	ServerSoftware
NodeA	NodeB
FTPServer	FTP
ConnectedWith	ConnectedThrough
Cable	Cable

FigureV.14. Les résultats renvoyés par l'approche ASCO3 pour la paire d'ontologie « network ».

On a aussi les correspondances correctes déterminées manuellement par des experts indiquées dans le tableau suivant :

Ontologie networkA	Ontologie networkB
Equipment	Equipment
Cable	Cable
NetworkNode	NetworkNode
Computer	Computer
SwitchEquipment	CentralHub
ServerSoftware	ServerSoftware
FTPServer	FTP
OtherServer	Other
NodePair	PairOfNodes
ConnectedWith	ConnectedThrough
NodeA	NodeA
NodeA	NodeB
NodeB	NodeA
NodeB	NodeB

Figure V.15. La liste des correspondances correctes pour l'ontologie "Network".

En analysant les résultats obtenus par notre application, on trouve que :

Seuil	F	T	C	I=F-C	M=T-C	Précision P=C/F	Rappel R=C/T	F-Mesure FM=2*P*R/(P+R)	Overall O=R*(2-1/P)
0.5	24	14	13	11	1	0.541	0.928	0.683	0.141
0.55	18	14	13	5	1	0.720	0.928	0.810	0.567
0.6	17	14	13	4	1	0.764	0.928	0.837	0.642
0.65	17	14	13	4	1	0.764	0.938	0.837	0.642
0.7	17	14	13	4	1	0.764	0.938	0.837	0.642
0.75	17	14	13	4	1	0.764	0.938	0.837	0.642
0.8	17	14	13	4	1	0.764	0.938	0.837	0.642
0.85	17	14	13	4	1	0.764	0.938	0.837	0.642
0.9	17	14	13	4	1	0.764	0.938	0.837	0.642

Tableau V.1. Analyse des résultats de l'alignement pour la paire d'ontologies « network ».

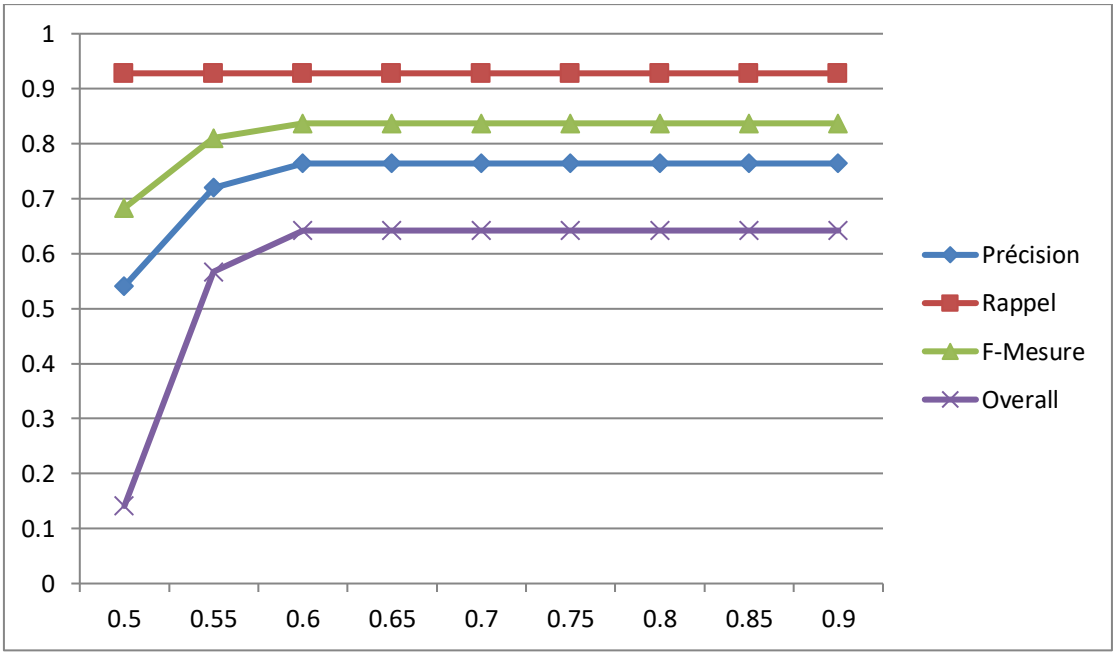

Figure V.16. Représentation graphique des résultats obtenus pour la paire d'ontologies « network ».

En comparant les résultats renvoyés par notre application avec les résultats de correspondance correctes déterminées manuellement par des experts, et les résultats renvoyés par l'approche ASCO3, on trouve que notre application a renvoyés presque tous les correspndances correctes déterminées manuellement par des experts, plus d'autres correspondances qui sont vraies ou proches.

On remarque que les résultats renvoyés par notre application et les résultats renvoyés par l'approche ASCO3 sont convergents pour la même paire d'ontologie « network ».

5.2.2. L'ontologie « hotel »

La Figure suivante représente un tableau qui contient les résultats de l'alignement trouvés par notre application en les filtrant par un seuil égale à 0.7:

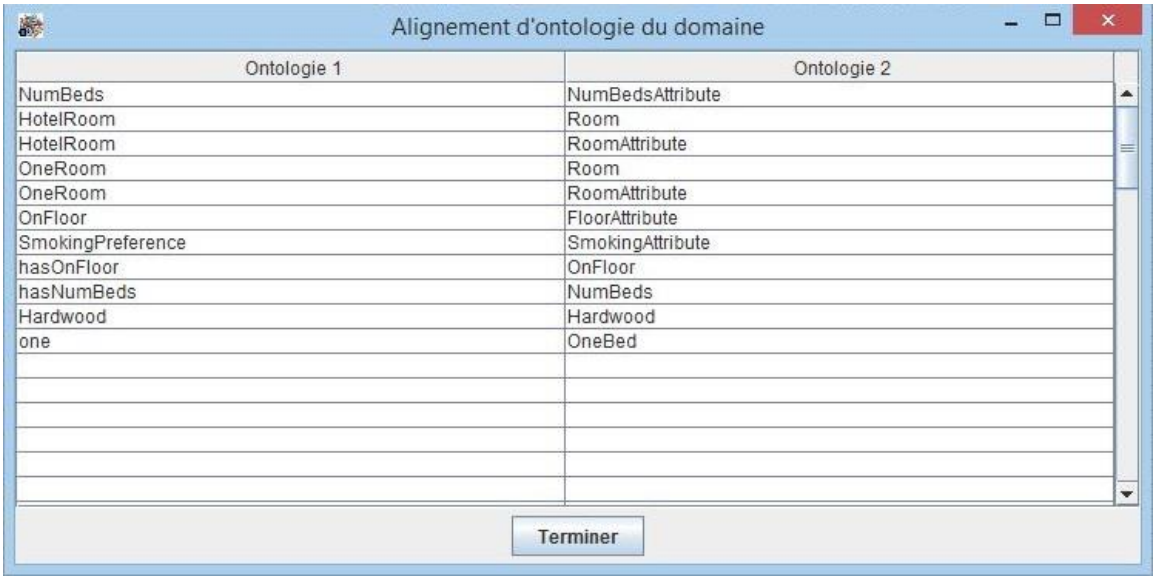

Figure V.17. Les résultats de l'alignement trouvés par notre application dans le domaine « hotel ».

On a les résultats renvoyés par l'approche ASCO3 pour la même paire d'ontologie « hotel » représentés dans la figure suivante :

Entité de l'Ontologie A	Entité de l'Ontologie B
hasNumBeds	NumBeds
NumBeds	NumBedsAttribute
OneRoom	Room
Suite	RoomAttribute
Hardwood	Hardwood
OnFloor	FloorAttribute
one	OneBed
hasOnFloor	OnFloor

Figure V.18. Les résultats renvoyés par l'approche ASCO3 pour la paire d'ontologie « hotel ».

On a aussi les correspondances correctes déterminées manuellement par des experts indiquées dans le tableau suivant :

Ontologie HotelA	Ontologie HotelB
HotelRoom	Room
OneRoom	SingleRoom
Suite	Suite
NumBeds	NumBedsAttribute
SmokingPreference	SmokingAttribute
OnFloor	FloorAttribute
hasNumBeds	NumBeds
hasOnFloor	OnFloor
One	OneBed
Hardwood	Hardwood

Figure V.19. La liste des correspondances correctes pour l'ontologie "hotel".

En analysant les résultats obtenus par notre application, on trouve que :

Seuil	**F**	**T**	**C**	**I=F-C**	**M=T-C**	**Précision P=C/F**	**Rappel R=C/T**	**F-Mesure FM=2*P*R/(P+R)**	**Overall O=R*(2-1/P)**
0.4	18	10	8	10	2	0.44	0.8	0.567	-0.217
0.45	14	10	8	6	2	0.571	0.8	0.665	0.199
0.5	11	10	8	3	2	0.727	0.8	0.761	0.5
0.55	11	10	8	3	2	0.727	0.8	0.761	0.5
0.6	11	10	8	3	2	0.727	0.8	0.761	0.5
0.65	11	10	8	3	2	0.727	0.8	0.761	0.5
0.7	11	10	8	3	2	0.727	0.8	0.761	0.5

Tableau V.2. Analyse des résultats de l'alignement pour la paire d'ontologies « hotel ».

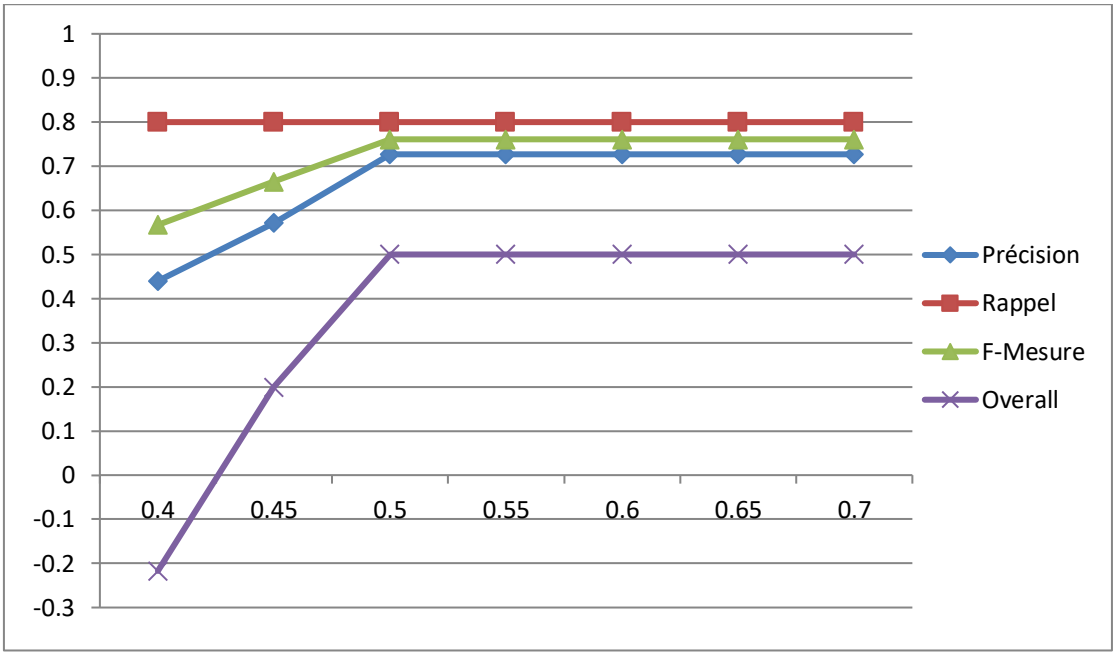

Figure V.20. Représentation graphique des résultats obtenus pour la paire d'ontologies « hotel ».

En comparant les résultats renvoyés par notre application avec les résultats de correspondance correctes déterminées manuellement par des experts, et les résultats renvoyés par l'approche ASCO3, on trouve que notre application a renvoyés tous les correspndances correctes déterminées manuellement par des experts sauf 2, plus d'autres correspondances qui sont vraies ou proches.

On remarque que les résultats renvoyés par notre application sont meilleurs que les résultats renvoyés par l'approche ASCO3 pour la même paire d'ontologie « hotel ».

5.2.3. L'ontologie « Person »

La Figure suivante représente un tableau qui contient les résultats de l'alignement trouvés par notre application en les filtrant par un seuil égale à 0.65 :

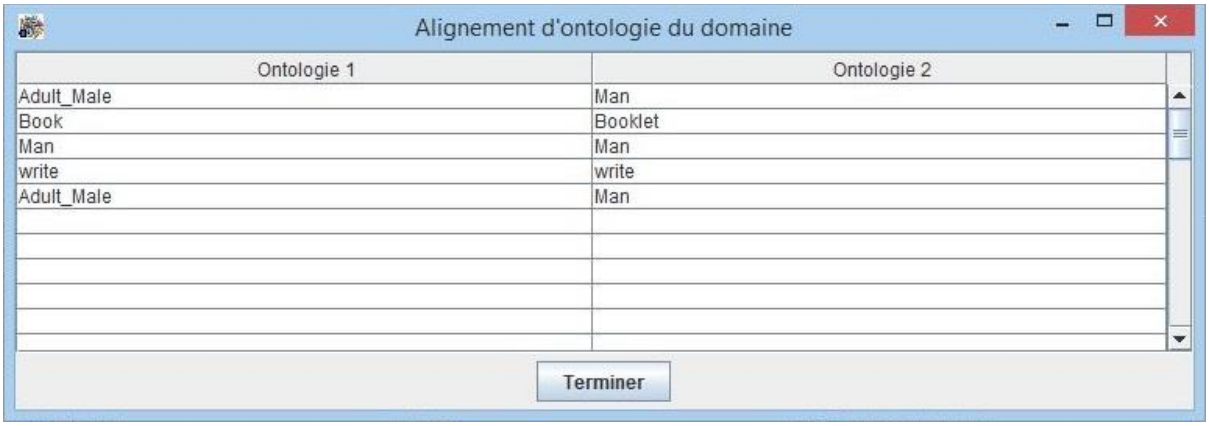

Figure V.21. Les résultats de l'alignement trouvés par notre application dans le domaine « person ».

On a les résultats renvoyés par l'approche ASCO3 pour la même paire d'ontologie « person » représentés dans la figure suivante :

Entité de l'Ontologie A	Entité de l'Ontologie B
Magazine	Booklet
write	create
Person	Human
Adult	Woman
Adult_Male	Man

FigureV.22. Les résultats renvoyés par l'approche ASCO3 pour la paire d'ontologie « person ».

Seuil	**F**	**C**	**I=F-C**	**Précision P=C/F**
0.5	18	9	9	0.5
0.55	13	7	6	0.538
0.6	13	7	6	0.538
0.65	5	5	0	1
0.7	3	3	0	1
0.75	2	2	0	1
0.8	2	2	0	1

Tableau V.3. Analyse des résultats de l'alignement pour la paire d'ontologies « person ».

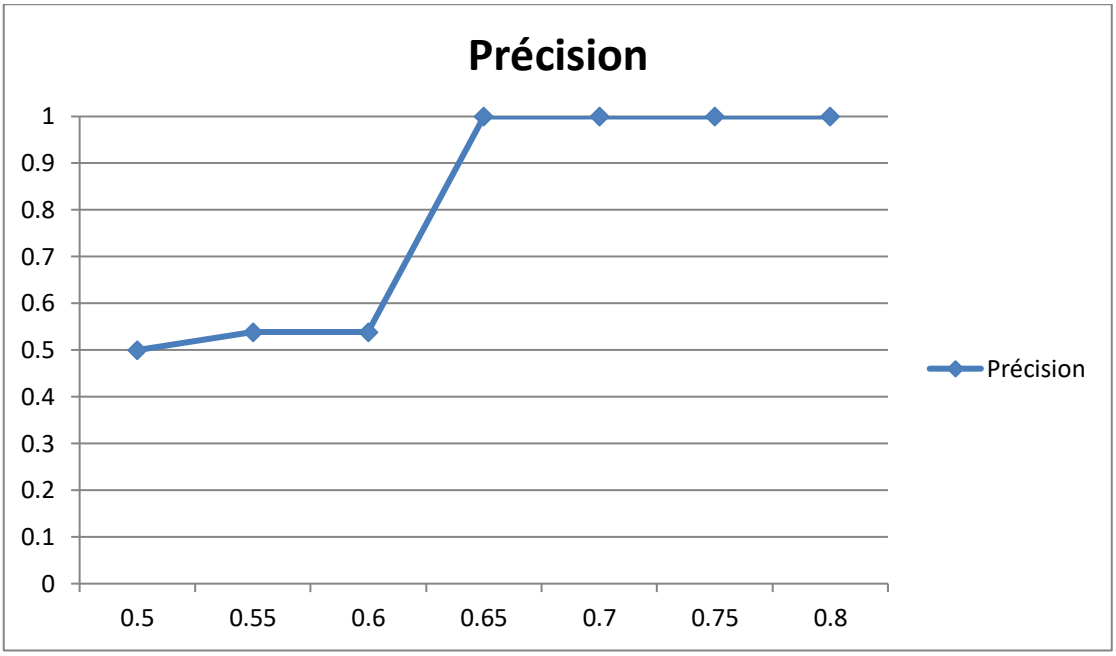

Figure V.23. Représentation graphique des résultats obtenus pour la paire d'ontologies « person ».

En comparant les résultats renvoyés par notre application avec les résultats renvoyés par l'approche ASCO3, on trouve que notre application a renvoyés des correspondances qui sont toutes correctes, donc la précision de notre algorithme dans cette paire d'ontologies égale à 1. La même chose pour les résultats renvoyés par l'approche ASCO3, la précision de l'approche ASCO3 égale à 1.

On remarque que notre algorithme et l'algorithme de l'approche ASCO3 sont équivalents pour la même paire d'ontologies « person ».

6. Conclusion

Dans ce chapitre, on a fait une petite représentation sur notre application, on a vu les outils qu'on a utilisés pour l'implémentation de notre système, qui sont le vocabulaire Wordnet qui nous a permis de trouver tout les synonymes d'un mot ou d'un terme en anglais, le langage JAVA qu'on a utilisé comme un langage de programmation. Ensuite l'environnement Eclipse qu'on considère comme une plate forme de programmation, et protégé pour l'édition d'ontologie.

Pour les ontologies du test , on n'avait pas beaucoup de choix, vu la complexité de l'algorithme et le temps d'exécution. On a utilisé que des petites ontologies, qui ont environ des dizaines d'entités, afin de pouvoir les aligner.

CONCLUSION GENERALE

Dans ce projet, nous avons traité un sujet très connu dans le cadre de l'alignement d'ontologies, qui est l'alignement d'ontologies de domaine. Au cours de ce traitement, nous avons commencé par le premier chapitre qui contient la définition de la notion d'ontologie , son rôle ,ses composants, ses différents types, son cycle de vie, ses langages de description et de représentation, et ses domaines d'application. Ensuite dans le deuxième chapitre, nous avons défini la notion de l'alignement, la naissance du besoin à ce dernier, ses différentes méthodes et ses domaines d'application. Dans le troisième chapitre, nous avons cité les différentes approches existantes dans le domaine de l'alignement d'ontologies avec une synthèse comparative entre ces différentes approches citées. Enfin , nous avons présenté la conception et l'implémentation de notre application, dans ces parties nous avons présenté les différentes méthodes principales proposées pour arriver au but d'aligner les ontologies, en expliquant le déroulement et le fonctionnement de notre algorithme proposé et implémenté. Nous avons aussi mis l'évaluation de notre système implémenté et l'analyse des résultats obtenus par ce système dans le dernier chapitre.

L'objectif de notre travail était de réaliser un outil informatique concernant le domaine d'alignement d'ontologies et l'imlémentation d'un algorithme qui a pour but d'aligner des ontologies et chercher les correspondances entre deux ontologies représentées en OWL (le langage d'ontologie recommandé par W3C pour le Web sémantique). Cet algorithme repose sur les avantages des méthodes de similarité terminologique, sémantique et structurelle pour déduire la similarité de deux entités dans les ontologies.

Finalement, bien qu'on constate qu'il y a plusieurs travaux et recherches qui traitent ce sujet, le problème d'alignement d'ontologies est loin d'être résolu à 100%, il reste beaucoup encore de travaux à faire dans ce domaine.

PERSPECTIVES

Comme perspective pour cette application, on doit essayer de trouver des solutions pour les limites qui face cet algorithme et son déroulement, afin de l'améliorer. Surtout pour le problème du temps d'exécution de l'algorithme, on doit trouver une manière pour

minimiser le temps de la recherche. Ce qui va nous permettre d'exécuter cet algorithme sur des ontologies de tailles normales avec des résultats meilleurs.

Bibliographie

[1] : dspace.univ-tlemcen.dz/bitstream/112/1062/5/ChapitreI.pdf

[2] : Abdeltif Elbyed. « ROMIE, une approche d'alignement d'ontologies à base d'instances ». Autre [cs.OH]. Institut National des Télécommunications, 2009. Français. <NNT:2009TELE0014>. <tel-00541874>

[3] : dspace.univ-tlemcen.dz/bitstream/112/1062/9/Annexe-Protege.pdf

[4] : Moharned TOUZANI. « Alignement des ontologies OWL-Lite ». Département d'informatique et de recherche opérationnelle, Faculté des arts et des sciences, Université de Montréal , 2005.

[5] : Mansouri Kelthom, Laoufi Noura. « Alignement d'ontologie ». Faculté des Sciences et de la Technologie, Département de Mathématiques et d'Informatique, Université de Djilali BOUNAAMA Khemis Miliana, 2018.

[6] : ARDJANI Fatima. « Evolution des Données Liées :Maintenance des liens ». Faculté des Sciences Exactes, Département d'informatique, Université Djillali Liabès de Sidi Bel Abbès, 2017.

[7] : Samer Abdul Ghafour. « Méthodes et outils Pour l'intégration des ontologies ». Ecole Doctorale « Informatique et Information pour la Société » EDA 335, Laboratoire d'Informatique en Images et Systèmes d'information, LIRIS FRE2672 CNRS, INSA Lyon, UCB Lyon 1, EC Lyon, 2004.

[8] : HACINE GHERBI Ahcine. « Construction d'une Ontologie pour le WEB Sémantique ». École doctorale d'informatique, UNIVERSITÉ FERHAT ABBES – SETIF1-, 2014.

[9] : Baroudi Abderrazak. « Application de recherche d'emploi ». Faculté desSciences, Département d'Informatique, Université Abou Bakr Belkaid– Tlemcen, 2015.

[10] : Zohra Bellahsene, « Rôle et techniques de l'alignement d'ontologies : un survol de l'état de l'art » , INFORSID 2017.

[11] : http://thesis.univ-biskra.dz/843/1/Inf_m1_2011, pdf.

[12] : Yassine NAMIR, Achraf RAGUI. « Alignement d'ontologies du web sémantique ». Faculté des sciences et techniques FES, Département d'informatique, Université Sidi Mohamed Ben Abdellah, 2016.

[13] : Warith-Eddine DJEDDI. « Alignementsémantique des ontologies de grande Taille ». Faculté des sciences de l'ingénieur, Département d'informatique, Université Badji Mokhtar –Annaba-, 2013.

[14] : Soumaya Kasri. « Etude des techniques d'alignement des ontologies: proposition d'un algorithme de passage à l'échelle ». Ecole doctorale de l'informatique de l'Est, (Université 20 Août 1955 Skikda), 2010.

[15] : Sami Zghal. « Contributions à l'alignement d'ontologies OWL par agrégation de Similarités ». Université d'ARTOIS, 2010.

[16] : Lê Bach Thanh. « Construction d'un Web sémantique multi-points de vue ». domain_other. École Nationale Supérieure des Mines de Paris, 2006. English. <pastel-00001989>.

[17] : Sami Zghal, Sadok Ben Yahia, Engelbert Mephu Nguifo, Yahya Slimani. « SODA : 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.