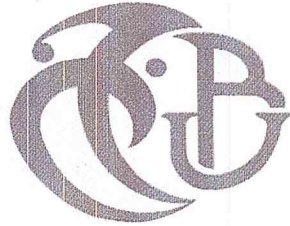
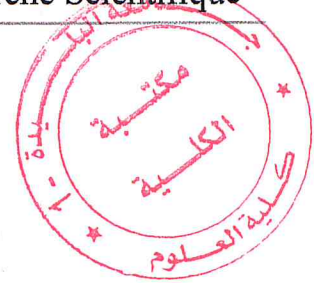


République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Saad Dahlab Blida

N° D'ordre :.....



Faculté des sciences

Département d'informatique

Mémoire Présenté par :

EL FARROUDJI Chaima GUELLATI Hadjer

En vue d'obtenir le diplôme de master

Domaine : Mathématique et informatique

Filière : Informatique

Spécialité : Informatique

Option : Ingénierie de logiciel

**Thème : Alignement des ontologies en utilisant la technique de réutilisation
des mapping**

Soutenu le :

Mme Boustia Narimane	Président
Mme Chikhi Emene	Examineur
M.	Examineur
Mme. Farih Messouda	Promotrice

Promotion
2014 / 2015

Remerciement

Avant de présenter notre travail, il est de notre devoir d'exprimer nos remerciements tout d'abord à Allah tout-puissant qui nous a donnés la force pour réaliser ce travail et puis nos parents qui nous ont été un grand soutien pendant toute la période de nos études et dès le début de ce travail.

Nous somme profondément reconnaissant à notre promotrice, Madame Fareh pour le temps qu'elle a investi pour suivre et analyser les résultats de ce travail ainsi que pour ses conseils qu'elle n'a cessé de nous prodiguer.

Un énorme remerciement à nos parents, nos sœurs et frères pour leurs encouragements.

Dédicace

Le parcours d'une vie est jalonné d'opportunités qui dépendent de nous, mais également à des personnes qu'il nous a été donné de rencontrer: des personnes qui nous guident, qui nous conseillent et qui nous font confiance, j'ai eu de la chance de rencontrer quelques-uns d'entre eux et à qui je tiens à dédier ce mémoire :

Mes chers et adorables Parents: pour votre amour, votre affection et votre sacrifice consentis, pour votre soutien indéfectible, bienveillance et Vos encouragements, Que Dieu bienveillant vous garde à mon côté

Ma chère sœur et mon cher frère : Salima et Abderrahmane pour vos encouragements et amour inconditionnel.

Ma douce grand-mère : Wahiba pour ses encouragements et son amour.

À mon amie et mon binôme Chaïma qui m'a accompagné dans tout le long de ce travail, pour son courage et sont sérieux, ainsi que sa gentillesse.

À mes chère amies : Ahlem, Rahma, Kahina, Asma, Imene...

À ma très chère cousine et âme sœur : Meriem

À tous mes professeurs pour l'effort et le dévouement pendant tout mon parcours d'études.

À tous ceux que j'aime.

Hadjer Guellati



Dédicace

Avec un énorme plaisir, un cœur ouvert et une immense joie, que je dédie mon travail à mes très chers, respectueux et magnifiques parents qui m'ont soutenu tout au long de ma vie ainsi à mes sœurs et mes frères El Roumaïssa, Zakaria Abd el Rakib, Oumama et Ahmed Abd El Mocketadîr et à toute ma famille.

et en particulier à mon binôme Hadjer et mes amis Kahina ,Naziha et Asma.

A toutes personnes qui m'ont encouragé ou aidé au long de mes études.

Chaima El Farroudji

Résumé

L'alignement des ontologies est une solution pour résoudre l'hétérogénéité des données, que ce soit au niveau syntaxique ou sémantique. En effet, des conflits sémantiques surviennent puisque les systèmes n'utilisent pas la même interprétation de l'information qui est définie différemment d'une organisation à l'autre en proposant des correspondances entre des entités sémantiques similaires de différentes ontologies. L'alignement des ontologies a pour but de réduire cette hétérogénéité.

Afin d'identifier des mapping entre les concepts de deux ontologies, de nombreux travaux récents portent sur l'utilisation de connaissances complémentaires dites de "background" ou de support. Le mapping est donc une approche intéressante parce qu'il ne modifie pas les ontologies.

Dans ce mémoire, nous proposons une architecture d'alignement d'ontologie en OWL, décrivant le même domaine, qui combine les techniques de base de l'alignement afin d'avoir un alignement pertinent, en utilisant la technique de réutilisation de mapping d'ontologie.

Mots clé : Mapping, Alignement, OWL, Réutilisation, Ontologie.

Abstract

The ontology alignment is a solution to resolve the data heterogeneity, whether it is on a syntactic or a semantic level. In fact semantic conflicts occur since the systems don't use the same interpretation of the information that is defined differently from an organisation to another by proposing matching between similar semantic Entities of different ontologies. The purpose of the ontology alignment is to reduce this heterogeneity.

To identify the mapping between two ontology concepts, numbers of recent studies are based on the use of additional knowledge named "background" or "support". So the mapping is an interesting approach because it doesn't change the ontologies.

In this memoire, we propose an architecture of ontology alignment with OWL, describing the same domains, which combine the basic alignment techniques in order to have a relevant alignment, by using the reusing of ontology mapping technique

ملخص

يعتبر تنسيق الانتولوجيات حلاً لمشكل عدم تجانس البيانات سواء على المستوى النحوي أو الدلالي. فهي الواقع، تنشأ التناقضات الدلالية لأن الأنظمة لا تستخدم تفسيراً موحداً للمعلومات. حيث يتم تعريف المعلومات بشكل مختلف من منظمة إلى أخرى من خلال تقديم توافقات مختلفة بين عناصر نحوية متماثلة. إن هدف تنسيق الانتولوجيات هو الإنقاذ من هذه التوافقات الغير متجانسة.

أهمية مؤخرًا عدة أعمال لإيجاد توافقات بين مفاهيم التولوجيات قائمة على استخدام معارف إضافية لدعم النتيجة. حيث يعتبر التخطيط منهجية مهمة لأنه لا يغير من الانتولوجيات.

من خلال مذكرة التخرج هذه، نقترح هندسة لتنسيق انتولوجيات تعالج نفس المجال. و هذا بدمج تشكيلة من التقنيات الأساسية لتحسين نوعية التنسيق، و باستخدام تقنية إعادة استعمال التخطيط.

Sommaire

Liste des abréviations

Liste des figures

Liste des tableaux

Introduction générale

1.Contexte de travail	1
2.Problématique.....	1
3.Objectif.....	2



Chapitre I : les Ontologies

1. Introduction	5
2. Historique	5
3. Définition.....	6
4. Rôle des ontologies.....	6
5. Les constituants d'une ontologie	7
5.1.Les classes/ les concepts	7
5.2.Les relations et les fonctions	8
5.3.Les axiomes	8
6. Le cycle de vie des ontologies.....	8
7. Typologie des ontologies.....	9
8. Domaine d'application des ontologies	11
8.1.Système d'information	11
9. Représentation des ontologies	11
10. Conclusion.....	13

Chapitre II : État de l'art sur les techniques de mapping d'ontologies

1. Introduction	15
-----------------------	----

2. Terminologies	15
2.1. Mapping ou Correspondances d'ontologies	15
2.2. Appariement ou Matching	16
2.3. Les méthodes de comparaison ou matchers	16
2.4. L'alignement d'ontologie	16
3. Classification des techniques de matching d'ontologies	16
3.1. La classification des techniques de matching selon Euznat et Schvaiko	16
3.1.1. Couche 1	18
3.1.2. Couche 2	19
3.1.3. Couche 3	23
3.2. La classification des techniques de matching selon Ehrig	26
3.3. La classification des techniques de matching selon Zanobini.....	26
4. Composition des techniques de mapping	27
4.1. L'approche hybride	27
4.2. L'approche composite	27
4.3. Comparaison entre l'approche hybride et l'approche composite.....	29
5. Analyse comparative entre quelques systèmes existants de mapping	29
5.1. Quelques systèmes d'alignement d'ontologies	30
5.2. Comparaison des systèmes d'alignement présenté (II.5.1) selon les techniques de mapping de base utilisées	34
5.3. Comparaison des caractéristiques externes des systèmes d'alignement présenté (II.5.1).....	35
6. Conclusion	36
 Chapitre III : Conception du système de mapping d'ontologies	
1. Introduction	38
2. Cycle de vie d'un logiciel.....	38
3. Spécification initial des besoins	39

3.1. Les diagrammes de cas d'utilisation.....	39
4.Analyse des besoins	43
4.1. Les diagrammes de séquences et d'activités	44
5.Conception	47
5.1. Conception générale	47
5.1.1. Structures modulaires du système	48
5.1.2. Diagramme des paquetages.....	49
5.1.3. Architecture détaillé du système	52
5.2. Conception détaillée	54
5.2.1. Diagramme de Classe.....	54
5.2.2. Calcule de la similarité.....	57
5.2.3. Normalisation des chaines de caractères.....	60
5.2.4. Composition des différentes stratégies de calcul de similarité	60
5.2.5. Comparaison entre notre système et les systèmes présentés (II.5.1) selon les techniques d'alignement de base utilisé.....	64
5.2.6. Comparaison entre notre système et les systèmes présentés dans (II.5.1) selon quelques caractéristiques externe.....	65
6. Conclusion.....	66

Chapitre IV : Implémentation du système

1. Introduction	68
2. Outils de développement de notre système	68
3. La mise en œuvre du système.....	69
3.1.L'authentification	70
3.3.Les interfaces de l'administrateur	71
3.4.Les interfaces de l'utilisateur	75
4. Conclusion.....	81

Chapitre V : Test et Validation

1. Introduction	83
2. L'évaluation du système	83
3. Qualité d'alignement d'ontologies	85
4. Tests.....	86
5. Validation	93
6. Conclusion	93
Conclusion générale	
Conclusion générale	94

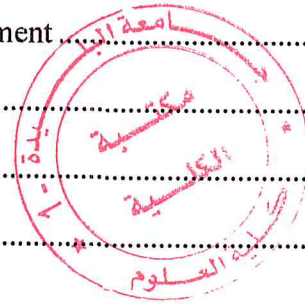
Liste des abréviations

OWL	Ontology Web Language.
URI	Universal Resource Identifier.
XML	eXtensible Markup Language.
W3C	World WideWeb Consortium.
RDF	Resource Description Framework.
RDFS	Resource Description Framework Schema.
SPARQL	SPARQL Protocol and RDF Query Language.
RIF	Rule Interchange Format.
OIL	Ontology Interchange Language and Ontology Inference Layer.
DAML+OIL	DARPA Agent Markup Language.

Liste des figures

Figure 1: Le triangle sémantique (Ogden et Richards, 1923).	8
Figure 2: Cycle de vie d'une ontologie	9
Figure 3: La pyramide des langages d'ontologies basés Web.	12
Figure 4: Mapping d'ontologies	15
Figure 5: Alignement d'ontologies	16
Figure 7: Exemple sur la technique basée sur les graphes	22
Figure 8: Composition parallèle des matchers	28
Figure 9: Composition séquentielle des matchers	28
Figure 10: Cycle de vie de notre système en « cascade »	39
Figure 11: Diagramme de cas d'utilisation global	40
Figure 12: Diagramme de cas d'utilisation d'administrateur « Applications »	42
Figure 13: Diagramme de cas d'utilisation pour l'utilisateur « Applications »	43
Figure 14: Diagramme de séquence du processus d'alignement d'ontologie	45
Figure 15: Diagramme d'activité du processus d'alignement d'ontologies	46
Figure 16: Architecture globale du système	47
Figure 17: Schéma XML du résultat d'alignement	48
Figure 18: diagramme de paquetages	51
Figure 19: Architecture du système d'alignement en utilisant la réutilisation du mapping.	52
Figure 20: Diagramme de classe	54
Figure 21: Exemple d'un extrait d'ontologie	57
Figure 22: La combinaison des différentes stratégies.	61
Figure 23: Interface « Authentification »	70
Figure 24: Interface « inscription »	70
Figure 25: Interface « Administrateur »	71
Figure 26: Interface « Répertoire de Mapping »	72

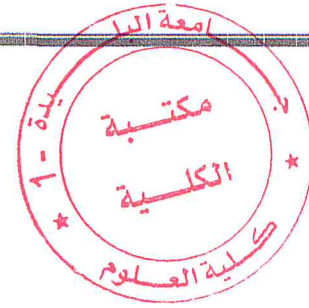
Figure 27: Interface « Gestion d'utilisateur »	73
Figure 28: Interface « Applications »	74
Figure 29: Interface « Utilisateur »	75
Figure 30: Interface « Alignement »	76
Figure 31: Interface du résultat d'alignement	77
Figure 32: Interface de l'exportation du résultat d'alignement	78
Figure 33: Le fichier exporté « RésultatXML»	79
Figure 34: Interface « Chercher un Concept »	80
Figure 35: Interface Voir Taxonomie	80
Figure 36: Résultat de Voir une Taxonomie	81



Liste des tableaux

Tableau 1: Comparaison d'approches hybrides vs. Approches composites	29
Tableau 2: Comparaison de quelques systèmes d'alignement selon les techniques de mapping de base utilisées	34
Tableau 3: Comparaison des systèmes présenté selon leurs exigences les relations et les cardinalités.	35
Tableau 4: Comparaison entre notre système et les systèmes présentés (II.5.1) selon les techniques d'alignement de base utilisées	64
Tableau 5: Comparaison entre notre système et les systèmes présentés dans (II.5.1) selon quelques caractéristiques externe	65
Tableau 6: La comparaison des correspondances trouvé manuellement et celle trouvé avec notre système d'alignement de deux ontologies sans la réutilisation de mapping.	89
Tableau 7: La comparaison des correspondances trouvé manuellement et celle trouvé avec notre système d'alignement de deux ontologies avec la réutilisation de mapping.	92

INTRODUCTION GÉNÉRALE



1. Contexte de travail

Les ontologies ont été reconnues comme une composante essentielle pour le partage des connaissances et la réalisation de cette vision. En définissant les concepts associés à des domaines particuliers, elles permettent à la fois de décrire le contenu des sources à intégrer et d'explicitier le vocabulaire utilisable dans des requêtes par des utilisateurs.

Toutefois, il est peu probable qu'une ontologie globale couvrant l'ensemble des systèmes distribués puisse être développée. Dans la pratique, les ontologies de différents systèmes sont développées indépendamment les unes des autres par des communautés différentes.

Plusieurs ontologies couvrant un même domaine ou des domaines connexes sont développés indépendamment les unes des autres. Elles sont très souvent source d'hétérogénéité. Cette hétérogénéité posera des problèmes pour l'échange, le traitement, l'intégration, et la recherche d'information, c'est la problématique d'interopérabilité. Pour cette raison, un certain nombre de techniques ont été proposées dont l'une d'elles l'alignement d'ontologies.

2. Problématique

L'alignement des ontologies est une solution pour résoudre l'hétérogénéité sémantique des données, en proposant des correspondances entre des entités sémantiques similaires de différentes ontologies.

Donc l'alignement consiste à réduire l'hétérogénéité, en effet, les ontologies ont été créées pour résoudre le problème de l'hétérogénéité des données sur le web, cependant elles sont devenues elles-mêmes source d'hétérogénéité. L'hétérogénéité se présente au niveau syntaxique, et sémantique.

L'hétérogénéité sémantique présente un défi majeur dans le processus d'alignement des ontologies. Elle est due aux différentes interprétations des objets du monde réel. En effet, les ontologies existantes sont conçues indépendamment, par des concepteurs différents, ayant des objectifs applicatifs différents. Chacune peut donc avoir un point de vue différent sur le même concept.

3. Objectif

L'objectif de ce travail est de proposer une architecture d'alignement d'ontologies pour deux ontologies en OWL qui décrivent le même domaine, en utilisant différentes techniques reposantes sur la définition de mesures de similarité et la réutilisation de mapping afin de trouver les relations entre les entités de ces ontologies;

L'utilisation des techniques de réutilisation des Mapping, ont un impact sur les résultats de l'alignement, en effet, l'utilisation des mesures de similarité, rend le système très lourd, vu l'accès à différentes sources externes pour le calcul de similarité, telles que les terminologies.

Un seul type de mesures se révélant la plupart du temps insuffisant pour détecter une similarité, donc notre système va pouvoir combiner des techniques de base de l'alignement tels que les techniques syntaxiques, structurelles, lexicales et à base d'instances afin d'avoir un alignement pertinent.

Ces similarités vont être raffinées à chaque réutilisation jusqu'à l'obtention d'un mapping stable et plus significatif.

Une fois que la valeur de mapping devient stable, nous la réutiliserons sans recalculer les mesures de similarité ce qui va réduire le temps d'exécution.

Afin d'atteindre le but de notre travail, le mémoire sera présenter en deux parties :

La première partie :

Contient deux chapitres, elle présente le contexte du travail, elle a pour but de présenter les ontologies, et de montrer l'intérêt de l'alignement.

Chapitre 1- Les Ontologies : Ce chapitre propose une généralité sur les ontologies. Nous présentons les ontologies avec leurs caractéristiques, leurs constructions et leurs méthodes d'application, ainsi que les domaines d'application de ces ontologies. Nous allons décrire quelques langages de représentation des ontologies utilisées dans le Web sémantique.

Chapitre 2- État de l'art sur les techniques de mapping d'ontologies : Ce chapitre a pour objectif de dresser un état de l'art sur les techniques de mapping d'ontologies. Tout au long de ce chapitre, nous éclaircissons la notion du mapping en présentant des différentes définitions des concepts liés au contexte du mapping. Nous présentons les classifications des techniques d'alignements selon plusieurs vues, et pour finir une étude comparatives des systèmes d'alignements selon quelques caractéristiques.

Introduction générale

La deuxième partie :

Après les généralités présentées dans la première partie. Cette partie est consacrée pour la conception et le développement du système.

Cette partie met l'accent sur le processus de développement de notre système d'alignement qui se présente comme une suite de phases enchaînées dans un déroulement linéaire, basé sur une succession de phases.

Chapitre 3: La conception : Ce chapitre contient la spécification et l'analyse des besoins qui présentent une compréhension des besoins et des exigences en utilisant des diagrammes UML. Il s'agit de donner des spécifications pour permettre de choisir la conception de notre système. Pour la partie conception, elle élabore la conception générale qui présente l'architecture de notre système, et la conception détaillée qui présente le diagramme de classes, ainsi que le calcul des mesures de similarité. Ce chapitre constitue un point de départ à l'implémentation.

Chapitre 4: Implémentation : Ce chapitre est le résultat de la conception pour réaliser le système de mapping d'ontologies, il s'agit de transformer les éléments décrits lors de la conception en éléments du langage cible.

Chapitre 5: Test et Validation : cette phase est basée sur des expérimentations effectuées sur le système et des tests sur deux ontologies concrètes permettent de vérifier les résultats de l'implémentation en testant la construction du système.

La conclusion de ce mémoire synthétise les principales contributions de notre travail.

CHAPITRE I : Les Ontologies

Chapitre I : Les ontologies

1. Introduction

Le développement et l'exploitation des connaissances en informatique ont tellement évolué, qu'un des principaux objectifs de la recherche actuelle est de ne plus considérer l'ordinateur comme une boîte noire, dans laquelle, sont stockées des informations, mais plutôt comme une machine intelligente avec laquelle on peut dialoguer et créer une coopération. Le système doit alors, 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.

La connaissance visée par ces ontologies est un thème de recherche dans divers axes tels que l'ingénierie des connaissances, la recherche d'information, les systèmes d'information coopératifs, l'intégration intelligente d'information, la gestion des connaissances etc. 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 leurs réutilisations. La définition explicite du concept ontologie soulève un questionnement qui tout à la fois d'ordre philosophique, cognitif et technique.

2. Historique

Une ontologie en informatique est un ensemble structuré de concepts permettant de donner un sens aux informations. Les concepts sont organisés dans un graphe dont les relations peuvent être:

- ✓ des relations sémantiques;
- ✓ des relations de composition et d'héritage (au sens objet) l'objectif premier d'une ontologie est de modéliser un ensemble de connaissances dans un domaine donné.

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

Le mot ontologie provient du grec «*onto*» qui signifie ce qui existe, comme l'être, et l'existant, et «*logos*» qui veut dire étude.

John McCarthy a introduit l'ontologie en intelligence artificielle en 1980, par le principe que les concepteurs de systèmes intelligents fondés sur la logique devraient d'abord établir une base de données complète de ce qui existe, et ensuite utiliser ces données. Les ontologies sont

Chapitre I : Les ontologies

apparues en intelligence artificielle, comme réponses aux problématiques de représentation et de manipulation des connaissances au sein des systèmes informatiques.

3. Définition

Dans le cadre de l'intelligence artificielle Neeches et ses collègues [1] furent les premiers à proposer une définition à savoir « une ontologie définit les termes et les relations de base de vocabulaire d'un domaine, ainsi que les règles qui indiquent comment combiner les termes et les relations de façon à pouvoir étendre le vocabulaire ». En 1993, GRUBER [2] propose la définition suivante: «une ontologie est une spécification explicite d'une conceptualisation». En 1995, GUARINO [3] a modifié légèrement la définition de GRUBER, et la définit par: «une ontologie est une spécification partielle et formelle d'une conceptualisation». En 1997, ces deux dernières définitions sont regroupées dans celle de BORST [4] comme « une ontologie est définie comme étant une spécification formelle d'une conceptualisation partagée ».

- ✓ Conceptualisation : le modèle abstrait d'un phénomène du monde réel par identification des concepts clefs de ce phénomène.
- ✓ Explicite : le type de concepts utilisés et les contraintes liés à leur usage sont définis explicitement.
- ✓ Formelle : une ontologie doit être traduite en langage interprétable par une machine.
- ✓ Partagée : capture la connaissance consensuelle, c'est-à-dire non réservée à quelques individus, mais partagée par un groupe ou une communauté.

4. Rôle des ontologies

Les ontologies qui sont actuellement au centre de la recherche dans le domaine du Web sémantique ont été développées dans le but de représenter des connaissances, que les machines peuvent comprendre, et du fait peuvent manipuler la sémantique des informations. La construction d'ontologies qui nécessite une étude de la connaissance humaine, pour la mise en œuvre des systèmes de manipulation dépend des langages de représentation. L'étude des connaissances humaines et plus largement de l'esprit humain, qui rattache l'Intelligence Artificielle aux sciences humaines, et la création de programmes évolutifs reflétant le travail d'un groupe de chercheurs (artefacts) représentent les dimensions scientifique et technique de la participation des ontologies dans le domaine de l'Intelligence Artificielle. Au fil des

Chapitre I : Les ontologies

années, les méthodologies de construction d'ontologies et les outils de développement adéquats sont apparus.

Partant d'une pratique limitée voir artisanale, les méthodologies de construction d'ontologies et les outils de développement adéquats se sont développés, au fil des années mettant à profit l'expérience et constitue actuellement une véritable ingénierie de construction et de gestion d'ontologie tout au long de leur cycle de vie.

Les ontologies représentent la composante logicielle qui s'intègre dans les systèmes d'information en leur fournissant un aspect sémantique, et de ce fait, leur champ d'application s'est considérablement élargie et couvre les systèmes conseillers comme les systèmes d'aide à la décision, les systèmes d'enseignement assisté par ordinateur " e-learning ", etc., les systèmes de résolution de problèmes et les systèmes de gestion de connaissances.

Parmi les projets d'actualité figure l'utilisation des ontologies qui peut procurer au Web une plate-forme de connaissances permettant, la recherche d'information des points de vue syntaxique et sémantique.

L'enjeu de l'effort engagé est de rendre les machines suffisamment sophistiquées pour qu'elles puissent intégrer le sens des informations, qu'elles ne font que manipuler formellement, à l'heure actuelle. Mais en attendant que des ordinateurs chargés 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. Les constituants d'une ontologie

Les ontologies définissent, le sens des termes et les relations entre eux, elles leur fournissent un vocabulaire commun. Malgré que les ontologies peuvent être très différentes surtout au niveau du traitement de leurs composantes de base telles que les choses, les relations..., elles caractérisent un même univers.

5.1. Les classes/ les concepts

Également appelés classes dans certains travaux. Un concept peut désigner un objet, une idée ou bien une notion abstraite [5]. Dans une ontologie, on désigne un concept par un label (terme, nom). Ce concept est défini par ses différentes relations avec les autres concepts de l'ontologie, ses attributs et les contraintes qui lui sont associées. Un concept peut représenter plusieurs objets.

Chapitre I : Les ontologies

Un concept peut donc décrire une tâche, une fonction, une action, une stratégie, un processus de raisonnement, etc.

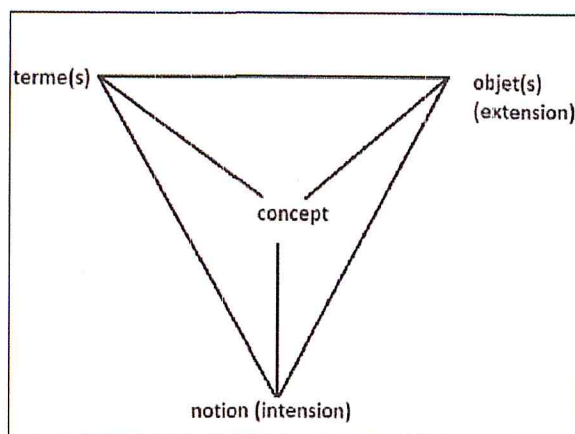


Figure 1: Le triangle sémantique (Ogden et Richards, 1923). [6]

5.2. Les relations et les fonctions :

L'interaction entre les notions d'un domaine définit les relations, On trouve des relations binaires, tertiaire. Certaines relations binaires entre des objets sont considérées comme des rôles comme le définit Borgida [7]. Les fonctions sont des cas particuliers de relations dans lesquelles un élément de la relation est défini à partir des autres éléments.

5.3. Les axiomes :

Les axiomes désignent des vérités indémontrables qui doivent être admises. Ce sont des affirmations considérées comme évidentes sans preuve. Ils permettent de contraindre les valeurs de classes ou d'instances.

5.4. Les instances :

Les instances représentent les éléments des concepts et des relations dans un domaine donné. [8]

6. Le cycle de vie des ontologies

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. Un cycle de vie d'une ontologie débute du besoin qui se transforme en idée, la concrétisation de l'idée qui se traduit par la conception qui est diffusée

Chapitre I : Les ontologies

pour son utilisation. Vient ensuite l'étape de l'évaluation qui donne naissance, le plus souvent à une étape d'évolution et de maintenance du modèle. Une réévaluation de l'ontologie et des besoins devra se faire après chaque utilisation significative. [9]

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. [10]

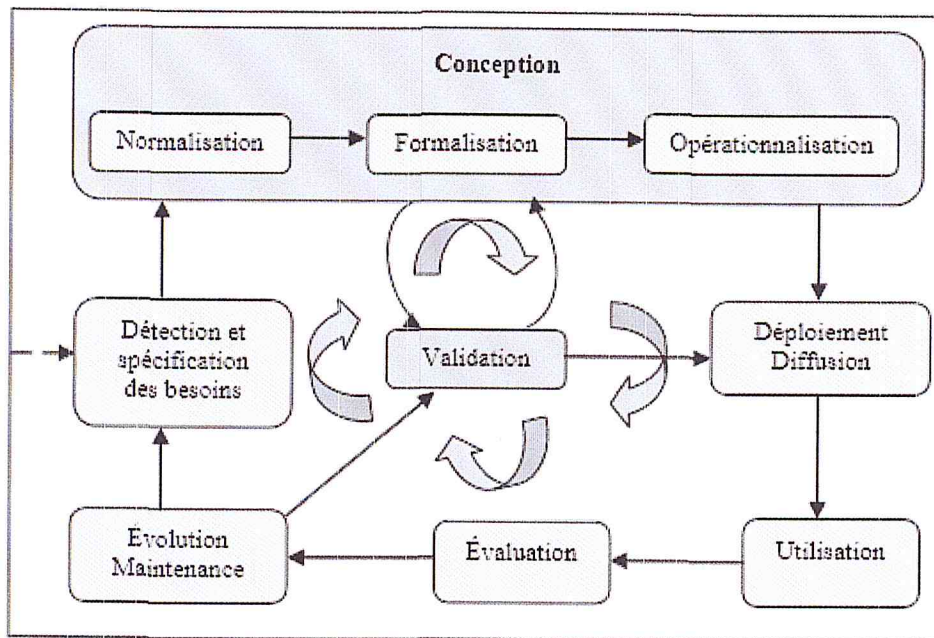


Figure 2: Cycle de vie d'une ontologie [11]

7. Typologie des ontologies

Il existe différentes façons de typer les ontologies. Nous présentons ci-dessous les plus usuelles.

Gomez-Perez distinguent les ontologies selon l'objet de conceptualisation [12] :

- ✓ Ontologie de formalisation des primitives utilisées dans la représentation des connaissances (p. ex. l'ontologie « Frame-Ontology » Gruber T décrit les primitives utilisées dans la représentation des langages basés sur les « frames » : classes, instances, attributs, etc. [13])
- ✓ Ontologie supérieure (« general/common ontologies ») : contient des descriptions hautniveau du monde, les plus généraux que l'on puisse définir : le temps, l'espace, les évènements, la causalité, etc.
- ✓ Ontologie générique (« meta-ontology/ core ontology / generic ontology ») : est réutilisable d'un domaine à l'autre.

Chapitre I : Les ontologies

- ✓ Ontologie de domaine : est réutilisable dans un même domaine, fournit le vocabulaire et les relations décrivant les concepts du domaine, les théories et les principes d'un domaine, etc.
- ✓ Ontologie d'application : contient la connaissance nécessaire pour modéliser une application d'un domaine particulier.
- ✓ Ontologie de tâches : contient la description nécessaire pour la résolution des problèmes associés à des tâches qui ne sont pas nécessairement de même domaine.
- ✓ Ontologie de tâche-domaine : une ontologie de tâches réutilisable seulement dans un même domaine.

Selon la granularité on distingue les ontologies de haut niveau qui décrivent le monde à une granularité moins fine, et les ontologies plus spécifiques, comme celle des tâches, qui décrivent le monde à une granularité plus fine. En fonction de l'exploitabilité de l'ontologie en vue de réaliser des inférences, on distingue l'ontologie « lightweight » qui est une simple hiérarchie, et l'ontologie « heavyweight » qui définit des propriétés plus avancées, permettant ainsi d'y associer des mécanismes d'inférence.

Uschold & Gruninger, classifient les ontologies selon le « niveau de formalisme » [14] :

- ✓ Les ontologies informelles, exprimées en langage naturel.
- ✓ Les ontologies semi-informelles, écrites dans un langage naturel, mais sous une forme limitée et structurée, permettant d'augmenter la clarté et la lisibilité.
- ✓ Les ontologies semi-formelles, exprimées dans un langage artificiel défini de manière formelle.
- ✓ Les ontologies strictement formelles définies elles aussi dans un langage artificiel, mais avec des théorèmes et des preuves sur des propriétés de l'ontologie, telles que la robustesse ou la complétude.

Un autre critère identifié dans la littérature est le niveau de complétude. Par exemple, Bachimont définit trois niveaux [15], comme suit :

- ✓ Niveau 1 - Sémantique : chaque concept a un sens univoque et se distingue d'autres concepts de la hiérarchie (concepts père et frère).
- ✓ Niveau 2 - Référentiel : à ce niveau on parle des concepts référentiels ou formels ; en plus de la description qui est donnée au niveau sémantique, des extensions sont attribuées aux concepts pour bien définir leur sémantique ; « Deux concepts formels sont identiques s'ils possèdent la même extension (ex. : les concepts d'étoile du matin et d'étoile du soir associés à Vénus) ».

Chapitre I : Les ontologies

- ✓ Niveau 3 - Opérationnel/computationnel : « Outre les caractéristiques énoncées au niveau précédent, les concepts du niveau opérationnel ou computationnel sont caractérisés par les opérations qu'il est possible de leur appliquer pour générer des inférences (engagement computationnel). ».

Il existe d'autres critères de classification des ontologies : selon le domaine d'application (ex. ontologies de gestion des connaissances, linguistiques, ontologies pour l'ingénierie, etc.). On peut aussi classer les ontologies selon le but de leur usage : résolution de problèmes, décision, vocabulaire partagé, etc.

8. Domaine d'application des ontologies

8.1. Système d'information

L'intégration d'une ontologie dans un système d'information vise à réduire, voire à éliminer, la confusion conceptuelle et trilogique à des points clés du système, et a 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.

L'ontologie se retrouve maintenant dans une large famille de système d'information.

Elle est utilisée pour :

- ✓ Décrire et traiter des ressources multimédia ;
- ✓ Assurer l'interopérabilité d'application en réseau ;
- ✓ Permettre l'intégration des ressources hétérogènes d'information ;
- ✓ Vérifier la cohérence de modèle ;
- ✓ Faire des approximations logiques; etc.

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

- ✓ Intégration d'information géographique ;
- ✓ Gestion de ressources humaines ;
- ✓ Aide à l'analyse en biologie ;
- ✓ Commerce électronique ;
- ✓ Enseignement assisté par ordinateur ;

Chapitre I : Les ontologies

- ✓ Bibliothèque numérique ;
- ✓ Recherche d'information. Etc...

9. Représentation des ontologies

Une des principales décisions à prendre dans le procédé de développement d'ontologies consiste à choisir le langage (ou l'ensemble de langages) dans lequel l'ontologie sera exprimée et utilisée. Plusieurs langages, dont la syntaxe est basée sur le langage XML, ont été conçus pour une utilisation des ontologies dans le cadre du web sémantique. Parmi eux, les plus importants sont RDF et RDF(S) [17], qui est la recommandation du W3C pour représenter les métadonnées, OIL, DAML+OIL, OWL qui est la recommandation du W3C pour représenter des ontologies.

Par ailleurs, dans le cadre de ses travaux sur le Web sémantique, le W3C a mis en place en 2002 un groupe de travail dédié au développement de langages standards pour modéliser des ontologies utilisables et échangeables sur le Web (Figure 4). S'inspirant de langages précédents comme DAML+OIL et des fondements théoriques des logiques de description, ce groupe a publié en 2004 une recommandation définissant le langage OWL (Web Ontology Language), fondé sur le standard RDF et en spécifiant une syntaxe XML. Plus expressif que son prédécesseur RDFS, OWL a rapidement pris une place prépondérante dans le paysage des ontologies et est désormais, de facto, le standard le plus utilisé.

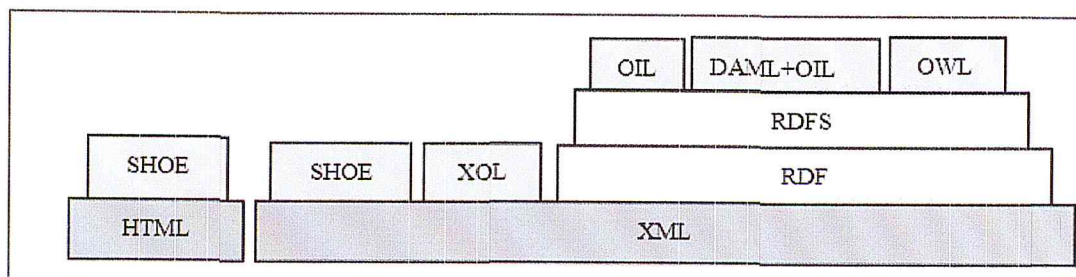


Figure 3: La pyramide des langages d'ontologies basés Web. [18]

- **RDF et RDFS :** (Resource Description Framework (Schema) : RDF est développé et recommandé par W3C. Il permet la représentation et l'échange de métadonnées sur le web grâce à sa syntaxe basée sur XML. Un document RDF est basé sur le triplet (ressource, propriété, valeur) ou encore (sujet, prédicat, objet). Ce type de triplet est appelé un graphe RDF, dans lequel : a) ressource (sujet) représente l'entité décrite, elle

Chapitre I : Les ontologies

est identifiée par une URI (Uniform Resource Identifier) ; b) Propriété (prédicat) représente la relation ou l'attribut utilisé pour décrire la source ; c) valeur (objet) représente une autre ressource ou une donnée. Le RDF schéma a été construit par le W3C comme extension du RDF. Il offre la possibilité d'un modèle de métadonnées qui permet de déclarer les propriétés des ressources, le type des ressources, ainsi que les contraintes associées à ces propriétés et ressources. Le schéma RDF permet de spécifier des concepts, des taxonomies et des relations binaires. Cependant il reste assez peu expressif.

- **OWL (Ontology Web Language)** : OWL est l'extension du langage RDF(S). Il est développé par le groupe de travail sur le web sémantique du W3C qui est basé sur le langage DAML+OIL [19]. Il est aujourd'hui, le standard recommandé par la W3C pour le développement des ontologies et certainement le langage le plus utilisé. Il est fractionné en trois sous-langages qui sont présentés, ci-dessous, selon leur expressivité et complexité croissante.
 - **OWL-Lite** : c'est le moins expressif des sous-langages d'OWL. En fait, il offre la possibilité d'avoir une hiérarchie de classification et des contraintes simples (contrainte de cardinalité limitée de 0 ou 1). L'avantage de ce langage est d'avoir une complexité formelle faible par rapport aux deux autres sous-langages d'OWL.
 - **OWL-DL** : OWL-DL comme son nom l'indique est basé sur la logique de description (DL : Description Logic). C'est le sous-langage qui offre une expressivité maximale en garantissant la complétude et la décidabilité informatique, ce qui signifie que toutes les conclusions sont calculables dans un temps raisonnable.
 - **OWL-Full** : il offre l'expressivité la plus complète, cependant, il ne possède pas les propriétés de complétude et de la décidabilité des calculs liées à l'ontologie. De plus, à ce jour, il n'existe pas d'implémentation complète de OWL-Full.

Chapitre I : Les ontologies

10. Conclusion

Dans ce chapitre nous avons présenté les ontologies avec leurs caractéristiques, leurs constructions et leurs méthodes d'application, ainsi que les domaines d'application de ces ontologies. Nous avons décrit quelques langages de représentation des ontologies utilisés dans le cadre du Web sémantique.

Dans ce qui est présenté dans ce chapitre, il ressort que la notion d'ontologie constitue l'une des approches les plus efficaces pour représenter et analyser et traiter des connaissances.



CHAPITRE II : État de l'art sur les techniques de mapping d'ontologies

Chapitre II : l'état de l'art sur les techniques de mapping d'ontologies

1. Introduction :

Le Web est un environnement fortement distribué, évolutif et hétérogène, une seule ontologie ne peut jamais être suffisante pour représenter toutes les connaissances d'un tel environnement, ce qui nécessite l'utilisation de plusieurs ontologies. Ces derniers peuvent être construits selon plusieurs méthodes, outils et langages, ce qui les rendent souvent hétérogènes et rend cette tâche difficile à accomplir. Le mapping d'ontologie est considéré comme solution afin d'assurer l'interopérabilité des systèmes pour lesquels ces dernières sont conçues.

Dans ce chapitre, nous examinons les outils, les systèmes et les travaux de recherche liés aux techniques de mapping d'ontologies. Nous commençons par définir quelques termes que nous employons tout au long de ce document, puis nous entamons l'existant concernant les techniques de mapping, ainsi qu'une analyse comparative entre ces derniers.

2. Terminologies :

2.1. Mapping ou Correspondances d'ontologies :

Ce sont deux concepts considérés comme similaires. Le mapping entre deux ontologies « o1 ; o2 » signifie que pour chaque entité dans une ontologie « o1 », il faut trouver l'entité correspondante dans l'ontologie « o2 » avec un sens équivalent ou le sens le plus proche. La caractéristique importante du mapping est qu'il produit en sortie un ensemble de correspondances sans modifier les deux ontologies impliquées.

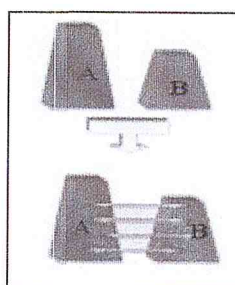


Figure 4: Mapping d'ontologies

2.2. Appariement ou Matching :

Le matching d'ontologies est le processus de définition d'un ensemble de fonctions permettant de spécifier des « correspondances » entre termes [20] [21]. Le matching est le processus pour trouver le mapping (la sortie de l'appariement est un mappage).

2.3. Les méthodes de comparaison ou matchers :

Un matcher est une fonction utilisée pour calculer la distance entre deux entités. Les matchers sont des fonctions qui peuvent être combinées dans le processus de matching.

2.4. L'alignement d'ontologie :

L'alignement d'ontologies est considéré comme le processus qui permet d'amener deux ou plusieurs ontologies hétérogènes à un "accord mutuel" en les rendant ainsi consistantes et mutuellement cohérentes. L'alignement d'ontologies nécessite la transformation des ontologies impliquées en procédant à l'élimination des entités non pertinentes et au rajout des entités manquantes. [22]

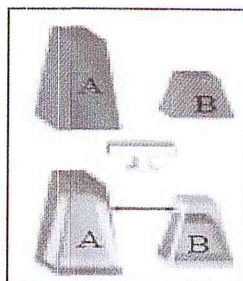


Figure 5: Alignement d'ontologies

3. Classification des techniques de matching d'ontologies :

3.1. La classification des techniques de matching selon Euznat et Schvaiko :

Il existe plusieurs classifications des techniques d'alignement; le modèle le plus représentatif parmi eux est celui de P. Shvaiko et J. Euznat [23] [24]; Ce modèle est constitué de trois couches horizontales, l'ensemble de la classification dans la Figure 6 peut être lu à la fois dans l'ordre décroissant (en se concentrant sur la façon dont les techniques du type d'entrée est interpréter) et croissant (en mettant l'accent sur les types d'objets manipulés) afin d'atteindre la couche de Techniques de base.

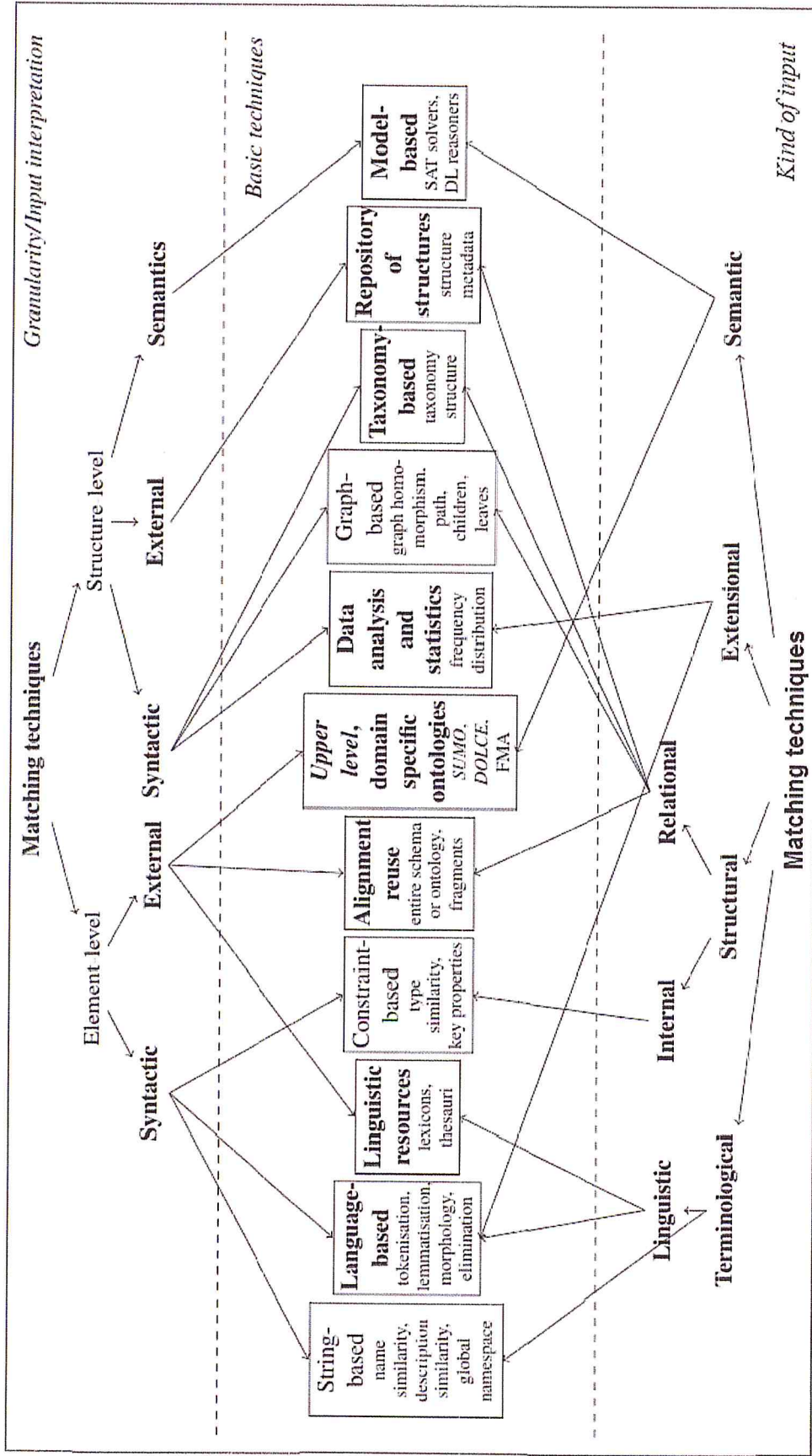


Figure 7 : La classification des techniques de matching [24]

3.1.1. Couche 1 :

Elle classifie les techniques selon :

◆ *Le type d'entrée :*

C'est le type d'entrée utilisé par les techniques du mapping. Il existe de nombreux formalismes de représentation de données hiérarchiques disponibles. Une méthode de mapping est généralement conçue pour fonctionner avec certains types de formalismes. Une méthode peut ainsi considérer des schémas de base de données (de type relationnel, XML, ou objet), des ontologies (décrites en OWL, RDFS, ou autre langage), ou même de simples structures arborescentes (structures des répertoires Web, structures des répertoires d'un système de fichiers). On peut également distinguer les approches en fonction de la nécessité ou non de leur fournir des hiérarchies peuplées. En effet, certaines méthodes s'appuient sur les données extensionnelles des hiérarchies, c'est-à-dire les instances d'une base de données objet ou d'une ontologie, les n-uplets d'une base relationnelle, le contenu des documents XML ou encore les fichiers d'un système de fichiers, etc.

◆ *La granularité :*

C'est la façon dont laquelle les techniques interprètent les informations en entrée, en distingue deux niveaux :

- Techniques basées sur les éléments :

Ce sont les techniques qui calculent des correspondances en analysant des entités ou des instances en isolation et ignorant leurs relations avec d'autres entités ou instances.

- Techniques basées sur les structures :

Ce sont celles qui calculent des correspondances en analysant comment les entités ou leurs exemples apparaissent ensemble dans une structure.

Ces deux techniques se décomposent en :

- Niveau syntaxique : la caractéristique des techniques syntaxiques est qu'elles interprètent les entités en ce qui concerne leurs structures.

Chapitre II : L'état de l'art sur les techniques de mapping d'ontologies

- Niveau externe : les techniques externes exploitent des ressources externes auxiliaires d'un domaine et d'une connaissance commune afin d'interpréter les entrées, ces ressources peuvent être des thésaurus ou des entrées par l'humain.
- Niveau sémantique : ce sont les techniques qui utilisent la sémantique formelle pour interpréter les entrées.

3.1.2. Couche 2 :

Cette couche contient les techniques de bases utilisées pour établir des correspondances entre les entités, chaque technique est portée sur une vue partielle d'ontologie, elles sont utilisables à la fois par les techniques de la couche 1 et la couche 2. Voici quelques techniques:

- ◆ Techniques basées sur l'analyse des chaînes de caractères :

On peut distinguer deux grandes catégories d'analyse des chaînes de caractères, d'une part les méthodes qui prennent en compte deux chaînes de caractères simples, et d'autre part celles qui comparent des ensembles de chaînes.

Techniques:

- Préfixe: prend en entrée deux chaînes de caractères et vérifie si la première chaîne est incluse au début de la deuxième chaîne de caractères (e.g net \equiv network).
- Suffixe: prend en entrée deux chaînes de caractères et vérifie si la première chaîne est incluse à la fin de la deuxième chaîne, exemple : phone \equiv telephone.
- Edit-distance: nombre minimal de caractères qu'il faut supprimer, insérer ou remplacer pour passer d'une chaîne à l'autre, exemple : EditDistance (NKN, Nikon)=2/5=0,4.
- N-grammes: calcule le nombre de séquences n caractères communs entre deux strings.

- ◆ Techniques basées sur la langue :

Techniques Basées sur les techniques de traitement du langage naturel (NLP) en exploitant les propriétés morphologiques des termes en entrée.

Techniques:

Chapitre II : L'état de l'art sur les techniques de mapping d'ontologies

- L'analyse lexicale (ou tokenization) : qui consiste à transformer un flot de caractères en un flot de jetons par un analyseur (tokenizer) qui reconnaît les ponctuations, les caractères blancs, les chiffres, etc. exemple : Hands-Free_Kits= (Hands, Free, Kits).
 - Lemmatisation: analyse lexicale du contenu d'un texte regroupant les mots d'une même famille. Chacun des mots d'un contenu se trouve ainsi réduit en une entité appelée lemme (forme canonique). La lemmatisation regroupe les différentes formes que peut revêtir un mot, soit : le nom, le pluriel, le verbe à l'infinif, etc. (exemple : Kits-> Kit).
- Ces techniques sont souvent appliquées aux noms des entités avant d'avoir exécuté les techniques.

◆ Techniques basées sur les ressources linguistiques :

Ces méthodes reposent sur l'utilisation de ressources externes. Une description de ces ressources externes est présentée comme suit.

- Un lexique ou dictionnaire : Il présente un ensemble de mots auxquels sont associées des définitions écrites en langage naturel. Un mot donné peut avoir plusieurs définitions s'il possède plusieurs sens. Un dictionnaire multilingue est constitué de la même façon qu'un lexique mais la définition est remplacée par son équivalent dans l'autre langage considéré.
- Une taxonomie : Elle présente une hiérarchie de termes. Un thésaurus étant la taxonomie en y ajoutant des relations de synonymie entre termes équivalents
- Une terminologie : Elle est considérée comme étant un thésaurus qui contient également des expressions en plus de mots simples. De plus une terminologie se limite à décrire le vocabulaire d'un domaine précis, contrairement aux autres structures présentées ci-dessus.
- Il existe diverses méthodes qui reposent sur l'utilisation de telles ressources. La méthode la plus simple utilise la synonymie, consiste à déclarer que deux termes sont équivalents simplement s'ils sont présentés comme étant des synonymes.

◆ Techniques à base de contraintes:

Techniques basées sur les contraintes sont des algorithmes qui traitent avec les contraintes internes appliqués aux définitions des entités, telles que les types, cardinalités (et la multiplicité) des attributs, et les clés. Exemple :

Chapitre II : L'état de l'art sur les techniques de mapping d'ontologies

- Comparaison des types de données, exemple : varchar et text.
- domaine des valeurs.

◆ Réutilisation d'alignement:

Techniques de réutilisation d'alignement représentent une autre façon d'exploiter les ressources externes, qui enregistrent les alignements d'ontologies précédemment appariés. Par exemple, quand nous avons besoin de faire correspondre les deux ontologies O et O' , étant donné le mapping entre ces deux ontologies, et le mapping disponible des ressources externes entre O et O' La réutilisation de mapping est réalisé par l'idée que de nombreuses ontologies qui vont être aligné sont semblables aux ontologies déjà aligné, surtout si elles décrivent le même domaine d'application.

Ces techniques sont particulièrement prometteuses lorsqu'il s'agit de grandes ontologies, comprenant des centaines et des milliers d'entités. Dans ces cas, d'abord, de grands problèmes d'alignement sont décomposés en plus petits sous problèmes, générant ainsi un ensemble de fragments de problème d'alignement d'ontologie. Ensuite, la réutilisation des résultats des alignements précédents peut être plus efficacement appliquée au niveau des fragments d'ontologie plutôt qu'au niveau d'ontologies entières. L'approche a été introduite en 2001 [25] et a ensuite été mis en œuvre dans deux comparateurs, i e, (i) réutilisation des alignements d'ontologies entières, ou leurs fragments [26], [27] et [28].

◆ Techniques basées sur les graphes:

Techniques fondées sur les graphiques sont des algorithmes de schémas qui considèrent les ontologies d'entrée comme graphes étiquetés. Les ontologies (y compris les schémas de base de données, et des taxonomies) sont considérées comme des structures de graphes étiquetés. Habituellement, la comparaison de similarité entre une paire de nœuds des deux ontologies est basée sur l'analyse de leurs positions dans les graphes. L'idée derrière cela est que, si deux nœuds de deux ontologies sont semblables, leurs voisins doivent aussi être en quelque sorte similaire.

Avec des techniques basées purement sur graphe, il existe d'autres techniques plus spécifiques basées sur la structure, par exemple, avec des arbres.

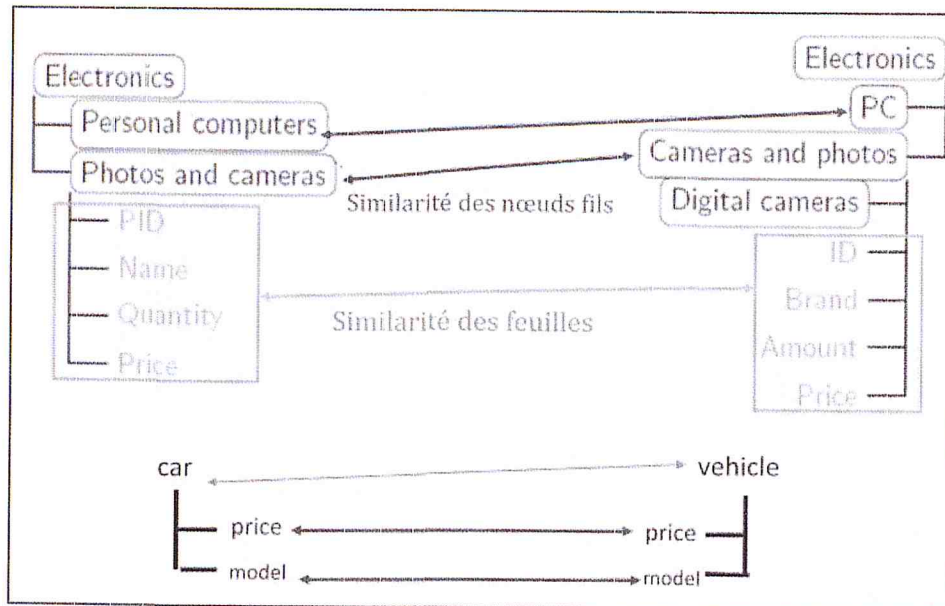


Figure 7: Exemple sur la technique basée sur les graphes [26]

- ◆ Haut niveau et domaine spécifique formel d'ontologie

Elle est utilisée pour lier des catégories et des relations provenant de différentes ontologies de haut niveau, dans le but d'améliorer l'interopérabilité, la communication et la recherche dans le domaine du Web Sémantique. Comme SUMO et DOLCE sont des ontologies de haut niveau. La caractéristique clé de ces ontologies est que ce sont des ontologies basés sur la logique, et par conséquent, les techniques d'alignement utilisées à leur exploitation sont basées sur la sémantique.

- ◆ Techniques basées sur la taxonomie:

Techniques basées sur les taxonomies sont également des algorithmes de graphes qui utilisent que la relation de spécialisation. L'idée de techniques taxonomiques est que « is-a-links » relient les termes qui sont déjà semblables (soit interprété comme un sous-ensemble ou sur-ensemble de l'autre), donc leurs voisins peuvent aussi être en quelque sorte similaire.

- ◆ Référentiels de structures :

Les référentiels de structures stockent les ontologies et leurs fragments ainsi que des mesures de similarité par paires, par exemple, les coefficients varient entre l'intervalle [0,1]. Contrairement à la réutilisation d'alignement, les dépôts de structures stockent seulement les similitudes entre ontologies, pas l'alignement. L'objectif est d'identifier les structures qui sont suffisamment similaires et valent la correspondance en détail, ou la réutilisation d'alignements

existants, ainsi, pour éviter la correspondance des structures dissemblables. De toute évidence, la détermination de similitude entre les structures devrait être moins coûteuse en calcul que l'alignement en détail. L'approche de [28] pour deux structures correspondantes propose d'utiliser certaines métadonnées permettant de décrire ces structures, telles que le nom de la structure, le nom de la racine, le nombre de nœuds, la longueur maximale de chemin, etc. Ces indicateurs sont ensuite analysés et regroupés en un seul coefficient, qui estime la similitude entre eux. Par exemple, deux structures peuvent avoir une correspondance appropriée s'ils ont tous les deux le même nombre de nœuds.

◆ Techniques basées sur les modèles:

Algorithmes basés sur des modèles (ou sémantiquement fondée) gèrent l'entrée en fonction de son interprétation sémantique. L'idée est que si deux entités sont les mêmes, ils partagent les mêmes interprétations. Ainsi, ils sont bien établis en méthodes déductives. Les exemples sont des techniques de satisfiabilité et la description logiques du raisonnement propositionnel.

◆ L'analyse des données et des techniques statistiques :

L'analyse des données et des techniques statistiques sont ceux qui bénéficieraient d'un vaste échantillon représentatif d'une population afin de trouver des régularités et des divergences. Cela aide à regrouper des éléments ou le calcul des distances entre eux. Parmi les techniques d'analyse de données et de classification basée sur la distance, sont l'analyse formelle de concepts et l'analyse de la correspondance.

3.1.3. Couche 3 :

Elle classe les techniques selon les types d'entrées en :

◆ Les techniques terminologiques :

Ces techniques reposent principalement sur des comparaisons de textes, elles s'appliquent sur les noms, les commentaires et les propriétés des concepts afin de trouver ceux qui sont similaires; Le problème de ces comparaisons est l'existence des synonymes et des homonymes, il n'est pas possible d'assumer que deux entités sont similaires si elles portent le même nom, ou qu'elles sont différentes si elles possèdent des noms différents. De plus, l'utilisation des langages différents pour représenter la même entités, ainsi que la variation

syntaxique du même terme qui est due lors de l'utilisation des abréviations, des préfixe et des suffixes,...

Il existe des méthodes de comparaison des termes selon leur considération :

- Méthodes basées sur les chaînes de caractères (texte) : comme leur nom l'indique, ces méthodes considèrent l'entité comme une séquence de lettres. Les résultats obtenus par ces méthodes sont utiles si les concepteurs utilisent des chaînes de caractères similaires pour définir la même entité, en revanche, s'il y a des synonymes avec des structures différentes ces méthodes donnent une mauvaise estimation de la similarité.
- Méthodes basées sur les langages : les entités ici sont considérées comme un texte (des mots) dans un langage donnée, elles font appel généralement aux techniques du traitement automatique du langage (TAL) pour extraire les termes les plus représentatifs à partir d'un texte, elles utilisent généralement des ressources externe tel que les dictionnaires, les thésaurus,...

L'inconvénient de ces méthodes est que les ressources externes ne sont pas toujours disponibles pour certains langages.

- ◆ Les techniques structurelles :

Elles se focalisent sur la comparaison des structures des entités à mapper, en fonction de la nature interne ou externe des structures à comparer, on peut distinguer :

- Méthodes structurelles internes : qui utilise la comparaison de structure interne d'une entité (ex : la comparaison des attributs, des noms, des types des attributs,..) indépendamment des autres entités, ces méthodes sont généralement combinées avec celles des techniques terminologiques.

Ce type de méthodes est très important pour le mapping du fait qu'il peut éliminer les correspondances incompatibles, cependant, il n'offre pas vraiment beaucoup d'information sur les entités à comparer, plusieurs entités très hétérogènes peuvent être décrites par le même structure interne et des entités très proches peuvent avoir des structures différentes.

- Méthodes structurelles externes : qui utilisent la comparaison externe en mettant par exemple en jeu la disposition des entités dans leur hiérarchie, le voisinage,...

Le mapping qui utilise ces méthodes est très performant parce qu'il prend en compte toutes les relations entre les entités ce qui nécessite l'utilisation d'autres méthodes terminologiques.

Chapitre II : L'état de l'art sur les techniques de mapping d'ontologies

◆ Les techniques extensionnelles :

Elles se basent sur la comparaison des instances des entités des ontologies à mapper, en fonction de l'intersection des extensions de deux entités, on peut distinguer :

- Méthodes de comparaison d'extensions communes : elles sont utilisées lorsqu'il y a un ensemble des instances communes entre les ontologies, elles considèrent que deux entités sont similaires si leur intersection se réduit à l'une des entités (en terme d'instances).

Le problème de ces méthodes est la capacité d'engendrer des fautes, un petit ensemble de données incorrecte peut conduire à des résultats erronés, de plus si deux entités ne partagent pas les mêmes instances, ces méthodes vont donner un 0 comme similarité sans tenir compte à la similarité entre les instances.

- Méthodes d'identification des instances : elles sont appliquées quand il n'existe pas un ensemble commun des instances, il est possible d'identifier qu'une instance d'un ensemble donné correspond à une instance d'un autre ensemble par le calcul de similarité entre les instances.

Ces méthodes sont utilisables dans le cas où l'on sait que les instances sont identiques (ex : lorsqu'on intègre deux ontologies de la même entreprise), mais quand il s'agit des entreprises différentes ou des ontologies qui n'ont aucune relation entre eux, ces méthodes ne seront jamais applicables.

Généralement, le savoir de la partie extension des ontologies est inestimable pour le mapping car elle est indépendante de toute conceptualisation, cependant si l'ensemble des instances n'est pas disponible, ces méthodes ne pourraient pas s'appliquer, dans tel cas, les autres techniques sont plus adéquates.

◆ Les techniques sémantiques :

Elles sont très souvent des techniques qui se basent sur des modèles théoriques et se focalisent sur des techniques déductives exploitent très souvent la logique de description (test de subsumption) ou le calcul des prédicats.

- Méthodes basées sur des ontologies externes : si deux ontologies qui ne possèdent pas d'un terrain commun pour la comparaison vont être mappées, ces méthodes lui

Chapitre II : L'état de l'art sur les techniques de mapping d'ontologies

permettent d'utiliser une ontologie intermédiaire qui définit un contexte commun ou des connaissances de bases pour les deux ontologies.

- Méthodes déductives : ces méthodes ne sont pas très performantes seules pour une tâche inductive comme le mapping. En revanche, elles ont une grande valeur pour le mapping une fois que les correspondances sont générées, du fait qu'elles ont la capacité de détecter les inconsistances dans les mapping.

3.2. La classification des techniques de matching selon Ehrig :

La classification d'Ehrig [29] qui est basée sur deux dimensions, Horizontale et verticale. La dimension horizontale contient trois couches :

- Couche de données : le mapping entre entités se fait par la comparaison des données.
- Couche ontologique : qui est elle-même divisée en quatre couches :
 - Réseau sémantique : les ontologies sont vues comme des graphs des concepts et des relations, le mapping est réalisé par la comparaison des graphs.
 - Logique de description : apporte un compte sémantique aux ontologies, le mapping inclut la similarité des taxonomies basées sur les relations de subsumption, il prend en compte j
 - Restrictions et règles: le mapping à ce niveau est basé sur l'idée que s'il existe deux relations similaire entre deux concepts alors les concepts peuvent être considérés similaires.
- Couche contexte : elle est concernée par l'usage des entités dans le contexte d'une application.

La dimension verticale représente les connaissances spécifiques d'un domaine, qui peuvent être situées à n'importe qu'elle couche de la dimension horizontale.

3.3. La classification des techniques de matching selon Zanobini:

La classification de Zanobini [30], qui classifie les techniques de mapping en trois catégories:

- Syntaxique : représente les techniques purement syntaxiques, comme les techniques basées sur les chaînes de caractères,...
- Pragmatique : représente les techniques relatives à la comparaison des instances.
- Conceptuelle : représente les techniques qui travaillent avec les concepts et compare leur sens pour établir le mapping.

Cette multitude des techniques de la découverte des correspondances est naturelle, une fois que chaque technique est portée sur une vision partielle d'ontologies. Il existe plusieurs autres classifications de techniques. Nous avons présenté les plus utilisées.

Nous constatons qu'il n'existe pas une technique meilleure que l'autre, ce sont nos besoins, les structures et natures des ontologies à mapper qui influencent sur le choix des techniques, nous avons remarqué aussi, que ces techniques ne peuvent pas s'appliquer tout seules, chacune peut profiter des avantages fournies par l'autre.

4. Composition des techniques de mapping :

Du point de vue de Rahm et Bernstein [25], il existe deux manières de combiner les différentes techniques de mapping au sein de l'approche à mettre au point.

4.1. L'approche hybride :

L'hybridation permettant de combiner, en un seul algorithme, plusieurs techniques basées sur différents critères et types d'information. Les méthodes hybrides ont l'avantage d'obtenir de meilleures performances que l'exécution de plusieurs algorithmes individuels séparément. En effet, la prise en compte de plusieurs critères simultanés permet d'écartier directement des relations n'en satisfaisant qu'un seul et ainsi d'éviter de parcourir en totalité et/ou plusieurs fois les structures à aligner.

4.2. L'approche composite :

La composition permettant de combiner les résultats produits par plusieurs algorithmes exécutés de manière indépendante. Lorsque les critères étudiés sont indépendants, il est alors plus intéressant d'utiliser une approche composite qui permet plus de flexibilité. Les méthodes composites ont l'avantage d'être modulaires et ainsi d'être adaptables plus facilement à différentes représentations de structures d'entrée. Cependant, les méthodes composites

nécessitent de fusionner les résultats produits par les différents algorithmes. Nous distinguons deux approches principales de composition d'algorithmes:

- La composition parallèle, comme décrit dans la figure 9, où les résultats des algorithmes individuels sont obtenus de manière indépendante, puis agrégés pour former le mapping final;

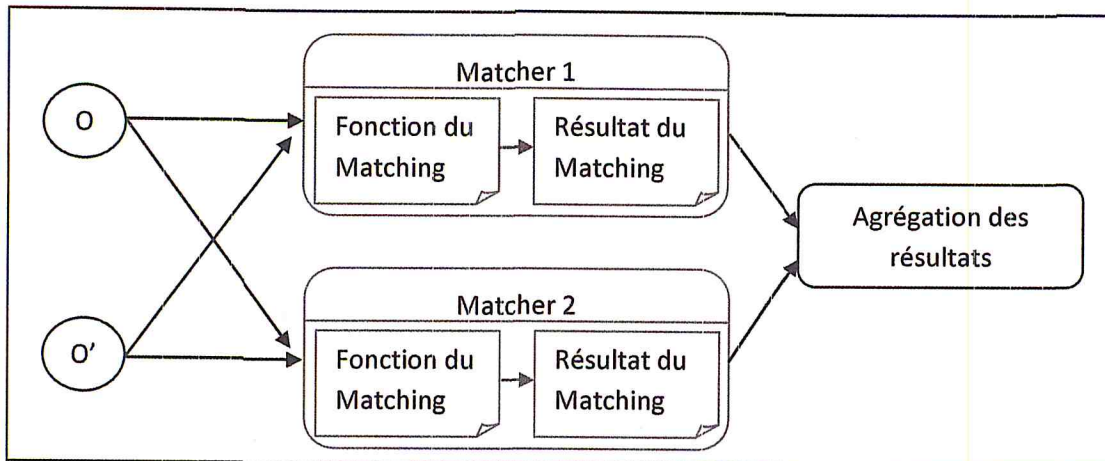


Figure 8: Composition parallèle des matchers

- La composition linéaire ou séquentielle, comme décrit dans la figure 10, où les résultats produits par un algorithme servent d'entrée à un suivant et ainsi de suite. La production du mapping final est ainsi successivement raffinée.

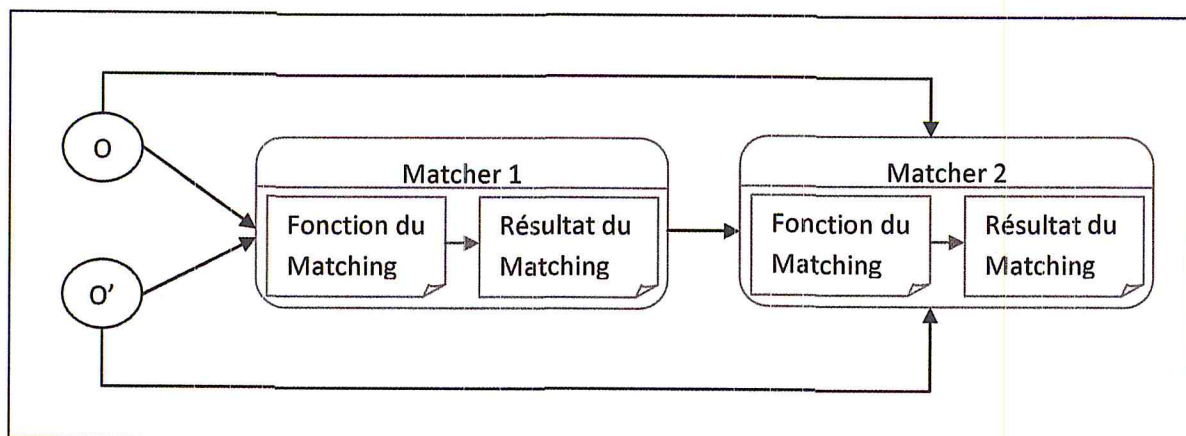


Figure 9: Composition séquentielle des matchers

4.3. Comparaison entre l'approche hybride et l'approche composite:

Hybride	Composite
Intègre plusieurs critères dans un seul algorithme	Intègre les résultats de plusieurs techniques d'alignement exécutées séparément
Les meilleurs paramètres sont stabilisés et une plus grande performance est réalisée	Grande flexibilité
Les techniques d'alignement sont exécutées dans un ordre fixe à l'avance	Les techniques d'alignement sont sélectionnées
Difficile à mettre à jour	Nécessite plus de travail pour combiner les techniques d'alignement

Tableau 1: Comparaison d'approches hybrides vs. Approches composites

5. Analyse comparative entre quelques systèmes existants de mapping :

Un grand nombre de matching sont encore réalisés manuellement par des experts et cela ne devient plus possible tant le nombre et la taille des ontologies augmentent. L'utilisation d'un outil graphique semi-automatique (suggérant différentes relations de correspondance) permet de minimiser l'intervention humaine et d'accélérer le traitement.

Des travaux ont été menés sur ce front dans différents contextes : pour la traduction et l'intégration de schémas, dans le domaine de la représentation des connaissances, pour l'apprentissage automatique et la recherche d'information, pour l'alignement et la fusion d'ontologies.

Parmi les premières recherches à ce sujet, [25] jugent que beaucoup de ces travaux ont été motivés par l'intégration de schémas de bases de données : l'appariement de schémas est la première étape vers l'intégration, permettant ainsi de récupérer les données dans la vue globale. Le matching entre schémas sert également pour traduire des données sources dans le format interne d'un entrepôt de données, pour traduire les messages lors des échanges du e-

commerce, etc. C'est seulement depuis 1999-2001 que des travaux similaires commencent à émerger dans le domaine des ontologies, notamment avec les publications [31] et [32].

Par ailleurs, il existe aujourd'hui de nombreuses méthodes automatiques permettant d'aligner des ontologies. Ces méthodes d'alignement sont basées sur des techniques très variées et obtiennent des performances très différentes en fonction des caractéristiques des ontologies à aligner. Dans ce contexte, il existe une campagne annuelle d'évaluation des outils d'alignement, appelée OAEI (Ontology Alignment Evaluation Initiative) [33], qui permet de comparer les résultats obtenus par les méthodes d'alignement participantes sur différents jeux d'ontologies. Cette campagne OAEI tente d'évaluer les algorithmes de mapping pour normaliser et améliorer le travail sur l'alignement d'ontologie. Parmi les principaux objectifs de cette initiative:

- L'évaluation des systèmes d'alignement et d'appariement ;
- La comparaison de la performance des techniques de mapping ;
- L'amélioration des techniques d'évaluation.

5.1. Quelques systèmes d'alignement d'ontologies :

- **COMA/COMA++**

COMA/COMA++ est un outil développé par Do et Rahm [34]. C'est un outil générique qui permet de combiner un ensemble de méthodes différentes. Il implémente par défaut douze méthodes, pour la plupart basées sur des comparaisons de chaînes de caractères. Le processus de mise en correspondance peut s'exécuter de façon itérative et interactive. COMA intègre également une méthode propre qui cherche à réutiliser des résultats précédents pour une nouvelle génération de mappings. COMA++ est une évolution de COMA qui améliore ses algorithmes et son interface graphique.

- **S-Match** [35]

S-Match est un algorithme et un système pour chercher sémantiquement des correspondances, basé sur l'idée d'employer le moteur de la satisfiabilité propositionnelle (SAT) [36] pour le problème de mise en correspondance des schémas. Il prend comme entrée deux graphes des concepts (schémas), et produit en sortie des rapports entre les concepts tels que l'équivalence, overlapping, différence (mismatch), plus général ou plus spécifique. L'idée principale de cette approche est d'utiliser la logique pour coder le concept d'un nœud dans le graphe et d'appliquer SAT pour trouver des rapports. Le concept à un nœud, qui est alors transformé en formule propositionnelle, est la conjonction de tous les concepts des étiquettes

des nœuds sur le chemin de la racine du graphe jusqu'au nœud en question. Le concept d'étiquette d'un nœud est construit en deux étapes : (i) la normalisation de l'étiquette : telle que la tokenization, la lemmatisation, et (ii) l'extraction du sens de l'étiquette normalisée (des lemmes) à partir de WordNet [37]. Ensuite, les relations sémantiques (l'équivalence, plus général, plus spécifique) entre deux étiquettes de deux schémas sont (i) calculées grâce aux « matchers », les modules qui calculent la similarité entre deux étiquettes en employant des mesures de similarité de base, telles que la similarité des préfixes, des suffixes, la similarité de n-gram; ou bien (ii) déduites en employant des matchers qui exploitent la sémantique dans WordNet, la similarité entre des hiérarchies, la similarité entre des commentaires [38]. Ces relations sémantiques sont aussi encodées en logique. Enfin, le rapport entre deux concepts qui doivent être prouvés est également converti en formule propositionnelle. Le moteur SAT calcule sur l'ensemble de formules propositionnelles pour vérifier si le rapport supposé est vrai ou faux. Cela permet donc de déduire des correspondances entre deux ontologies.

- **NOM /QOM**

Marc Ehrig et Steffen Staab présentent l'algorithme **QOM** (Quick Ontology Mapping) dans [39]. Cet algorithme est un échange entre la qualité d'alignement et la vitesse de trouver des alignements, il s'agit d'une variante optimisée de **NOM** [40]. Il est employé pour mettre en correspondance des ontologies « light-weight » (plutôt des thesaurus), la structure des répertoires dans des ordinateurs personnels, le WordNet, ou l'UMLS. Ce sont des taxonomies ayant un nombre énorme de concepts (>104 concepts).

Ehrig et Staab ont montré que leur QOM met en correspondance ces ontologies dans un délai acceptable sans sacrifier beaucoup la qualité du résultat final. Comme NOM, QOM représente des ontologies en OWL et utilise aussi des mesures de similarité telles que l'égalité des chaînes de caractères, la similarité des chaînes sur des noms de concept, de relation, d'instance ; la similarité des ensembles, etc.

Cependant, QOM a quelques modifications en comparaison avec NOM pour réduire la complexité de calcul, donc le temps d'exécution, telles que des stratégies de sélection des candidats à comparer (par hasard, des candidats ayant des étiquettes à proximité dans la liste triée...) ; des ensembles à comparer sont aussi limités (par exemple, QOM ne compare que deux ensembles de concepts parentaux directs de deux instances, au lieu de tous les concepts ancestraux comme dans NOM).

Chapitre II : L'état de l'art sur les techniques de mapping d'ontologies

fichier OWL qui contient la fusion sera produit en sortie. Les stratégies envisagées sont Matching lexicale, Matching sémantique utilisant Wordnet, vérification de similarité des propriétés et des fonctions Heuristiques. [43]

5.2. Comparaison des systèmes d'alignement présentés (II.5.1) selon les techniques de mapping de base utilisées :

Le tableau 2 présente une recherche bibliographique sur les systèmes de mapping présentés dans la section (II.5.1), selon les techniques de bases utilisées, l'objectif est de comparer leurs processus d'alignement de base, en particulier leurs tâches de calcul de similarité à l'étape d'alignement basé sur la classification de Schvaiko et Euznat [24].

Techniques Systèmes	Automation	String- based	Language- based	Constraint- based	Linguistic Resources	Alignement Reuse	Graph- based	Taxonomy- based	Repository of Structures	Model- based
COMA	Full / Semi	✓	✓	✓	✓	✓	✓	-	-	-
COMA++	Full / Semi	✓	✓	✓	✓	✓	✓	-	-	-
S-Match	Full	✓	✓	-	✓	-	-	✓	-	Semantic Reasoning
NOM	Full	✓	-	✓	✓	-	-	✓	✓	-
QOM	Full	✓	-	✓	✓	-	-	✓	-	-
GLUE	Semi	-	-	-	-	-	-	Probabilistic Reasoning	-	-
SMTO	Semi	✓	-	-	✓	-	✓	✓	-	-
C.R. Rene Robin & G.V. Uma	Full	✓	✓	-	✓	-	✓	-	-	-

Tableau 2: Comparaison de quelques systèmes d'alignement selon les techniques de mapping de base utilisées

5.3. Comparaison des caractéristiques externes des systèmes d'alignement présentés (II.5.1):

Le tableau 3 décrit pour les systèmes de la section (II.5.1) leurs caractéristiques générales, c. -à-d., le type de l'ontologie à aligner (l'entrée), leurs besoins et le type de sortie, le niveau des relations prises en considération et éventuellement les cardinalités des alignements produits.

Systemes Caractéristiques	COMA	COMA++	Match	NOM	QOM	GLUE	SMTO	C.R. Rene Robin & G.V. Uma
Entrées	Schéma relationnel, schéma XML, OWL	Schéma relationnel, schéma XML, OWL	Classification, schéma XML, OWL	RDF, OWL	RDF, OWL	Schéma relationnel, Schéma XML, Taxonomie	Schéma relationnel, schéma XML, OWL, RDF	OWL
Sorties	Alignement	Alignement	Alignement	Alignement	Alignement	Alignement	Intégration	Fusion
Besoins et Opérations	Utilisateur: Traduction de données	Utilisateur: Traduction de données	Auto	Auto, Instances (opt.)	Auto, Instances (opt.)	Auto, Instances, Formation	Utilisateur: Confirmation des similarités complexes	Auto
Relation	↔	↔	↔ ↔	-	-	-	-	-
Cardinalité	(0,1)-(0,1), (0,1)-(0,n), (0,n)-(0,n)	(0,1)-(0,1), (0,1)-(0,n), (0,n)-(0,n)	-	-	(0,1)-(0,1)	-	-	-
Domaine d'application	Générique	Générique	Générique	-	-	-	Générique	-

Tableau 3: Comparaison des systèmes présentés selon leurs exigences les relations et les cardinalités.

Chapitre II : l'état de l'art sur les techniques de mapping d'ontologies

Dans le tableau 3, la ligne **Entrées** représente l'entrée prise par les systèmes. En particulier, elle mentionne les langages utilisés pour représentation les ontologies que les systèmes sont capables de gérer.

La ligne **Sorties** représente les résultats donnés par les systèmes: Alignement signifie que le système renvoie un ensemble de correspondances.

La ligne **Besoins et Opérations** représente les ressources qui doivent être disponibles pour que le système fonctionne. Cela couvre l'aspect automatique qui est ici notée utilisateur lorsque la rétroaction des utilisateurs est nécessaire, et automatique lorsque le système fonctionne sans intervention de l'utilisateur (bien sûr, les utilisateurs peuvent influencer le système en fournissant les entrées initiales).

La ligne **Relation** représente les relations détectées entre les entités des ontologies. La quasi-totalité des méthodes d'alignement se limitent à la relation d'équivalence (\Leftrightarrow). S-Match est le seul système prenant en considération la relation d'équivalence ainsi que la relation d'implication (\Rightarrow).

La ligne **Cardinalité** représente les types de cardinalités proposé par les systèmes. Les méthodes d'alignement d'ontologies proposent généralement un seul type de cardinalités (dans la majorité des cas une cardinalité de type (0,1)-(0,1)). Cependant, la méthode COMA offre encore une grande flexibilité au niveau de la cardinalité en autorisant les trois types de cardinalités.

La ligne **Domaine d'application** représente le domaine des ontologies que les systèmes traitent. Chaque ontologie traite un seul domaine. Ici générique veut dire que le système aligne les différents domaines des ontologies.

6. Conclusion :

Ce chapitre a eu pour objectif de dresser un état de l'art sur les techniques de d'alignement d'ontologies. Tout au long de ce chapitre, nous avons essayé d'éclaircir la notion du mapping en présentant des différentes définitions des concepts liées au contexte du mapping. Nous avons présenté les classifications des techniques d'alignements selon plusieurs vues, et pour finir une étude comparative de quelques techniques d'alignements selon quelques caractéristiques, ainsi que les techniques de base utilisées.

CHAPITRE III : Conception du système de mapping d'ontologies

1. Introduction

« Tout problème de conception débute par un effort pour parvenir à l'adaptation réciproque entre deux entités »¹. Donc, la conception peut se conformer à une logique de propositions, de diagrammes et de probabilités susceptibles d'assurer une solution finale satisfaisante et une architecture de qualité.

L'objectif de ce chapitre est de présenter la conception du système de mapping d'ontologies. Cette partie élabore en premier le diagramme de cas d'utilisation globale puis quelques diagrammes de cas d'utilisation essentiel de notre système.

Les diagrammes de séquences, pour détailler les démarches du système et ses interactions ainsi que les diagrammes d'activité et le diagramme de classe, et enfin l'architecture de notre système.

Ainsi, le modèle que nous cherchons à construire n'a pas la prétention d'être une description exhaustive de tous les processus mobilisés au cours de la conduite de la conception. Les questions auxquelles nous tentons de répondre portent sur les besoins spécifiques pour :

- Assurer les tâches essentielles de conduite des activités de chacun des acteurs.
- Assurer une cohérence et une bonne signification entre les résultats du mapping, issues du mapping entre les ontologies.
- Minimiser le temps de réponse du processus d'alignement.
- Assurer une bonne signification des résultats, en appliquant le raffinement sur les mapping résultants.

C'est donc à ce point particulier du processus de conception que s'intéresse notre modélisation.

2. Cycle de vie d'un logiciel

Le cycle de vie d'un logiciel est un ensemble de phases permettant de transformer à travers un logiciel les besoins en des processus de traitement automatique d'information en provenance de son utilisateur, de son environnement ou de lui-même pour répondre à ces besoins. En ce qui concerne notre système nous avons suivi Le modèle en «Cascade» proposé par W. Royce [44] qui est considéré comme le modèle classique de développement d'un logiciel. Il est vu comme un processus séquentiel comprenant les phases suivantes: la Spécification initial des besoins, la Conception Préliminaire qui est l'analyse pour notre système; la Conception Détaillée, l'Implémentation, des Tests.

¹ Christophe ALEXAN DER, la synthèse de la forme.

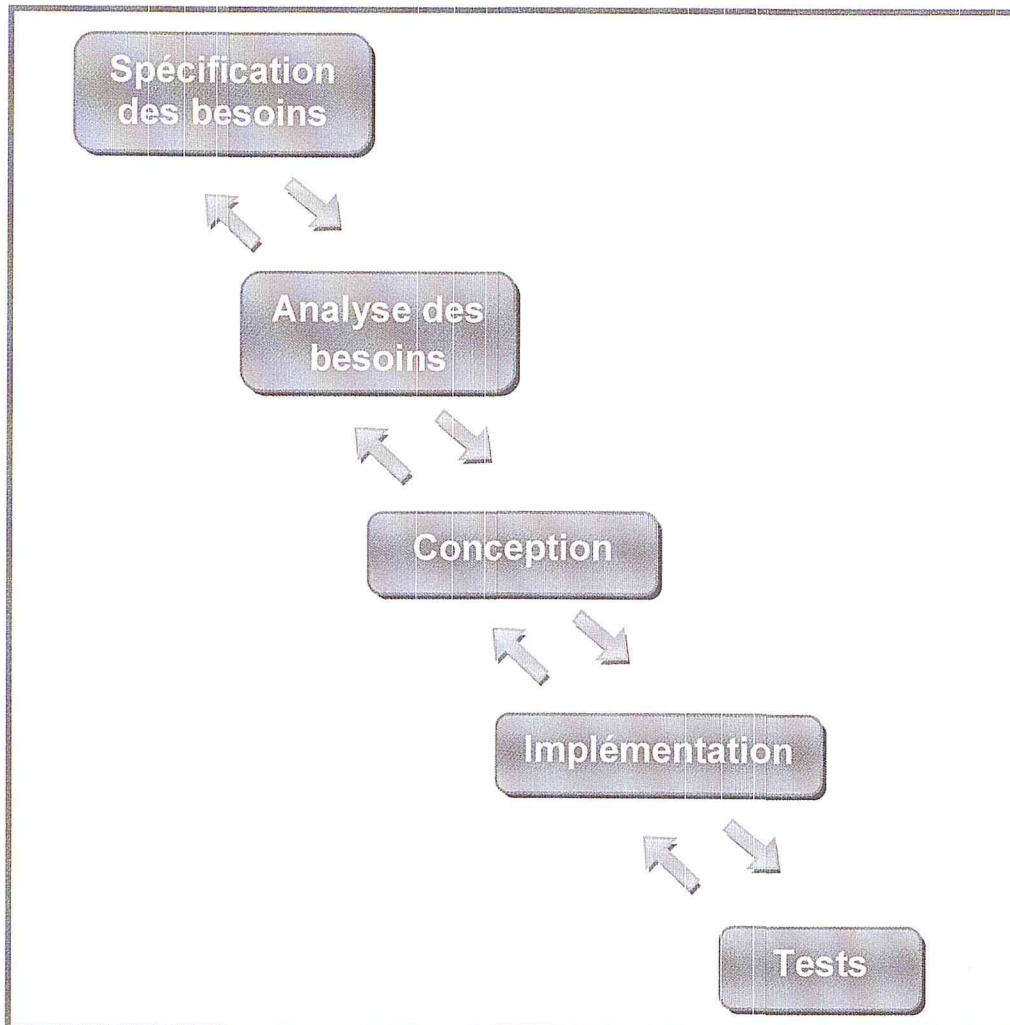


Figure 10: Cycle de vie de notre système en « cascade »[44]

3. Spécification initial des besoins

Cette phase consiste à comprendre le contexte du système. Il s'agit de déterminer les fonctionnalités et les acteurs les plus pertinents, et d'identifier le cas d'utilisation global.

3.1. Les diagrammes de cas d'utilisation

Un cas d'utilisation (*use case*) représente un ensemble de séquences d'actions qui sont réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier. [45]

Chapitre III : Conception du système de mapping d'ontologies

- **Les Acteurs** : Les acteurs sont des entités externes qui interagissent avec le système, cependant l'activité du système a pour objectif de satisfaire les besoins de l'acteur [42]. Les acteurs humains pour notre système sont :

- **L'administrateur** : la personne qui gère et aligne des ontologies, normalise et ajoute des mapping au système, gère les utilisateurs système de façon globale.
- **L'utilisateur général** : la personne qui utilise le système pour aligner des ontologies, et garder le résultat pour une utilisation externe. Il s'agit bien sûr de l'acteur le plus important, celui pour lequel le système existe.

- **Le cas d'utilisation global :**

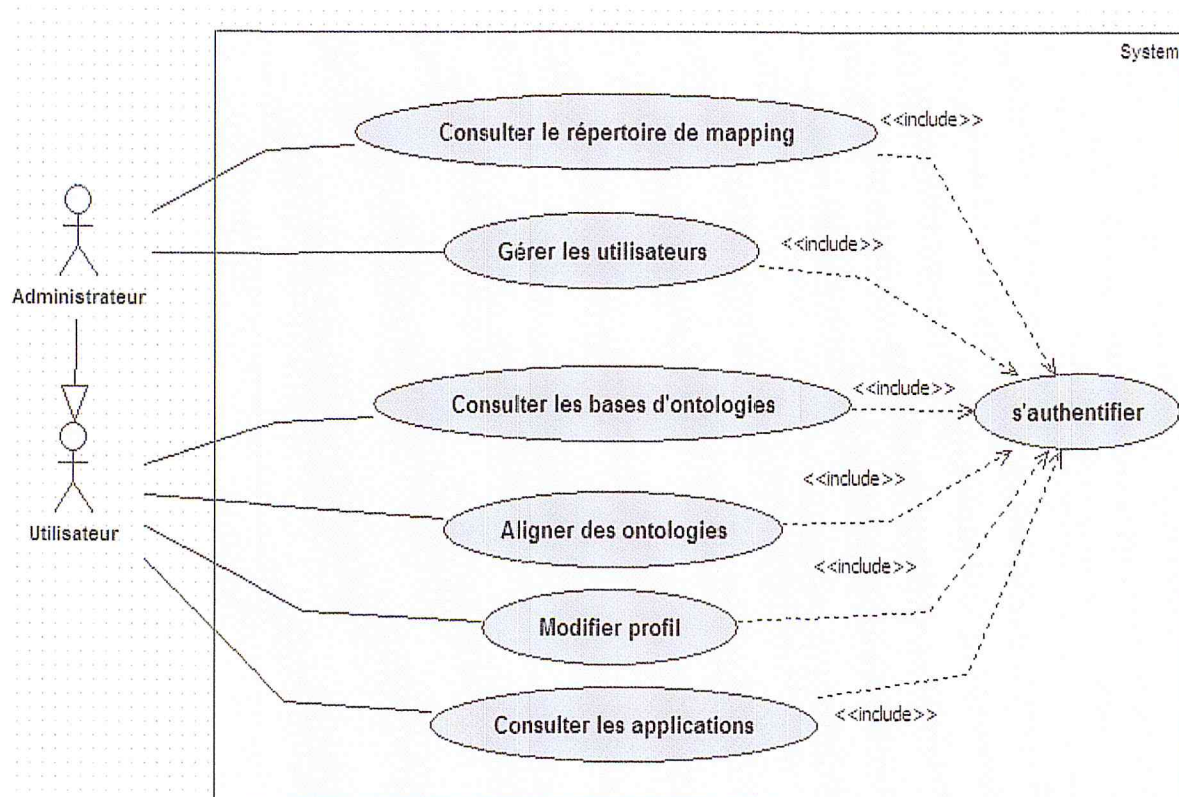


Figure 11: Diagramme de cas d'utilisation global

Cas d'utilisation	Description
Consulter le répertoire de Mapping	L'administrateur peut consulter le contenu du répertoire de Mapping.

Chapitre III : Conception du système de mapping d'ontologies

Gérer les utilisateurs	L'administrateur gère les utilisateurs inscrit (ajout, modification, suppression, recherche)
Consulter les bases d'ontologies	L'administrateur et l'utilisateur peuvent consulter le contenu des bases d'ontologies.
Consulter les applications	L'administrateur et l'utilisateur peuvent consulter leurs applications dédiées.
Aligner des ontologies	L'administrateur et l'utilisateur peuvent aligner des ontologies importées dans la base d'ontologie.
Modifier le profil	L'utilisateur peut modifier son mot de passe ou son adresse e-mail.

❖ Remarques :

- Afin qu'un utilisateur ou un administrateur puissent utiliser ce système, ils doivent s'authentifier avec un nom d'utilisateur et un mot de passe.
- Une personne n'ayant pas de compte, peut créer un compte à partir de l'interface d'authentification.
- L'administrateur peut ajouter une personne autant qu'utilisateur, le supprimer, le modifier, ou chercher un utilisateur inscrit dans la base de données de notre système.

- **Cas d'utilisation détaillé de L'administrateur:** Les diagrammes suivants représentent le détail de la partie essentiel « Application » de notre système pour chaque acteur.

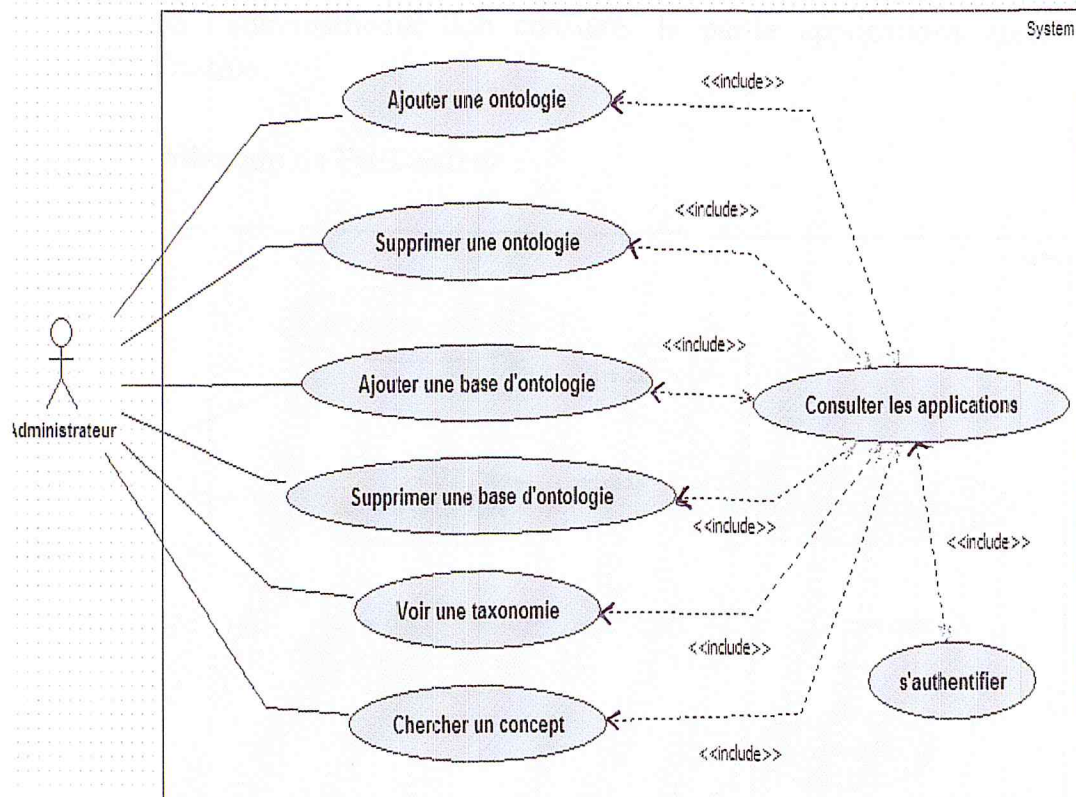


Figure 12: Diagramme de cas d'utilisation d'administrateur « Applications »

Cas d'utilisation	Description
Ajouter une ontologie	L'administrateur peut importer des ontologies à une base d'ontologie existante.
Supprimer une ontologie	L'administrateur peut supprimer des ontologies d'une base d'ontologie existante.
Ajouter une base d'ontologies	L'administrateur peut ajouter une base d'ontologies.
Supprimer une base d'ontologie	L'administrateur peut supprimer une base d'ontologies.
Voir taxonomie	L'administrateur peut voir une taxonomie d'une ontologie choisie.
Chercher un concept	L'administrateur peut voir les ontologies qui contiennent un concept recherché.

Chapitre III : Conception du système de mapping d'ontologies

diagrammes de séquences qui expliqueront le processus d'alignement d'ontologies et l'ajout des mapping.

4.1. Les diagrammes de séquences et d'activités

Nous allons maintenant décrire de façon détaillée deux cas d'utilisation que nous avons identifiés dans la section 3. Nous apprendrons ainsi à remplir une fiche-type pour chaque cas d'utilisation. Nous compléterons cette description textuelle par une représentation graphique UML très utile.

- L'alignement des ontologies:

Acteur principal: Utilisateur.

Objectif : l'utilisateur veut aligner deux ontologies.

Pré-condition : L'utilisateur doit s'authentifier au système.

Post-condition : Un alignement.

Le scénario nominal :

N° d'ordre	Explication
1	Un utilisateur choisit dans son interface l'icône Alignement
2	Le système lui affichera l'interface d'alignement d'ontologies
3	L'utilisateur va choisir deux ontologies ainsi que leurs situations
4	Le système va charger les deux ontologies choisies par l'utilisateur
5	L'utilisateur doit cliquer sur le bouton Aligner pour lancer le processus d'alignement
6	Le système va chercher si un mapping existe dans le répertoire de mapping qui convient à cet alignement
7	Si le système trouve un mapping il le réutilise.
8	Si le système ne trouve pas le mapping convenable, il utilisera la classe de mesure de similarité pour calculer les mapping

Chapitre III : Conception du système de mapping d'ontologies

- 9 Le système va charger les résultats de l'alignement
- 10 Le système Affichera à l'utilisateur l'alignement des ontologies choisies

Alternatives :

6a- Le système trouve le mapping dans le répertoire de mapping :

1. Le système réutilise le mapping trouvé pour calculé la mesure de similarité globale.

8a- Le système ne trouve pas le mapping dans le répertoire de mapping :

1. Le système calcule les mesures de similarités.
2. Le système ajoute le mapping au répertoire de mapping.

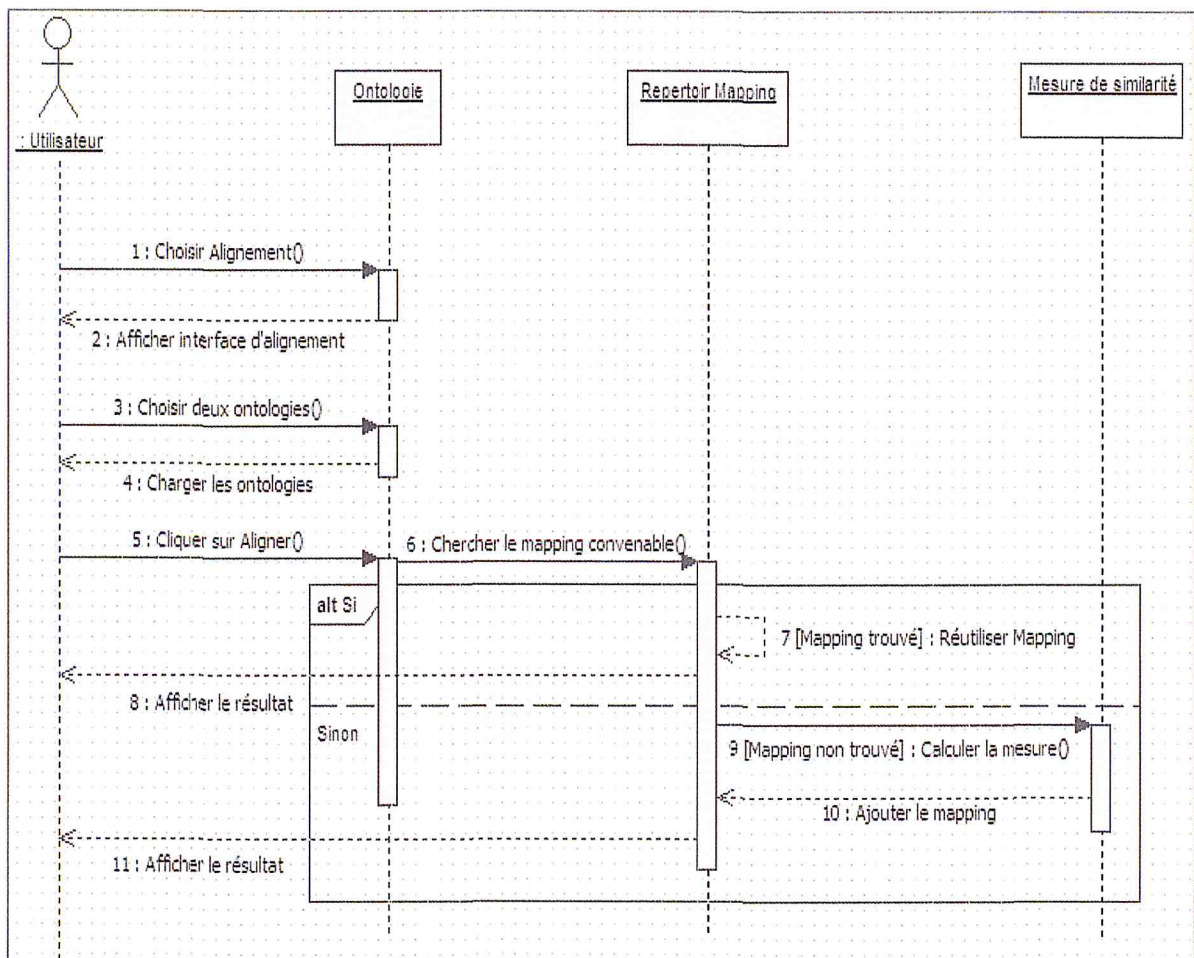


Figure 14: Diagramme de séquence du processus d'alignement d'ontologie

❖ **Remarque :**

- Un Administrateur peut aussi Aligner des ontologies.

◆ **Diagramme d'activité « Alignement »:**

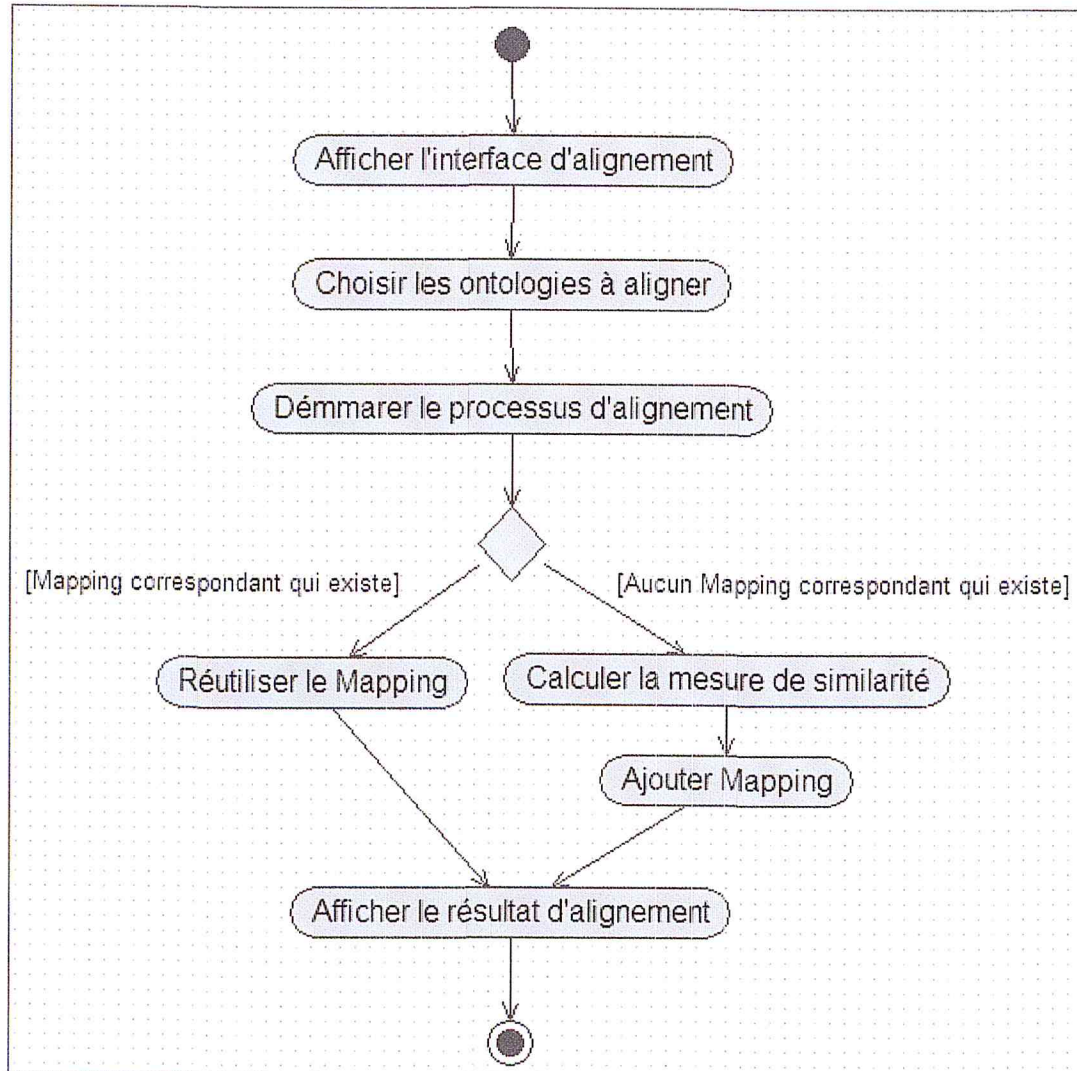


Figure 15: Diagramme d'activité du processus d'alignement d'ontologies

5. Conception

5.1. Conception générale :

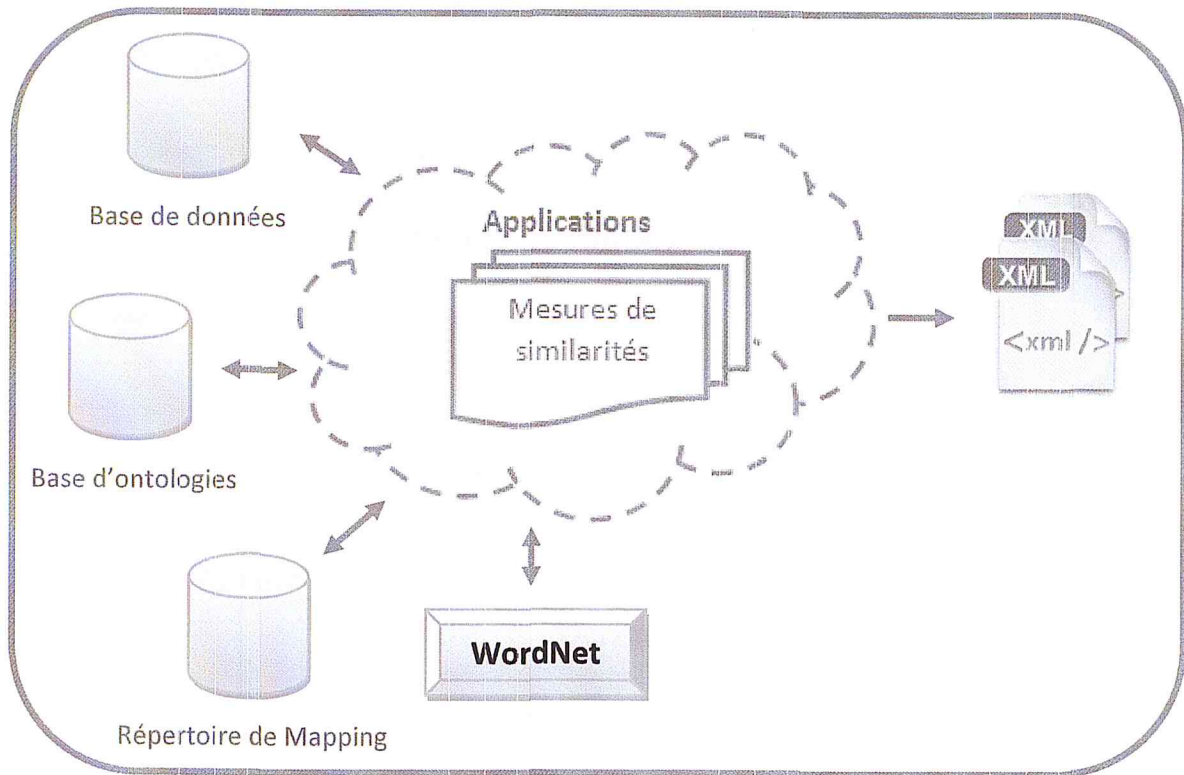


Figure 16: Architecture globale du système

Afin que notre projet soit bien structuré, le système se constitue de plusieurs modules qui permettront un bon fonctionnement du le système.

Le système utilise donc 3 bases et le dictionnaire lexical WordNet, le module qui effectue le processus d'alignement est donc celui des mesures de similarité, et nous allons avoir en sortie un alignement comme résultat, ce dernier est sauvegardé dans un fichier XML dont le schéma est le suivant :

Chapitre III : Conception du système de mapping d'ontologies

➤ Module de Répertoire de Mapping :

C'est la base qui contient tous les mapping que nous allons réutiliser pour l'alignement des ontologies.

➤ Module de WordNet :

C'est module externe WordNet (similarité lexicale), car nous ne sommes pas parvenus à implémenter les mesures de comparaison qui utilisent Wordnet. Ce qui nous empêche à l'intégrer directement dans notre application. En revanche, nous avons utilisé l'API JAWS qui nous permet de récupérer des données à partir de la base de données WordNet.

➤ Module de mesures de similarités :

Le module de mesure de similarité comporte toutes les techniques de mesure de similarité entre les concepts d'ontologies utilisées (voir section 5 « Conception détaillée », calcul de similarités).

À partir de tous ces modules, notre application est mise en œuvre de façon ergonomique, afin que l'IHM (Interface Homme Machine) soit facile à manipuler et à comprendre.

5.1.2. *Diagramme des paquetages*

Le Paquetage (ou « **package** » en anglais) : c'est le mécanisme général de regroupement d'éléments tels que les classes, les interfaces, mais aussi les acteurs, cas d'utilisation, etc.

Le diagramme de paquetage a la possibilité de ne pas représenter tous les éléments contenus, ce dernier est représenté en UML.

Détails des paquetages de notre système :

- ✓ Les ontologies : Ce package englobe les classes qui ont relation avec les ontologies, il contient :
 - Base d'ontologies : représente le regroupement des ontologies existantes dans notre système, dans des bases différentes.
 - Ontologies : contiennent toutes les ontologies qui existent dans toutes les bases d'ontologies de notre système.

Chapitre III : Conception du système de mapping d'ontologies

-Concept : Cette classe contient l'unité élémentaire des ontologies. Ou toutes les opérations se feront sur les concepts afin d'aligner des ontologies.

✓ Les matchers : Ce package représente toutes les fonctions utilisées pour calculer les mesures de similarité, il importe donc des informations (les concepts, les instances..) depuis les classes du paquetage des ontologies, il contient :

- Mesure de similarité : c'est la classe qui calcule la mesure de similarité globale dans notre système, en utilisant les mesures calculées par les autres classes filles (Jaro, Jaccard, Levenshtein, Wu et Palmer)
- Répertoire de mapping : c'est la classe qui contient tous les mapping valides, ces mappings seront réutilisé pour notre technique de réutilisation de mapping.
- WordNet : c'est un dictionnaire lexical utilisé pour le calcul de la mesure de similarité lexicale.

✓ Utilisateurs : C'est un package qui représente les utilisateurs (l'utilisateur normal, et l'administrateur), il a accès au package « les Matchers » et « les ontologies » car le package « les Matchers » peut importer les ontologies depuis le package « Les ontologies »

Le diagramme suivant représente les paquetages de notre système et les dépendances entre eux :

5.1.3. Architecture détaillé du système

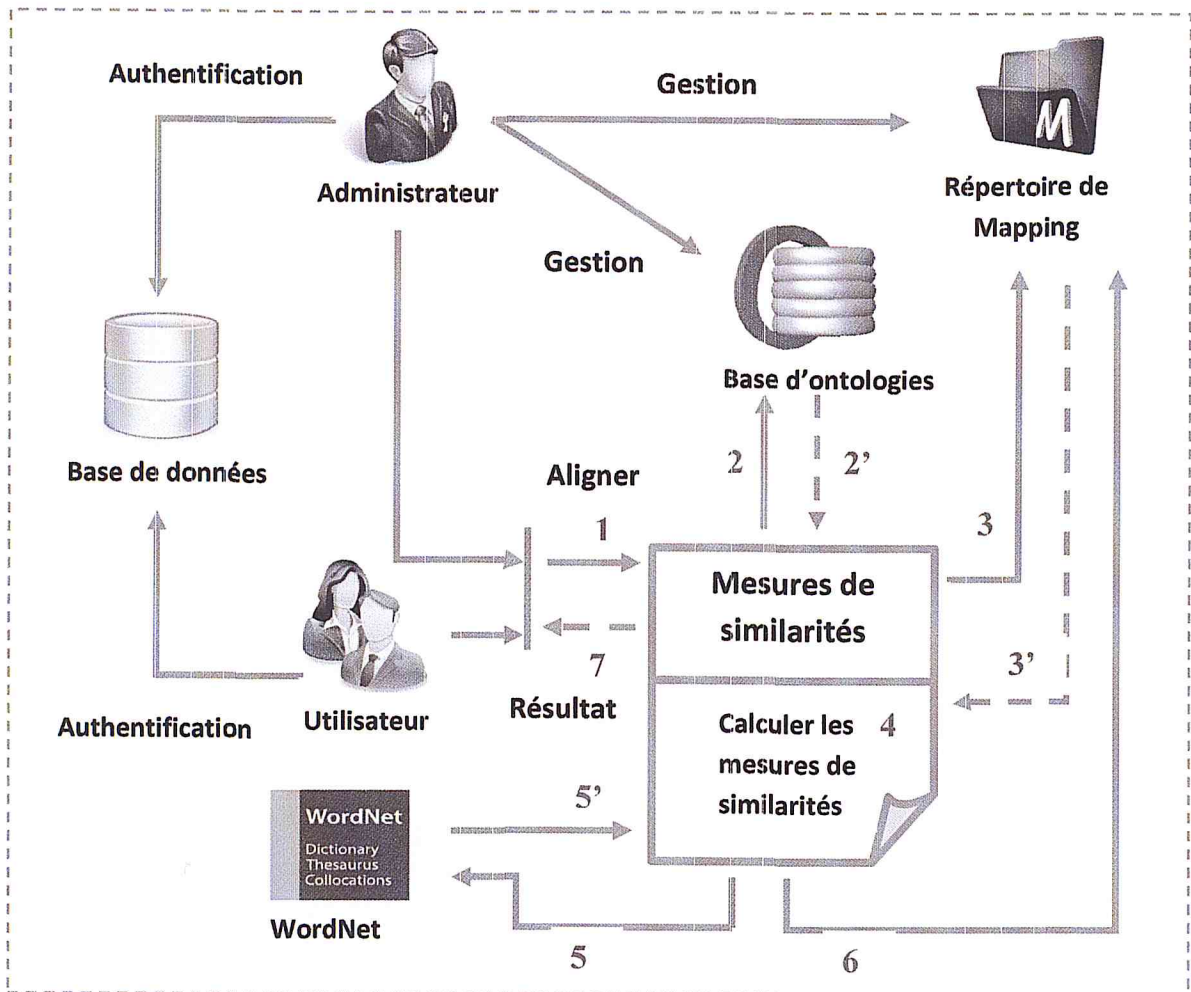


Figure 19: Architecture du système d'alignement en utilisant la réutilisation du mapping

Définition des éléments de l'architecture :

- Base de données : c'est la base qui contient les informations des utilisateurs afin qu'ils puissent s'authentifier.
- Utilisateur : c'est la personne qui utilise notre système pour l'alignement des ontologies.
- Administrateur : c'est la personne qui gère notre système (Répertoire de mapping, base d'ontologies ainsi que les utilisateurs etc.)
- Word Net : est une base de données lexicale développée par des linguistes. Son but est de répertorier, classifier et mettre en relation de diverses manières le contenu sémantique et lexical de la langue anglaise, nous utiliserons cette base pour chercher la synonymie des concepts.

Chapitre III : Conception du système de mapping d'ontologies

- Base d'ontologies : c'est la base qui contient les ontologies qu'un utilisateur va aligner
- Répertoire de Mapping : c'est la base qui contient les mapping qu'en va réutiliser pour l'alignement des ontologies.
- Mesure de similarités : c'est un module de calcul de mesure de similarité, il regroupe toutes les matchers utilisées pour calculer la mesure de similarité globale.

Numéro	Explication
1	L'administrateur et l'utilisateur peuvent lancer le processus d'alignement après leurs authentifications.
2	Le module mesure de similarité sélectionne deux ontologies à partir de la base d'ontologies, soit existante ou bien chargée dans la base d'ontologies par l'utilisateur.
2'	Chargement des deux ontologies à aligner.
3	Chercher un Mapping dans le répertoire de Mapping.
3'	Réponse de la recherche de Mapping correspondant.
4	Calculer les mesures de similarité.
5	Chercher les synonymes dans WordNet.
5'	Réponse de la recherche des synonymes.
6	Ajouter un Mapping au répertoire de Mapping.
7	Le système envoie le résultat d'alignement sous format XML.

Remarque :

- Répertoire de Mapping : c'est la base qui contient les mapping à réutiliser. Après l'alignement des ontologies établies par un utilisateur, les mapping vont être ajouté au répertoire de mapping s'ils ne sont pas déjà sauvegarder.
- Base d'ontologies : au fur est à mesure d'utilisation du système l'utilisateur remplit la base d'ontologie en les important à travers l'interface Ajouter une ontologie.

5.2. Conception détaillées

5.2.1. Diagramme de Classe

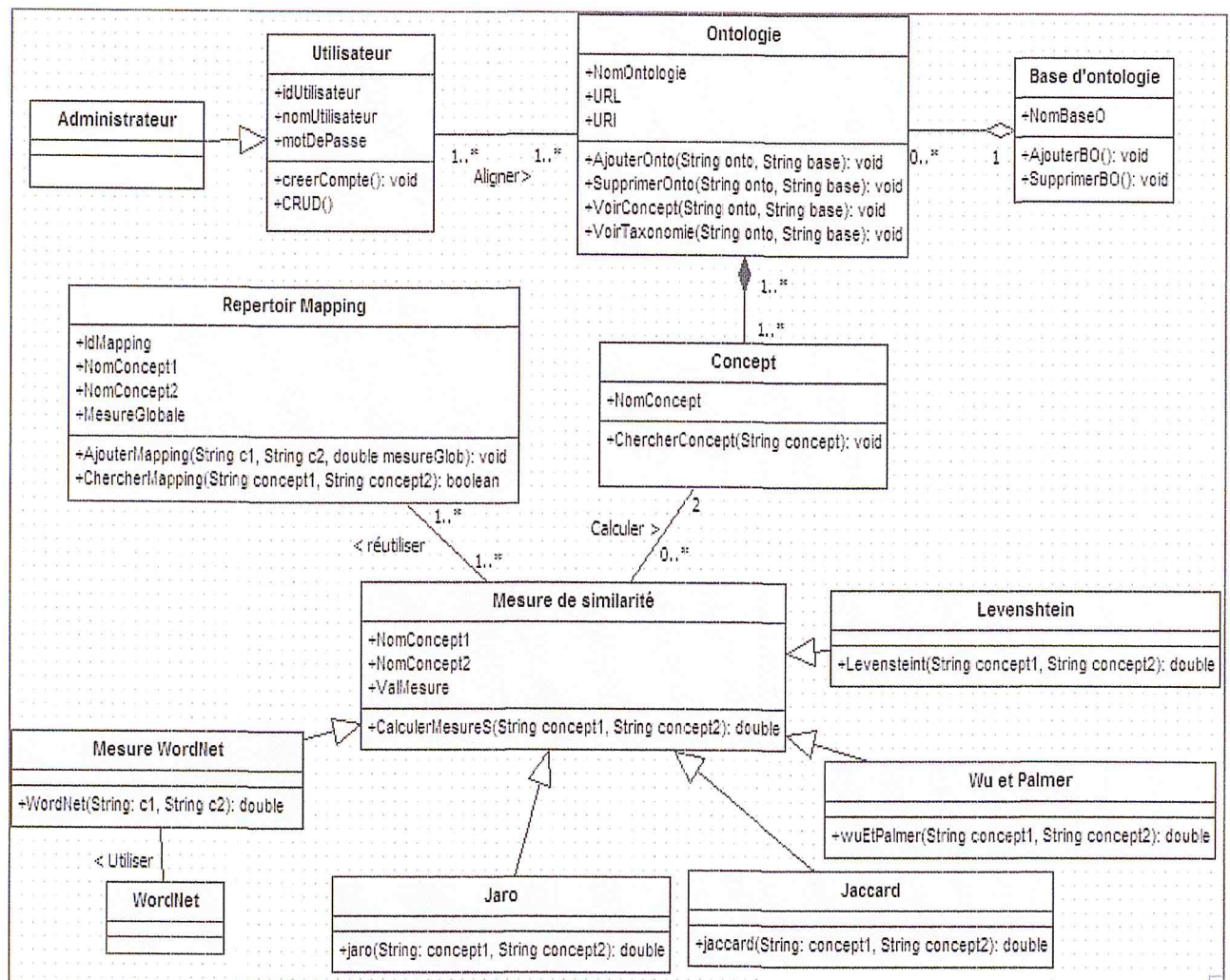


Figure 20: Diagramme de classe

- Description du diagramme de classe :

Classe	Attributs et types	Méthodes	Description des classes
Base d'ontologie	#NomBase: String	-AjouterBO(String base) : void (jOption pour afficher si la base est ajouté) -SupprimerBO(String base) : void (jOption pour afficher si la base est supprimé)	Elle représente l'agrégation des ontologies existantes
Ontologie	#NomOntologie : String	-AjouterOnto(String onto, String base):void	Elle représente l'unité de la base d'ontologie

Chapitre III : Conception du système de mapping d'ontologies

	+URL :String +URI:String	(jOption pour afficher si l'onto est ajouté) -SupprimerOnto(String onto, String base):void (jOption pour afficher si l'onto est supprimé) -VoirConcept(String onto,String base):void (afficher la liste des concepts) -VoirTaxonomie (String onto,String base) :void (afficher le schéma de la taxonomie)	représentée en OWL
Concept	#NomConcept: String	-ChercherConcept(String concept):void (afficher la liste des ontologies)	Elle représente l'unité élémentaire des ontologies. Ainsi que toutes les opérations se feront sur les concepts afin d'aligner des ontologies.
Mesure de similarité	#NomConcept1: String #NomConcept2: String +ValMesure : double	-CalculerMesureS(String concept1, String concept2):double	Elle regroupe les mesures utilisées dans le système afin de calculer les similarités selon les approches utilisées.
Répertoire de Mapping	#IdMapping :int +NomConcept1 : String +NomConcept2: String +MesureGlobale: double	-AjouterMapping(String concept1, String concept2,double mesureGlobale):void (jOption pour afficher si le mapping est ajouté) -ChercherMapping(String concept1, String concept2):boolean (pour voir s'il existe ou pas et l'afficher)	Elle regroupe tous les mapping raffinés que le système va réutiliser
Utilisateur	#IdUtilisateur : int +NomUtilisateur : String	CreerCompte() CRUD()	Elle représente les utilisateurs inscrits dans la base de données

Chapitre III : Conception du système de mapping d'ontologies

	+MotDePasse: String		
Administrateur	#IdUtilisateur : int NomUtilisateur: String MotDePasse: String	CreerCompte() CRUD()	C'est la classe fille de la classe utilisateur, où l'administrateur a plus de privilège.
Mesure WordNet		WordNet(String: ct1, String c2): double	Elle calcule la mesure entre deux concepts en utilisant l'outil WordNet.
WordNet			C'est la classe de similarité lexicale WordNet
Jaro		jaro(String: concept1, String concept2): double	Elle mesure le nombre et l'ordre des caractères communs entre deux chaînes de caractères.
Jaccard		jaccard(String: concept1, String concept2): double	Elle calcule le nombre des objets communs divisé par le nombre total des objets
Wu et Palmer		wuEtPalmer(String concept1, String concept2): double	Elle se sert de la structure hiérarchique de l'ontologie pour déterminer la similarité sémantique entre les concepts.
Levenshtein		Levenshtein(String concept1, String concept2): double	Elle mesure la similarité entre deux chaînes de caractères selon le nombre d'opérations nécessaires pour la transformation de chaîne.

5.2.2. Calcul de la similarité

- Définition de la mesure de similarité :

Une mesure de similarité S est une fonction $X \times X \rightarrow \mathbb{R}$ qui satisfait les propriétés suivantes:

- ✓ Positivité : $\forall x,y \in X, S(x,y) \geq 0$
- ✓ Symétrie : $\forall x,y \in X, S(x,y) = S(y,x)$
- ✓ Maximalité : $\forall x,y \in X, S(x,x) \geq S(x,y)$

D'autres propriétés peuvent être requises comme la normalisation qui impose que les valeurs appartiennent à l'intervalle $[0;1]$.

◆ Afin de calculer la similarité dans notre système nous avons utilisé 3 méthodes terminologique (parmi eux 2 syntaxique et 1 lexical) et une méthode structurelle.

- Mesure de similarité de Wu & Palmer (structurelle):

La mesure de similarité de [46] est basée sur le principe suivant :

Etant donnée une ontologie Ω formée par un ensemble de nœuds et un nœud racine R (Figure 22). Soit X et Y deux éléments de l'ontologie dont nous allons calculer la similarité. Le principe de calcul de similarité est basé sur les distances (N1 et N2) qui séparent les nœuds X et Y du nœud racine et la distance qui sépare le concept subsumant²(CS) de X et de Y du nœud R. La mesure de Wu et Palmer est définie par la formule suivante :

$$\text{Simc}(X,Y) = \frac{2*N}{N1+N2}$$

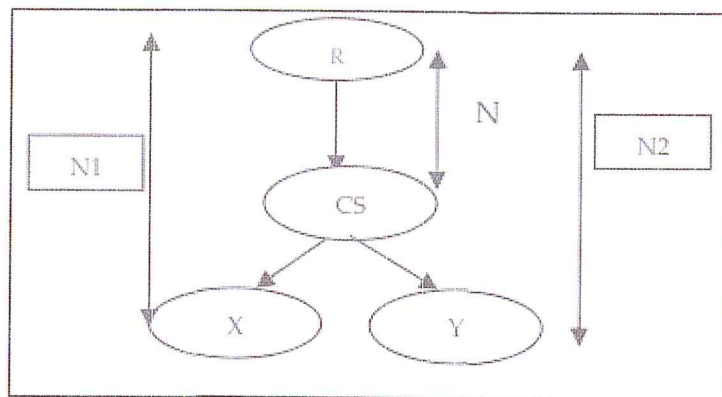


Figure 21: Exemple d'un extrait d'ontologie [46]

C'est la mesure la plus utilisée, car cette mesure a l'avantage d'être faciles à implémenter et peut donner une idée sur le lien sémantique entre les concepts.

² Le concept subsumant c'est le concept commun le plus spécifique.

Chapitre III : Conception du système de mapping d'ontologies

- Mesure de similarité de Jaccard (Syntaxique):

1- Pour les concepts :

La mesure de similarité de Jaccard est définie par le nombre des objets communs divisé par le nombre total des objets moins le nombre d'objets communs [47] :

$$\text{Sim}_j = \frac{x.y}{\|x\|_2^2 + \|y\|_2^2 - x.y}$$

Tels que x et y sont des vecteurs extraits à partir des concepts X et Y . $\|x\| = \Phi^x_{i=1}$ désigne la norme du vecteur x et $\|x\|_2 = \sqrt{\sum_{i=1}^n |x_i|^2}$

2- Pour les instances :

Pour comparer les ontologies à partir des instances associées aux concepts :

L'approche que nous avons utilisée dans notre travail s'inscrit dans la catégorie des approches à base d'instances lorsque les deux ontologies à aligner partagent beaucoup d'instances communes (même domaine d'application). Elle utilise des mesures telles que la métrique de JACCARD pour calculer les similarités entre les entités à aligner.

$$\text{Jac}(C_1, C_2) = \frac{|i_1 \cap i_2|}{|i_1 \cup i_2|}$$

Tel que : i est l'ensemble des instances qui sont annotées par le concept C_X .

- Mesure de similarité de Levenshtein (Syntaxique):

Hall et Dowling (1980) définissent cette distance comme étant la mesure la similarité entre deux chaînes de caractères. Elle est égale au nombre d'opérations nécessaires pour transformer une chaîne en une autre. Les opérations consistent en suppression, insertion ou remplacement.[48] Nous définissons la mesure de similarité entre deux mots w_1 et w_2 comme :

$$\text{Sim}_{ed}(w_1, w_2) = \frac{1}{1 + ed(w_1, w_2)}$$

où $\text{Sim}_{ed}(w_1, w_2)$ est la distance de Levenshtein entre les deux mots w .

- Mesure de similarité de Jaro (Syntaxique) :

Elle mesure le nombre et l'ordre des caractères communs entre deux chaînes de caractères. Nous définissons la distance de similarité de Jaro entre deux mots w_1, w_2 comme la distance Jaro entre les deux chaînes de caractères des deux mots. [48]

La distance de Jaro entre chaînes S_1 et S_2 est définie par :

$$d_j = \left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m} \right)$$

Où:

- $|S_i|$ est la longueur de la chaîne de caractères S_i ;
- m est le nombre de *caractères correspondants* (voir ci-dessous);
- t est le nombre de *transpositions* (voir ci-dessous).

Deux caractères identiques de S_1 et de S_2 sont considérés comme *correspondants* si leur éloignement (i.e. la différence entre leurs positions dans leurs chaînes respectives) ne dépasse pas :

$$\left\lfloor \frac{\max(|s_1|, |s_2|)}{2} \right\rfloor - 1$$

Le nombre de transpositions est obtenu en comparant le i -ème caractère *correspondant* de S_1 avec le i -ème caractère *correspondant* de S_2 . Le nombre de fois où ces caractères sont différents, divisé par deux, donne le nombre de *transpositions*.

- Utilisation de WordNet [49] (Lexicale) :

WordNet est une ressource lexicale de langue anglaise, qui regroupe des termes (noms, verbes, adjectifs et adverbes) en ensembles de synonymes appelés synsets. Un synset regroupe tous les termes dénotant un concept donné. Les synsets sont reliés entre eux par des relations sémantiques: relation de généralisation / spécialisation, relation composant/composé. Les techniques basées sur les chaînes de caractères ne sont pas suffisantes quand les concepts sont sémantiquement proches et quand leurs noms sont différents, l'interrogation d'une ressource linguistique telle que WordNet peut indiquer que les concepts sont similaires. Pour le calcul de la similarité linguistique la fonction $Syn(c)$ calcul l'ensemble des Synsets de WordNet du concept c ; soit $S = Syn(c1) \cap Syn(c2)$ l'ensemble des sens communs entre $c1$ et $c2$ à comparer, la cardinalité de S est :

$$\lambda(S) = |Syn(c1) \cap Syn(c2)| ;$$

Soit $\min(|Syn(c1)|, |Syn(c2)|)$ le minimum entre les cardinalité des deux ensembles $Syn(c1)$ et $Syn(c2)$ alors la similarité entre deux concepts $c1$ et $c2$ sera définie comme suit :

$$Sim_{ling} = \frac{\lambda(s)}{\min(|Syn(c1)|, |Syn(c2)|)}$$

Cette mesure retourne 1.0 si au moins $c1$ est le seul synonyme de $c2$ ou $c2$ est le seul synonyme de $c1$.

5.2.3. Normalisation des chaînes de caractères

Les résultats de la comparaison des chaînes de caractères seront améliorés si ces chaînes sont « nettoyées » ou traitées avant de les fournir aux formules calculant la similarité. Cette phase est appelée la phase de normalisation ou de normalisation textuelle, qui diffère de la normalisation des valeurs de similarité dans un intervalle de [0, 1]. Les différents types de normalisation textuelle sont ceux empruntés au domaine de traitement automatique de la langue naturelle (TALN) :

- Normalisation des caractères : ce type de normalisation convertit toutes les majuscules dans une chaîne de caractères en leurs formes minuscules ou vice-versa. Par exemple, la chaîne de caractères « VoitureS » sera convertie à « voitures » et ensuite, elle est considérée comme égale exactement à l'autre chaîne de caractères « voitures ».

-Normalisation des espaces : ce type de normalisation remplace toutes les séquences consécutives des espaces, des tabulations, des retours de chariot (les caractères CR) trouvées dans une chaîne de caractères par un seul caractère d'espace.

Par exemple, l'expression « ma voiture » est normalisée à « ma voiture ».

- Suppression des signes diacritiques ou des accents (aigus, graves...) : ce type de traitement remplace des caractères avec des signes diacritiques par caractères correspondants sans signes diacritiques. Par exemple, le mot « Hanoï » est remplacé par le mot « Hanoi » sans changer la signification du mot, la capitale du Vietnam. Cependant, certaines suppressions changeront la signification du terme : « là » (adverbe de lieu) et « la » (article).

-Suppression des chiffres. Par exemple, l'expression «voiture5 » est normalisée à «voiture».

5.2.4. Composition des différentes stratégies de calcul de similarité

Dans notre approche, nous avons utilisé la composition comme une méthode de combinaison des différentes stratégies de calcul de similarité. Elle permet de combiner les résultats produits par les différentes stratégies exécutées. Notre approche est parallèle, et fonctionne en deux étapes successives. La première étape permet de calculer les similarités (terminologiques et structurelles) à savoir que les différentes stratégies sont exécutées indépendamment, chaque stratégie produit des valeurs de similarités partielles. La deuxième étape permet de calculer la similarité globale, dite sémantique, comme dans ce qui suit:

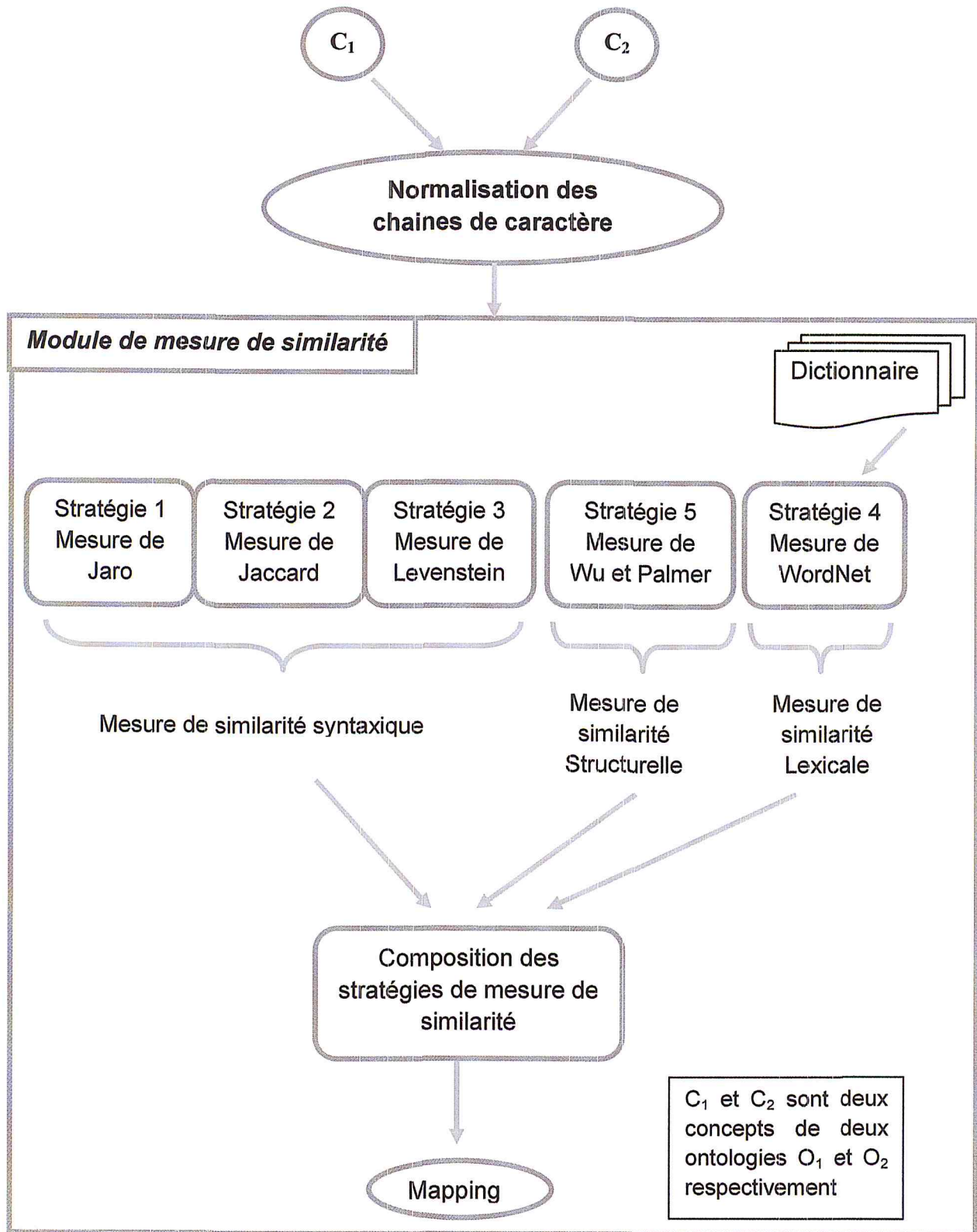


Figure 22: La combinaison des différentes stratégies.

- Pondération des similarités partielles

Les valeurs de similarités partielles obtenues précédemment, doivent être composées pour former la similarité globale entre concepts et elles sont pondérées par des coefficients. Le coefficient associé à la valeur de similarité de Jaro par exemple, est donnée par la formule suivante : $co_Jar = e^{simJar(C_1, C_2)}$.

Les autres coefficients se calculent de la même manière que le coefficient de la similarité de description.

- Calcul de la similarité globale

Pour calculer la mesure de similarité globale nous avons suivi les étapes suivantes :

➤ Si le mapping n'existe pas dans le répertoire de mapping :

- Si la mesure syntaxique entre les deux concepts est égale à 1 et les concepts n'ont pas d'instances alors :

SimGlob(C_1, C_2) =

$$\frac{(\mathit{simWp}(C_1, C_2).co.WP) + (\mathit{simLev}(C_1, C_2).co.Lev) + (\mathit{simJar}(C_1, C_2).co.Jar) + (\mathit{simJac}(C_1, C_2).co.Jac) + (\mathit{simWD}(C_1, C_2).co.WN)}{co.wp + co.Lev + co.jar + co.jac + co.wn}$$

- Si la mesure syntaxique entre les deux concepts est égale à 1 et les concepts ne sont pas dans le dictionnaire WordNet (WordNet ne connais pas les concepts) alors :

SimGlob(C_1, C_2) =

$$\frac{(\mathit{simWp}(C_1, C_2).co.WP) + (\mathit{simLev}(C_1, C_2).co.Lev) + (\mathit{simJar}(C_1, C_2).co.Jar) + (\mathit{simJac}(C_1, C_2).co.Jac) + (\mathit{simIn}(C_1, C_2).co.In)}{co.wp + co.Lev + co.jar + co.jac + co.In}$$

- Sinon La similarité globale est calculée par la fonction moyenne pondérée, Elle est définit comme suit :

$$\mathit{SimGlob}(C_1, C_2) = \frac{\sum(\mathit{sim}(C_1, C_2).co)}{\sum co} \quad (1)$$

Tel que :

$$\left\{ \begin{array}{l} \sum \mathit{sim}(C_1, C_2).co = (\mathit{simWp}(C_1, C_2) * co.WP) + (\mathit{simLev}(C_1, C_2) * co.lev) + (\mathit{simJar}(C_1, C_2) * \\ co.Jar) + (\mathit{simJac}(C_1, C_2) * co.Jac) + (\mathit{simWn}(C_1, C_2) * co.WN) + (\mathit{simIn}(C_1, C_2) * co.In) \\ \sum co = co.wp + co.Lev + co.jar + co.jac + co.wn + co.In \end{array} \right.$$

➤ Sinon :

$$\mathit{SimGlob}(C_1, C_2) = \frac{\mathit{ValMap}(C_1, C_2) * co.ValMap + \mathit{SimGlob}(C_1, C_2) * co.SimGlob}{co.ValMap + co.SimGlob} \quad (2)$$

Tel que:

$$\left\{ \begin{array}{l} \mathbf{ValMap}(C_1, C_2) = \mathbf{ValMap}(C_1, C_2) + \alpha \\ \quad \text{Si le mapping est valide } (\mathbf{ValMap}(C_1, C_2) \geq 0.75) \\ \mathbf{ValMap}(C_1, C_2) = \mathbf{ValMap}(C_1, C_2) - \alpha \\ \quad \text{Si } \mathbf{ValMap} < 0.75 \end{array} \right.$$

Où :

$\mathbf{ValMap}(C_1, C_2)$: c'est la valeur de similarité entre le concept C_1 et le concept C_2 sauvegardé dans le répertoire de mapping.

$\mathbf{SimGlob}(C_1, C_2)$: c'est la valeur de similarité calculé entre le concept C_1 et le concept C_2 par la formule (1)

α : c'est la valeur fixé par l'administrateur tel que : $0 < \alpha < 0.05$

La valeur α

Après avoir effectué plusieurs tests sur les résultats obtenus par notre système d'alignement, nous avons trouvé que les mapping qui ont une valeur de similarité globale supérieure ou égale à 0.75 sont des mapping valides et les mapping qui ne vérifient pas cette condition sont des mapping non valides.

Et depuis ces testes aussi, nous avons constaté que la valeur de similarité entre les mêmes concepts ne change pas ou change très très légèrement.

Alors nous avons opté à raffiner ces mapping avec notre formule proposée (2) pour avoir une valeur de similarité plus significative. Ce raffinement sera effectué à chaque fois qu'on réutilise le mapping. Après plusieurs itérations la valeur de similarité va stabiliser.

Dans cette formule nous avons utilisé une valeur α qui sera fixés par l'administrateur cette valeur va changer les similarités en des similarités plus significatives telles que:

- Les mapping valides qui n'ont pas encore atteint la mesure 1 vont s'approcher de la mesure 1.
- Les mapping non valides vont s'approcher de la mesure 0. Ces mapping ne sont pas égaux à 0 dès le début à cause des caractères qu'ils partagent mais pas dans le même ordre.

Cette valeur n'influe pas sur la validité des mapping, les mapping valides seront toujours valide et les mapping non valide seront aussi toujours non valides.

5.2.5. Comparaison entre notre système et les systèmes présentés (II.5.1) selon les techniques d'alignement de base utilisé

Techniques / Systèmes	Automation	String-based	Language-based	Constraint-based	Linguistic Resources	Alignment Reasoning	Graph-based	Taxonomy-based	Repository of Structures	Model-based
COMA	Full / Semi	✓	✓	✓	✓	✓	✓	-	-	-
COMA++	Full / Semi	✓	✓	✓	✓	✓	✓	-	-	-
S-Match	Full	✓	✓	-	✓	-	-	✓	-	Semantic Reasoning
NOM	Full	✓	-	✓	✓	-	-	✓	✓	-
QOM	Full	✓	-	✓	✓	-	-	✓	-	-
GLUE	Semi	-	-	-	-	-	-	Probabilistic Reasoning	-	-
SMTO	Semi	✓	-	-	✓	-	✓	✓	-	-
C.R. Rene Robin & G.V. Uma	Full	✓	✓	-	✓	-	✓	-	-	-
Notre système	Full	✓	✓	-	✓	✓	✓	✓	-	-

Tableau 4: Comparaison entre notre système et les systèmes présentés (II.5.1) selon les techniques d'alignement de base utilisées

Remarque :

- Notre système utilise aussi une technique à base d'instance

5.2.6. Comparaison entre notre système et les systèmes présentés dans (II.5.1) selon quelques caractéristiques externe

Systemes Caractéristiques	COMA	COMA++	Match	NOM	QOM	GLUE	SMTO	C.R. Rene Robin & G.V. Uma	Notre système
Entrées	Schéma relationnel, schéma XML, OWL	Schéma relationnel, schéma XML, OWL	Classification, schéma XML, OWL	RDF, OWL	RDF, OWL	Schéma relationnel, Schéma XML, Taxonomie	Schéma relationnel, schéma XML, OWL, RDF	OWL	OWL
Sorties	Alignement	Alignement	Alignement	Alignement	Alignement	Alignement	Intégration	Fusion	Alignement
Besoins et Opérations	Utilisateur: Traduction de données	Utilisateur: Traduction de données	Auto	Auto, Instances (opt.)	Auto, Instances (opt.)	Auto, Instances Formation	Utilisateur: Confirmation des similarités complexes	Auto	Auto
Relation	↔	↔	↔	-	-	-	-	-	↔
Cardinalité	(0,1)-(0,1), (0,1)-(0,n), (0,n)-(0,n)	(0,1)-(0,1), (0,1)-(0,n), (0,n)-(0,n)	-	-	(0,1)-(0,1)	-	-	-	(0,1)-(0,1), (0,1)-(0,n)
Domaine d'application	Générique	Générique	Générique	-	-	-	Générique	-	Générique

Tableau 5: Comparaison entre notre système et les systèmes présentés dans (II.5.1) selon quelques caractéristiques externe

6. Conclusion

Après avoir spécifié et analysé les besoins du système, nous avons présenté la mise en œuvre de son architecture ainsi que la conception globale et détaillée, afin de pouvoir construire notre système de façon objective et organisée. Nous allons entamer donc dans le chapitre suivant l'implémentation de notre système.

CHAPITRE IV :

Implémentation du système

1. Introduction

Implémenter c'est « effectuer l'ensemble des opérations qui permettent de définir un projet et de le réaliser, de l'analyse du besoin à l'installation et la mise en service du système ou du produit »³,

Dans ce chapitre nous allons décrire de façon visuelle l'implémentation de notre système, en effectuant des captures d'écran des différentes interfaces du système.

2. Outils de développement de notre système

➤ NetBeans et JDK

Lors d'un développement d'application ou de la création d'un site web, il est plus qu'indispensable d'avoir un outil comme **NetBeans IDE** pour simplifier la tâche. NetBeans IDE « intégré » (en anglais, pour *Integrated Development Environment*, le produit d'un outil peut servir de matière première pour un autre outil [50]) n'est autre qu'un outil de développement professionnel. Il offre de nombreuses fonctionnalités pratiques. Ainsi qu'un environnement «Java Development Kit » JDK est requis pour les développements en Java de notre système.

➤ SGBDR « «Postgres 9.4 »

Postgres est un **SGBDR** (*Système de gestion de bases de données relationnelles*) où l'une des principales qualités est d'être un logiciel libre, c'est-à-dire gratuit et dont les sources sont disponibles.

➤ JDBC

JDBC (*Java DataBase Connectivity*) est une interface de programmation pour les programmes utilisant la plateforme Java. Elle permet à notre système d'accéder par le biais d'une interface commune, qui est Postgres dans notre cas, à des sources de données relationnelle.

➤ OWL2Prefuse

OWL2Prefuse est un package Java qui crée des graphes et des arbres Prefuse à partir de fichiers OWL. Il prend soin de convertir la structure de données OWL en une structure de données Prefuse.

³ La Commission générale de terminologie et de néologie 2007

➤ JAWS

JAWS (*Java API for WordNet Searching*) comme son nom l'indique c'est un API⁴ Java qui permet aux applications Java de récupérer des données à partir de la base de données WordNet.

➤ JENA

Jena est une API Java qui peut être utilisée pour créer et manipuler des graphes RDF et les ontologies exprimé en OWL. Jena possède des classes pour représenter des graphes, des ressources, des propriétés et des littéraux. Dans Jena, un graphe est appelé un modèle et est représenté par l'interface Model.

➤ JDOM

JDOM est une *API* open source dont le but est de représenter et manipuler un document XML en Java.

➤ WordNet 2.1

WordNet est une base de données lexicale. Son but est de répertorier, classier et mettre en relation de diverses manières le contenu sémantique et lexical de la langue anglaise.

3. La mise en œuvre du système

Notre système contient deux genres d'interface globale, une pour l'utilisateur général et l'autre pour l'utilisateur administrateur.

Afin qu'un utilisateur puisse accéder à son interface, il doit passer par l'interface d'authentification.

⁴ Une interface de programmation

3.1. L'authentification

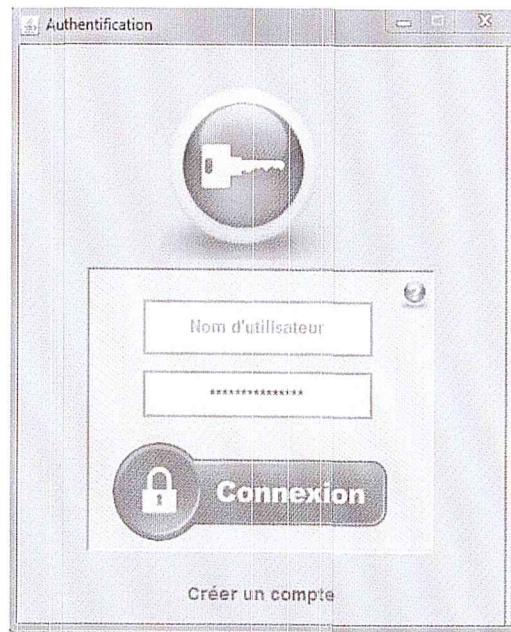


Figure 23: Interface « Authentification »

C'est l'interface qui permet à un utilisateur de s'authentifier avec un nom d'utilisateur, et un mot de passe.

- S'il s'agit d'un nouveau utilisateur il doit créer un compte en cliquant sur créer un compte en bas de l'interface.

3.2. L'inscription

Figure 24: Interface « inscription »

Chaque utilisateur qui veut utiliser notre système doit passer par l'étape d'inscription :

- L'utilisateur doit remplir tous les champs de l'interface inscription.
- Les valeurs saisies doivent être cohérentes et personnelles.
- L'utilisateur clique sur le bouton S'inscrire afin d'envoyer son formulaire au système.

3.3. Les interfaces de l'administrateur

◆ L'interface globale de l'administrateur



Figure 25: Interface « Administrateur »

L'administrateur a 5 boutons dans son interface :

4. Base d'ontologies : Il permettra à l'administrateur de consulter les bases d'ontologie existante.
5. Alignement : Il permettra à l'administrateur d'aligner deux ontologies à la fois.
6. Répertoire de Mapping : L'administrateur pourra consulter les Mapping sauvegardé.
7. Gestion d'utilisateurs : L'administrateur pourra gérer les utilisateurs inscrit dans notre système.
8. Applications : L'administrateur peut accéder aux applications proposées par notre système.

◆ L'interface « Répertoire de Mapping »



The screenshot shows a window titled 'Répertoire de mappings'. It contains a table with three columns: 'Concept1', 'Concept2', and 'Valeur de similarité'. The table lists various mappings between concepts and their similarity values. To the right of the table is a 'Détails' section with a button labeled 'Afficher'.

Concept1	Concept2	Valeur de similarité
MastersThesis	Unpublished	0.247
Unpublished	InBook	0.168
List	LectureNotes	0.187
PersonList	PersonList	1.0
PersonList	Person	0.451
PersonList	LectureNotes	0.315
Chapter	Deliverable	0.182
Book	Publisher	0.161
Book	Proceedings	0.168
Book	Manual	0.297
Misc	PhdThesis	0.049
Misc	Misc	1.0
Misc	Collection	0.143
Misc	Informal	0.153
Misc	Institution	0.175
Misc	PageRange	0.013
Misc	Report	0.022
Misc	Booklet	0.018
Misc	Book	0.034
Report	InCollection	0.19
Report	Book	0.183
Manual	MastersThesis	0.158
Deliverable	PhdThesis	0.2
Deliverable	Misc	0.125
Deliverable	Conference	0.208
Monograph	Proceedings	0.257
School	MotionPicture	0.157
PageRange	Unpublished	0.162
PageRange	MastersThesis	0.209
Person	Monograph	0.234
Person	Conference	0.234
Conference	InCollection	0.316
Conference	LectureNotes	0.354
Proceedings	Deliverable	0.217
Proceedings	Date	0.22
Collection	Proceedings	0.342
Institution	Publisher	0.333
Institution	Informal	0.216
TechReport	Article	0.341
TechReport	Conference	0.219
Reference	LectureNotes	0.339

Figure 26: Interface « Répertoire de Mapping »

- L'administrateur peut voir les données sauvegarder dans répertoire de Mapping.
- En cliquant sur Afficher dans la section Détails, l'administrateur peut afficher les détails des mesures de similarités.

◆ L'interface « Gestion d'utilisateur »

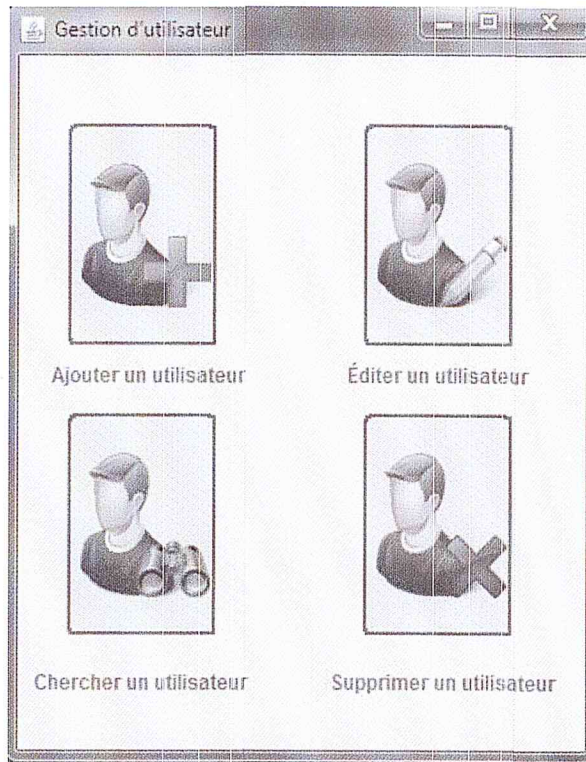


Figure 27: Interface « Gestion d'utilisateur »

L'administrateur peut gérer les utilisateurs comme suit :

- Ajouter un utilisateur.
- Éditer un utilisateur.
- Cherche un utilisateur s'il existe.
- Supprimer un utilisateur.

◆ L'interface « Applications »

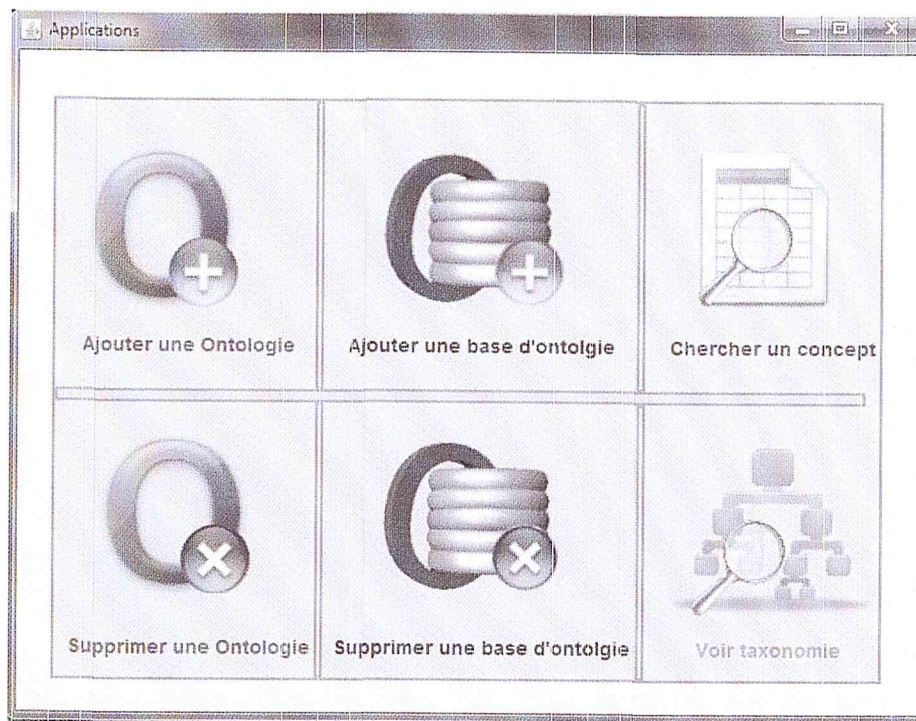
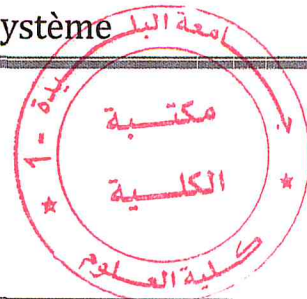


Figure 28: Interface « Applications »

L'administrateur à une interface Applications contenant différentes tâches à réaliser :

- Ajouter une ontologie à une base d'ontologie existante.
- Supprimer une ontologie existante.
- Ajouter une base d'ontologie.
- Supprimer une base d'ontologie.
- Chercher un concept. (l'interface est décrite dans la partie de l'utilisateur)
- Voir Taxonomie. (l'interface est décrite dans la partie de l'utilisateur)



3.4. Les interfaces de l'utilisateur

◆ L'interface globale de l'utilisateur

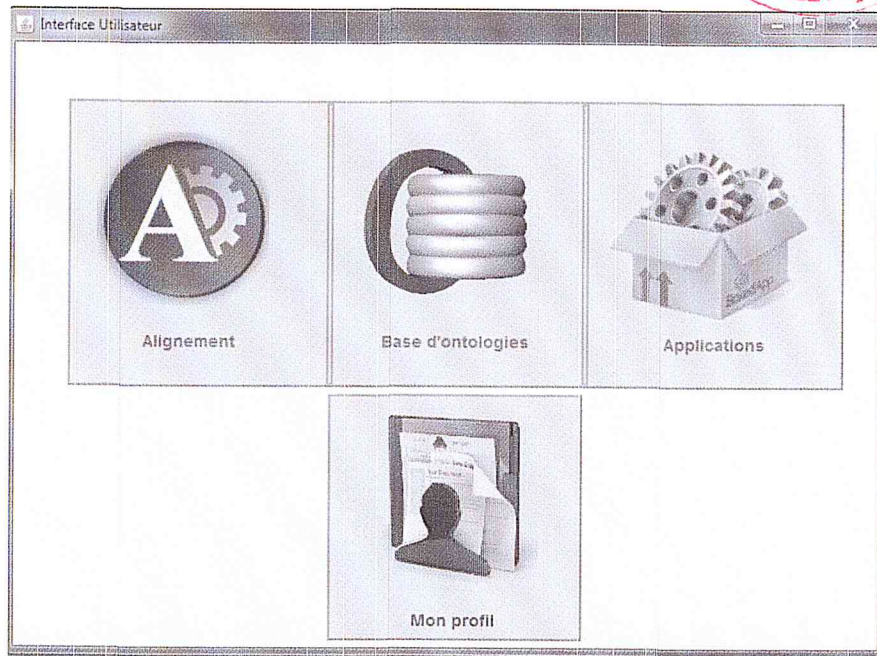


Figure 29: Interface « Utilisateur »

L'utilisateur a 4 boutons pour 4 interfaces :

- Alignement : Elle permettra à l'utilisateur d'aligner deux ontologies.
- Base d'ontologies : Elle lui permettra de consulter les bases d'ontologies existantes.
- Applications : Elle contient différentes applications proposées par le système.
- Mon Profil : Lui permettra de modifier son mot de passe ou l'adresse e-mail.

◆ Interface d'alignement :

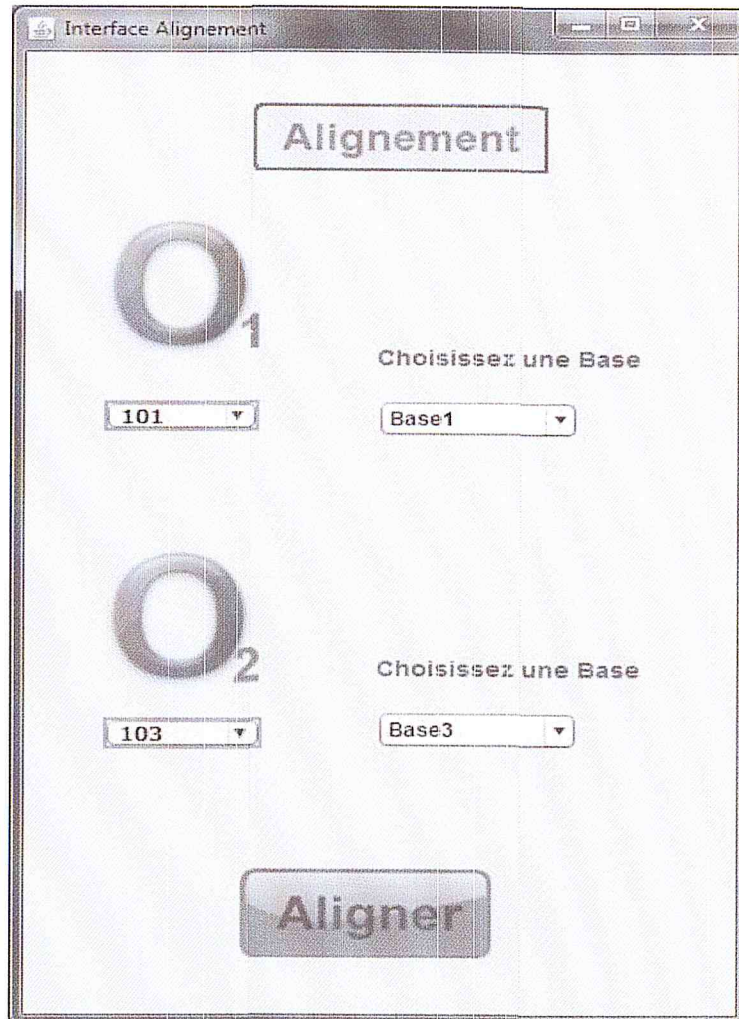
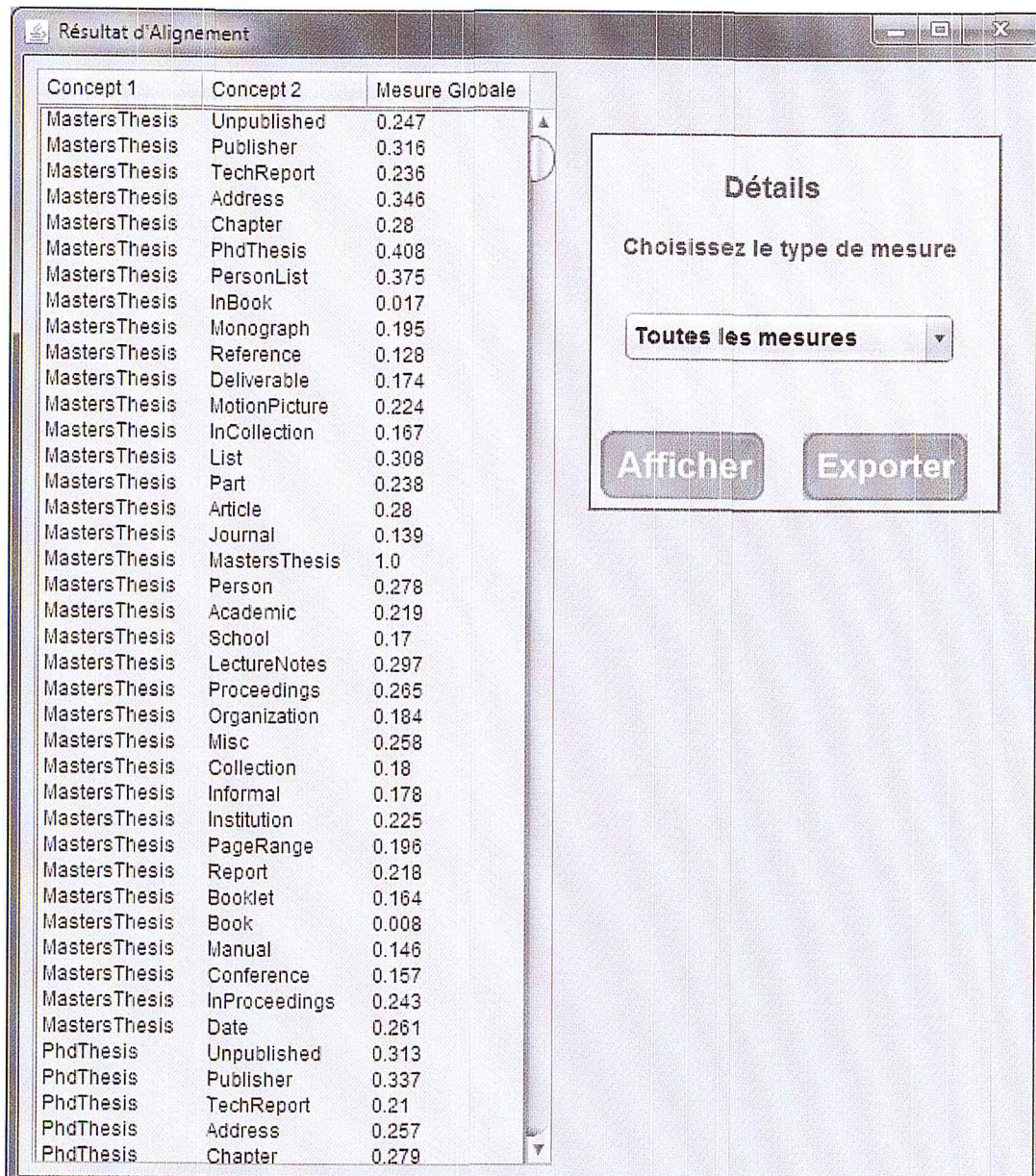


Figure 30: Interface « Alignement »

Afin qu'un utilisateur puisse aligner deux ontologies, il doit faire ce qui suit :

- Choisir la base où se situe l'ontologie 1 puis il choisit l'ontologie à sélectionner.
- Choisir la base où se situe l'ontologie 2 puis il choisit l'ontologie à sélectionner.
- Cliquer sur le bouton Aligner.

L'interface qui s'affichera est la suivante :



The screenshot shows a window titled 'Résultat d'Alignement'. It contains a table with three columns: 'Concept 1', 'Concept 2', and 'Mesure Globale'. The table lists various alignments between 'MastersThesis' and 'PhdThesis' concepts and their corresponding global measures. To the right of the table is a 'Détails' panel with a dropdown menu for 'Choisissez le type de mesure' (currently set to 'Toutes les mesures') and two buttons: 'Afficher' and 'Exporter'.

Concept 1	Concept 2	Mesure Globale
MastersThesis	Unpublished	0.247
MastersThesis	Publisher	0.316
MastersThesis	TechReport	0.236
MastersThesis	Address	0.346
MastersThesis	Chapter	0.28
MastersThesis	PhdThesis	0.408
MastersThesis	PersonList	0.375
MastersThesis	InBook	0.017
MastersThesis	Monograph	0.195
MastersThesis	Reference	0.128
MastersThesis	Deliverable	0.174
MastersThesis	MotionPicture	0.224
MastersThesis	InCollection	0.167
MastersThesis	List	0.308
MastersThesis	Part	0.238
MastersThesis	Article	0.28
MastersThesis	Journal	0.139
MastersThesis	MastersThesis	1.0
MastersThesis	Person	0.278
MastersThesis	Academic	0.219
MastersThesis	School	0.17
MastersThesis	LectureNotes	0.297
MastersThesis	Proceedings	0.265
MastersThesis	Organization	0.184
MastersThesis	Misc	0.258
MastersThesis	Collection	0.18
MastersThesis	Informal	0.178
MastersThesis	Institution	0.225
MastersThesis	PageRange	0.196
MastersThesis	Report	0.218
MastersThesis	Booklet	0.164
MastersThesis	Book	0.008
MastersThesis	Manual	0.146
MastersThesis	Conference	0.157
MastersThesis	InProceedings	0.243
MastersThesis	Date	0.261
PhdThesis	Unpublished	0.313
PhdThesis	Publisher	0.337
PhdThesis	TechReport	0.21
PhdThesis	Address	0.257
PhdThesis	Chapter	0.279

Figure 31: Interface du résultat d'alignement

Le système lui affichera le résultat de l'alignement des deux ontologies choisies :

- L'utilisateur peut voir les détails des mesures de similarité en choisissant toutes les mesures ou bien une mesure spécifique dans la liste de type de mesure, puis il cliquera sur Afficher.

- L'utilisateur peut exporter le résultat de l'alignement dans un fichier XML.

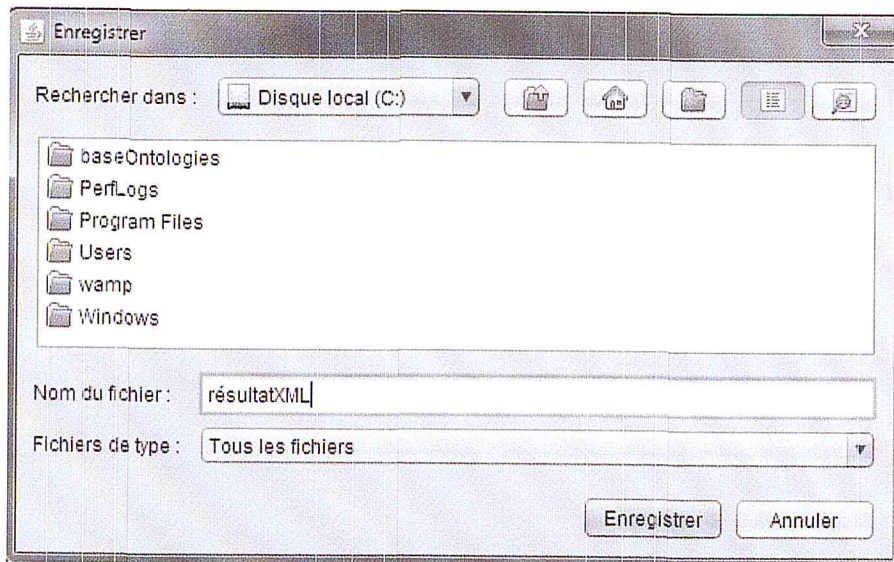


Figure 32: Interface de l'exportation du résultat d'alignement

Remarque :

- 1- L'affichage des résultats dans le fichier XML est organisé de façon que les mapping valide soient au début du fichier, et les mapping non valide soient juste après.
- 2- Le résultat du fichier est exprimé en XML Schema qui est publié comme recommandation par le W3C, c'est un langage de description de format de document XML permettant de définir la structure et le type de contenu d'un document XML. Un schéma d'un document définit:
 - Les éléments possibles dans le document.
 - Les attributs associés à ces éléments.
 - La structure du document et les types de données
- 3- Présente de nombreux avantages :
 - Structures de données avec types de données
 - Extensibilité par héritage.

Chapitre IV : Implémentation du système

```
RésultatXml.xml
1  <?xml version="1.0" encoding="ISO-8859-1"?>
2  <Mapping>
3  <Correspondance>
4    <concept1>Misc</concept1>
5    <concept2>Misc</concept2>
6    <meusre>1.00000</meusre>
7  </Correspondance>
8  <Correspondance>
9    <concept1>School</concept1>
10   <concept2>School</concept2>
11   <meusre>1.00000</meusre>
12 </Correspondance>
13 <Correspondance>
14   <concept1>MotionPicture</concept1>
15   <concept2>MotionPicture</concept2>
16   <meusre>1.00000</meusre>
17 </Correspondance>
18 <Correspondance>
19   <concept1>PageRange</concept1>
20   <concept2>PageRange</concept2>
21   <meusre>1.00000</meusre>
22 </Correspondance>
23 <Correspondance>
24   <concept1>LectureNotes</concept1>
25   <concept2>LectureNotes</concept2>
26   <meusre>1.00000</meusre>
27 </Correspondance>
28 <Correspondance>
29   <concept1>Academic</concept1>
30   <concept2>Academic</concept2>
31   <meusre>1.00000</meusre>
32 </Correspondance>
33 <Correspondance>
34   <concept1>Report</concept1>
35   <concept2>Report</concept2>
36   <meusre>1.00000</meusre>
37 </Correspondance>
38 <Correspondance>
39   <concept1>
40   <concept2>
41   <meusre>
42 </Correspondance>
43 </Mapping>
```

en length : 163292 lines : 6484 Ln : 44 Col : 28 Sel : 0 | 0

Figure 33: Le fichier exporté « RésultatXML »

◆ L'interface « Chercher un Concept »

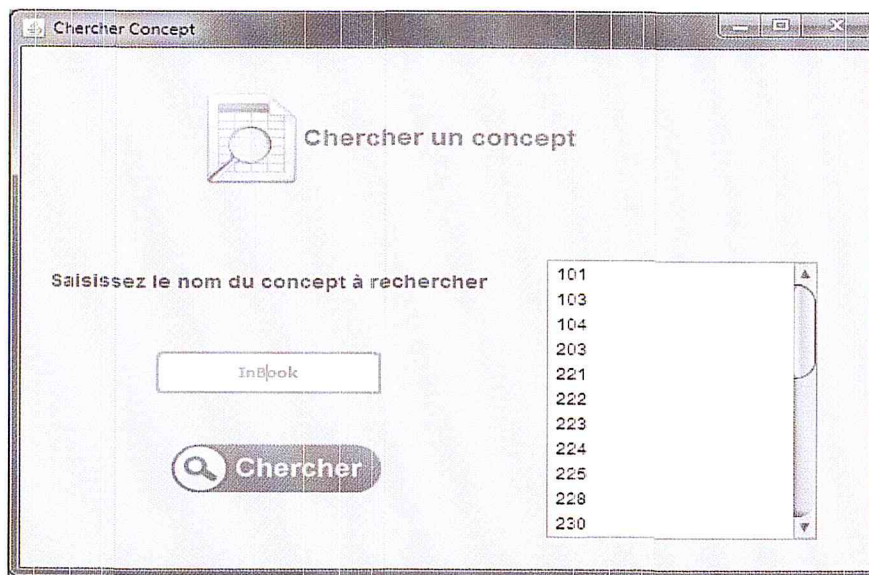


Figure 34: Interface « Chercher un Concept »

- L'utilisateur saisit le nom du concept, et clique sur le bouton chercher.
- Le résultat sera affiché à droite dans une liste, le système lui affichera toutes les ontologies contenant le concept saisi.

◆ Interface Voir Taxonomie

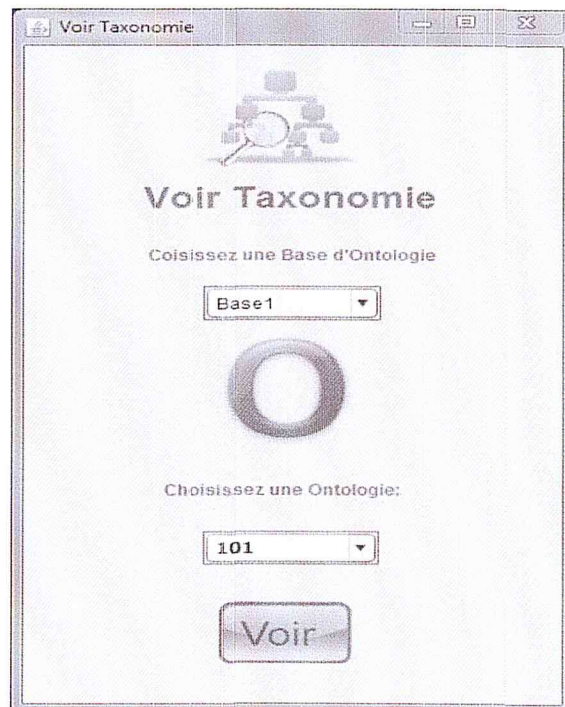


Figure 35: Interface Voir Taxonomie

Chapitre IV : Implémentation du système

L'utilisateur peut voir une taxonomie d'une ontologie en suivant ces étapes :

- Choisir une base d'ontologie contenant l'ontologie voulue.
- Choisir une ontologie.
- Cliquez sur le bouton voir.

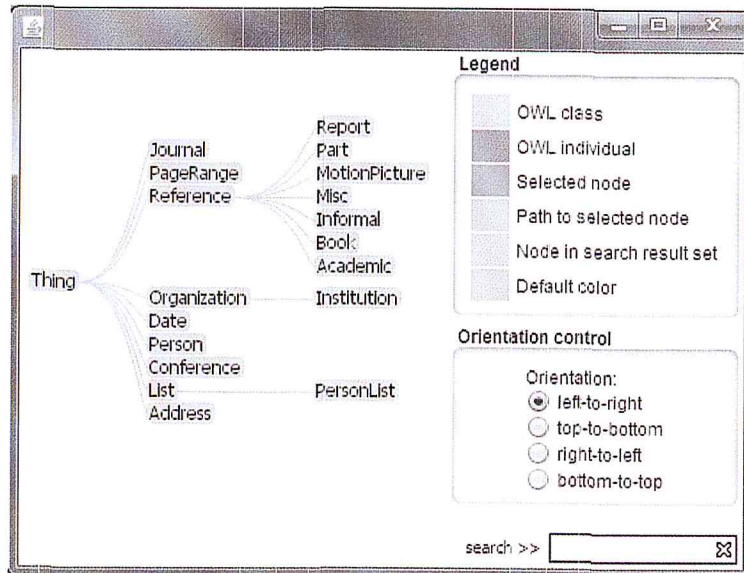


Figure 36: Résultat de Voir une Taxonomie

4. Conclusion

Après avoir implémenté notre système, et mis en œuvre toutes les interfaces requises, nous allons entamer dans le chapitre suivant, le teste et la validation des résultats de notre système.

CHAPITRE V :

Test et Validation

Chapitre V : Test et Validation

1. Introduction

Dans cette dernière partie nous allons effectuer un test sur notre système implémenté, ainsi que sa validation pour pouvoir dire que nous avons atteint le but de notre projet. Afin d'évaluer l'alignement des ontologies s'il est pertinent, en utilisant les mesures de similarités sélectionnées tout le long de notre projet et la technique de réutilisation de mapping.

2. L'évaluation du système

La méthode que nous avons utilisée pour évaluer notre système c'est l'utilisation du corpus de test de la campagne d'évaluation des systèmes d'alignements OAEI 2013. Ce corpus offre la possibilité d'avoir des alignements prêts provenant de plusieurs outils d'alignement.

Dans cette évaluation nous avons utilisé les ontologies 101 et 103 fournis par l'OAEI, nous allons donc comparer les résultats donnés par la campagne et ceux effectués par notre système.

- ✓ Le résultat suivant est l'alignement des deux ontologies 101 et 103 de la campagne OAEI :

Source: <http://oaei.ontologymatching.org/tests/101/onto.rdf>

Target: <http://oaei.ontologymatching.org/tests/103/onto.rdf> Correspondences

Unpublished = Unpublished 1.0

MastersThesis = MastersThesis 1.0

PhdThesis = PhdThesis 1.0

proceedings = proceedings 1.0

PersonList = PersonList 1.0

Address = Address 1.0

Chapter = Chapter 1.0

InBook = InBook 1.0

Date = Date 1.0

PageRange = PageRange 1.0

Booklet = Booklet 1.0

LectureNotes = LectureNotes 1.0

TechReport = TechReport 1.0

organization = organization 1.0

school = school 1.0

Misc = Misc 1.0

Collection = Collection 1.0

Academic = Academic 1.0

MotionPicture = MotionPicture 1.0

Article = Article 1.0

Informal = Informal 1.0

Book = Book 1.0

School = School 1.0

Proceedings = Proceedings 1.0

Reference = Reference 1.0

InCollection = InCollection 1.0

Report = Report 1.0

Conference = Conference 1.0

Part = Part 1.0

Publisher = Publisher 1.0

Manual = Manual 1.0

InProceedings = InProceedings 1.0

Journal = Journal 1.0

Monograph = Monograph 1.0

Institution = Institution 1.0

Deliverable = Deliverable 1.0

Chapitre V : Test et Validation

- ✓ Le résultat suivant est l'alignement des deux ontologies 101 et 103 de notre système de mapping d'ontologie :

Concept 1	Concept 2	Mesure
MastersThesis	MastersThesis	1.0
PhdThesis	PhdThesis	1.0
Unpublished	Unpublished	1.0
List	List	1.0
PersonList	PersonList	1.0
Chapter	Chapter	1.0
Book	Book	1.0
Misc	Misc	1.0
Date	Date	1.0
Report	Report	1.0
Manual	Manual	1.0
Deliverable	Deliverable	1.0
Monograph	Monograph	1.0
School	School	1.0
PageRange	PageRange	1.0
Person	Person	1.0
Conference	Conference	1.0
Proceedings	Proceedings	1.0
Collection	Collection	1.0
Institution	Institution	1.0
TechReport	TechReport	1.0
Reference	Reference	1.0
Journal	Journal	1.0
Organization	Organization	1.0
Article	Article	1.0
InProceedings	InProceedings	1.0
Informal	Informal	1.0
Part	Part	1.0
InCollection	InCollection	1.0
Academic	Academic	1.0
Address	Address	1.0
Publisher	Publisher	1.0
LectureNotes	LectureNotes	1.0
MotionPicture	MotionPicture	1.0
InBook	InBook	1.0
Booklet	Booklet	1.0

3. Qualité d'alignement d'ontologies

Les mesures de Précision, Rappel et Fallout [51] ont été des métriques largement exploitées pour estimer la qualité des alignements obtenus. Le EON3 "Evaluation of Ontology-based Tools" [52][53][54] retient ces mesures pour l'évaluation de la qualité de l'alignement. L'objectif principal de ces mesures est l'automatisation du processus de comparaison des méthodes d'alignement ainsi que l'évaluation de la qualité des alignements produits. La première phase dans le processus d'évaluation de la qualité de l'alignement consiste à résoudre le problème manuellement. Le résultat obtenu manuellement est considéré comme l'alignement de référence.

La comparaison du résultat de l'alignement de référence avec celui de l'appariement obtenu par la méthode d'alignement produit trois ensembles : N_{found} , $N_{expected}$ et $N_{correct}$. L'ensemble N_{found} représente les paires alignées avec la méthode d'alignement. L'ensemble $N_{expected}$ désigne l'ensemble des couples appariés dans l'alignement de référence. L'ensemble $N_{correct}$ est l'intersection des deux ensembles N_{found} et $N_{expected}$. Il représente l'ensemble des paires appartenant à la fois à l'alignement obtenu et l'alignement de référence.

La précision est le rapport du nombre de paires pertinentes trouvées, i.e., " $N_{correct}$ ", rapporté au nombre total de paires, i.e., " N_{found} ". Il renvoie ainsi, la partie des vraies correspondances parmi celles trouvées. Ainsi, la fonction précision est définie par : $Précision = \frac{|N_{correct}|}{|N_{found}|}$. Le rappel est le rapport du nombre de paires pertinentes trouvées, " $N_{correct}$ ", rapporté au nombre total de paires pertinentes, " $N_{expected}$ ". Il spécifie ainsi, la part des vraies correspondances trouvées. La fonction rappel est définie par : $Rappel = \frac{|N_{expected}|}{|N_{correct}|}$. La mesure Fallout permet d'estimer le pourcentage d'erreurs obtenu au cours du processus d'alignement. Elle est définie par le rapport des paires erronées, " $(N_{found} - N_{correct})$ ", rapporté au nombre total des paires trouvées, " N_{found} ", i.e.,

$$Fallout = \frac{|N_{found}| - |N_{correct}|}{|N_{found}|}$$

Chapitre V : Test et Validation

4. Tests

Le tableau suivant représente la comparaison des correspondances trouvées de façon manuelle et celle trouvée avec notre système d'alignement de deux ontologies sans la réutilisation de mapping :

- Cet alignement représente un alignement de référence.
- le tableau suivant est juste une partie du résultat d'alignement avec notre système.

Concept 1	Concept 2	Correspondances Manuelles	Mesure de similarité	Correspondances de notre système
MastersThesis	Unpublished	✗	0.252	✗
MastersThesis	Publisher	✗	0.321	✗
MastersThesis	TechReport	✗	0.241	✗
MastersThesis	Address	✗	0.351	✗
MastersThesis	Chapter	✗	0.285	✗
MastersThesis	PhdThesis	✗	0.413	✗
MastersThesis	PersonList	✗	0.38	✗
MastersThesis	InBook	✗	0.022	✗
MastersThesis	Monograph	✗	0.2	✗
MastersThesis	Reference	✗	0.133	✗
MastersThesis	Deliverable	✗	0.179	✗
MastersThesis	MotionPicture	✗	0.229	✗
MastersThesis	InCollection	✗	0.172	✗
MastersThesis	List	✗	0.313	✗
MastersThesis	Part	✗	0.243	✗
MastersThesis	Article	✗	0.285	✗
MastersThesis	Journal	✗	0.144	✗
MastersThesis	MastersThesis	✓	1.0	✓
MastersThesis	Person	✗	0.283	✗
MastersThesis	Academic	✗	0.224	✗
MastersThesis	School	✗	0.175	✗
MastersThesis	LectureNotes	✗	0.302	✗
MastersThesis	Proceedings	✗	0.27	✗
MastersThesis	Organization	✗	0.189	✗
MastersThesis	Misc	✗	0.263	✗
MastersThesis	Collection	✗	0.185	✗
MastersThesis	Informal	✗	0.183	✗
MastersThesis	Institution	✗	0.23	✗
MastersThesis	PageRange	✗	0.201	✗
MastersThesis	Report	✗	0.223	✗
MastersThesis	Booklet	✗	0.169	✗
MastersThesis	Book	✗	0.013	✗
MastersThesis	Manual	✗	0.151	✗

Chapitre V : Test et Validation

MastersThesis	Conference	*	0.162	*
MastersThesis	InProceedings	*	0.248	*
MastersThesis	Date	*	0.266	*
PhdThesis	Unpublished	*	0.318	*
PhdThesis	Publisher	*	0.342	*
PhdThesis	TechReport	*	0.215	*
PhdThesis	Address	*	0.262	*
PhdThesis	Chapter	*	0.284	*
PhdThesis	PhdThesis	✓	1.0	✓
PhdThesis	PersonList	*	0.29	*
PhdThesis	InBook	*	0.03	*
PhdThesis	Monograph	*	0.056	*
PhdThesis	Reference	*	0.119	*
PhdThesis	Deliverable	*	0.175	*
PhdThesis	MotionPicture	*	0.157	*
PhdThesis	InCollection	*	0.171	*
PhdThesis	List	*	0.211	*
PhdThesis	Part	*	0.194	*
PhdThesis	Article	*	0.177	*
PhdThesis	Journal	*	0.018	*
PhdThesis	MastersThesis	*	0.3	*
PhdThesis	Person	*	0.212	*
PhdThesis	Academic	*	0.198	*
PhdThesis	School	*	0.183	*
PhdThesis	LectureNotes	*	0.184	*
PhdThesis	Proceedings	*	0.277	*
PhdThesis	Organization	*	0.139	*
PhdThesis	Misc	*	0.082	*
PhdThesis	Collection	*	0.181	*
PhdThesis	Informal	*	0.029	*
PhdThesis	Institution	*	0.19	*
PhdThesis	PageRange	*	0.152	*
PhdThesis	Report	*	0.184	*
PhdThesis	Booklet	*	0.162	*
PhdThesis	Book	*	0.018	*
PhdThesis	Manual	*	0.018	*
PhdThesis	Conference	*	0.116	*
PhdThesis	InProceedings	*	0.232	*
PhdThesis	Date	*	0.259	*
Unpublished	Unpublished	✓	1.0	✓
Unpublished	Publisher	*	0.476	*
Unpublished	TechReport	*	0.149	*
Unpublished	Address	*	0.141	*
Unpublished	Chapter	*	0.164	*
Unpublished	PhdThesis	*	0.249	*
Unpublished	PersonList	*	0.261	*
Unpublished	InBook	*	0.173	*

Chapitre V : Test et Validation

Unpublished	Monograph	*	0.154	*
Unpublished	Reference	*	0.123	*
Unpublished	Deliverable	*	0.183	*
Unpublished	MotionPicture	*	0.211	*
Unpublished	InCollection	*	0.164	*
Unpublished	List	*	0.07	*
Unpublished	Part	*	0.13	*
Unpublished	Article	*	0.191	*
Unpublished	Journal	*	0.206	*
Unpublished	MastersThesis	*	0.185	*
Unpublished	Person	*	0.22	*
Unpublished	Academic	*	0.137	*
Unpublished	School	*	0.146	*
Unpublished	LectureNotes	*	0.22	*
Unpublished	Proceedings	*	0.247	*
Unpublished	Organization	*	0.132	*
Unpublished	Misc	*	0.046	*
Unpublished	Collection	*	0.161	*
Unpublished	Informal	*	0.158	*
Unpublished	Institution	*	0.159	*
Unpublished	PageRange	*	0.154	*
Unpublished	Report	*	0.131	*
Unpublished	Booklet	*	0.191	*
Unpublished	Book	*	0.13	*
Unpublished	Manual	*	0.22	*
Unpublished	Conference	*	0.139	*
Unpublished	InProceedings	*	0.23	*
Unpublished	Date	*	0.045	*
List	Unpublished	*	0.07	*
List	Publisher	*	0.307	*
List	TechReport	*	0.133	*
List	Address	*	0.15	*
List	Chapter	*	0.297	*
List	PhdThesis	*	0.177	*
List	PersonList	*	0.203	*
List	InBook	*	0.152	*
List	Monograph	*	0.138	*
List	Reference	*	0.183	*
List	Deliverable	*	0.215	*
List	MotionPicture	*	0.172	*
List	InCollection	*	0.231	*
List	List	✓	0.932	✓
List	Part	*	0.266	*
List	Article	*	0.332	*
List	Journal	*	0.167	*
List	MastersThesis	*	0.186	*
List	Person	*	0.195	*

Chapitre V : Test et Validation

List	Academic	*	0.073	*
List	School	*	0.234	*
List	LectureNotes	*	0.192	*
List	Proceedings	*	0.117	*
List	Organization	*	0.251	*
List	Misc	*	0.288	*
List	Collection	*	0.285	*
List	Informal	*	0.158	*
List	Institution	*	0.3	*
List	PageRange	*	0.018	*
List	Report	*	0.252	*
List	Booklet	*	0.104	*
List	Book	*	0.078	*
List	Manual	*	0.078	*
List	Conference	*	0.099	*
List	InProceedings	*	0.136	*
MastersThesis	Unpublished	*	0.252	*

Tableau 6: La comparaison des correspondances trouvé manuellement et celle trouvé avec notre système d'alignement de deux ontologies sans la réutilisation de mapping.

Le tableau suivant représente la comparaison des correspondances trouvé de façon manuelle et celle trouvé avec notre système d'alignement de deux ontologies avec la réutilisation de mapping :

Concept 1	Concept 2	Correspondances Manuelles	Mesure de similarité	Correspondances de notre système
MastersThesis	Unpublished	*	0.247	*
MastersThesis	Publisher	*	0.316	*
MastersThesis	TechReport	*	0.236	*
MastersThesis	Address	*	0.346	*
MastersThesis	Chapter	*	0.28	*
MastersThesis	PhdThesis	*	0.408	*
MastersThesis	PersonList	*	0.375	*
MastersThesis	InBook	*	0.017	*
MastersThesis	Monograph	*	0.195	*
MastersThesis	Reference	*	0.128	*
MastersThesis	Deliverable	*	0.174	*
MastersThesis	MotionPicture	*	0.224	*
MastersThesis	InCollection	*	0.167	*
MastersThesis	List	*	0.308	*
MastersThesis	Part	*	0.238	*
MastersThesis	Article	*	0.28	*
MastersThesis	Journal	*	0.139	*
MastersThesis	MastersThesis	✓	1.0	✓

Chapitre V : Test et Validation

MastersThesis	Person	*	0.278	*
MastersThesis	Academic	*	0.219	*
MastersThesis	School	*	0.17	*
MastersThesis	LectureNotes	*	0.297	*
MastersThesis	Proceedings	*	0.265	*
MastersThesis	Organization	*	0.184	*
MastersThesis	Misc	*	0.258	*
MastersThesis	Collection	*	0.18	*
MastersThesis	Informal	*	0.178	*
MastersThesis	Institution	*	0.225	*
MastersThesis	PageRange	*	0.196	*
MastersThesis	Report	*	0.218	*
MastersThesis	Booklet	*	0.164	*
MastersThesis	Book	*	0.008	*
MastersThesis	Manual	*	0.146	*
MastersThesis	Conference	*	0.157	*
MastersThesis	InProceedings	*	0.243	*
MastersThesis	Date	*	0.261	*
PhdThesis	Unpublished	*	0.313	*
PhdThesis	Publisher	*	0.337	*
PhdThesis	TechReport	*	0.21	*
PhdThesis	Address	*	0.257	*
PhdThesis	Chapter	*	0.279	*
PhdThesis	PhdThesis	✓	1.0	✓
PhdThesis	PersonList	*	0.285	*
PhdThesis	InBook	*	0.025	*
PhdThesis	Monograph	*	0.051	*
PhdThesis	Reference	*	0.114	*
PhdThesis	Deliverable	*	0.17	*
PhdThesis	MotionPicture	*	0.152	*
PhdThesis	InCollection	*	0.166	*
PhdThesis	List	*	0.206	*
PhdThesis	Part	*	0.189	*
PhdThesis	Article	*	0.172	*
PhdThesis	Journal	*	0.013	*
PhdThesis	MastersThesis	*	0.295	*
PhdThesis	Person	*	0.207	*
PhdThesis	Academic	*	0.193	*
PhdThesis	School	*	0.178	*
PhdThesis	LectureNotes	*	0.179	*
PhdThesis	Proceedings	*	0.272	*
PhdThesis	Organization	*	0.134	*
PhdThesis	Misc	*	0.077	*
PhdThesis	Collection	*	0.176	*
PhdThesis	Informal	*	0.024	*
PhdThesis	Institution	*	0.185	*
PhdThesis	PageRange	*	0.147	*

Chapitre V : Test et Validation

PhdThesis	Report	*	0.179	*
PhdThesis	Booklet	*	0.157	*
PhdThesis	Book	*	0.013	*
PhdThesis	Manual	*	0.013	*
PhdThesis	Conference	*	0.111	*
PhdThesis	InProceedings	*	0.227	*
PhdThesis	Date	*	0.254	*
Unpublished	Unpublished	✓	1.0	✓
Unpublished	Publisher	*	0.471	*
Unpublished	TechReport	*	0.144	*
Unpublished	Address	*	0.136	*
Unpublished	Chapter	*	0.159	*
Unpublished	PhdThesis	*	0.244	*
Unpublished	PersonList	*	0.256	*
Unpublished	InBook	*	0.168	*
Unpublished	Monograph	*	0.149	*
Unpublished	Reference	*	0.118	*
Unpublished	Deliverable	*	0.178	*
Unpublished	MotionPicture	*	0.206	*
Unpublished	InCollection	*	0.159	*
Unpublished	List	*	0.065	*
Unpublished	Part	*	0.125	*
Unpublished	Article	*	0.186	*
Unpublished	Journal	*	0.201	*
Unpublished	MastersThesis	*	0.18	*
Unpublished	Person	*	0.215	*
Unpublished	Academic	*	0.132	*
Unpublished	School	*	0.141	*
Unpublished	LectureNotes	*	0.215	*
Unpublished	Proceedings	*	0.242	*
Unpublished	Organization	*	0.127	*
Unpublished	Misc	*	0.041	*
Unpublished	Collection	*	0.156	*
Unpublished	Informal	*	0.153	*
Unpublished	Institution	*	0.154	*
Unpublished	PageRange	*	0.149	*
Unpublished	Report	*	0.126	*
Unpublished	Booklet	*	0.186	*
Unpublished	Book	*	0.125	*
Unpublished	Manual	*	0.215	*
Unpublished	Conference	*	0.134	*
Unpublished	InProceedings	*	0.225	*
Unpublished	Date	*	0.04	*
List	Unpublished	*	0.065	*
List	Publisher	*	0.302	*
List	TechReport	*	0.128	*
List	Address	*	0.145	*

Chapitre V : Test et Validation

List	Chapter	*	0.292	*
List	PhdThesis	*	0.172	*
List	PersonList	*	0.198	*
List	InBook	*	0.147	*
List	Monograph	*	0.133	*
List	Reference	*	0.178	*
List	Deliverable	*	0.21	*
List	MotionPicture	*	0.167	*
List	InCollection	*	0.226	*
List	List	✓	0.937	✓
List	Part	*	0.261	*
List	Article	*	0.327	*
List	Journal	*	0.162	*
List	MastersThesis	*	0.181	*
List	Person	*	0.19	*
List	Academic	*	0.068	*
List	School	*	0.229	*
List	LectureNotes	*	0.187	*
List	Proceedings	*	0.112	*
List	Organization	*	0.246	*
List	Misc	*	0.283	*
List	Collection	*	0.28	*
List	Informal	*	0.153	*
List	Institution	*	0.295	*
List	PageRange	*	0.013	*
List	Report	*	0.247	*
List	Booklet	*	0.099	*
List	Book	*	0.073	*
List	Manual	*	0.073	*
List	Conference	*	0.094	*
List	InProceedings	*	0.131	*
MastersThesis	Unpublished	*	0.239	*

Tableau 7: La comparaison des correspondances trouvé manuellement et celle trouvé avec notre système d'alignement de deux ontologies avec la réutilisation de mapping.

Remarque :

- Pour ce test nous avons fixé la valeur de α à 0.01.
- Nous remarquons bien que les valeurs de similarité sont raffinées.
- Nous remarquons bien aussi que ce raffinement n'a pas influencé sur la validité des mapping.

5. Validation

➤ Les mesures de rappel, précision, et Fallout :

$$\text{On a : } \left\{ \begin{array}{l} |\text{Nfound}| = 5 \\ |\text{Nexpected}| = 5 \\ |\text{Ncorrect}| = 5 \end{array} \right.$$

$$\text{Donc : } \left\{ \begin{array}{l} \text{Précision} = \frac{|\text{Ncorrect}|}{|\text{Nfound}|} = 1 \\ \text{Rappel} = \frac{|\text{Nexpected}|}{|\text{Ncorrect}|} = 1 \\ \text{Fallout} = \frac{|\text{Nfound}| - |\text{Ncorrect}|}{|\text{Nfound}|} = 0 \end{array} \right.$$

- ✓ Les résultats obtenus par l'application des opérations sont comparés aux résultats obtenus par le meilleur outil d'alignement selon l'OAEI. Nous avons montré que nos résultats seront raffinés et s'améliorent à chaque fois qu'on réutilise le mapping et va correspondre à ceux de la campagne OAEI, donc notre système de mapping d'ontologie donne un bon alignement entre les ontologies.
- ✓ Il est clair que le résultat avec la réutilisation de mapping à générer un alignement raffiné et prend que les valeurs $\geq 0,75$. Donc c'est un alignement efficace.

6. Conclusion

Ce chapitre a été consacré aux tests et validation de notre système. En ce qui concerne les objectifs listés dans le chapitre d'introduction, l'évaluation et l'expérimentation de notre système ont permis de répondre au but de notre système qui est la combinaison des techniques de base de l'alignement et les techniques de réutilisation des Mapping pour avoir un bon résultat de l'alignement.

Conclusion générale

Ces dernières années, les ontologies ont suscité de nombreux travaux dans le domaine du web sémantique. Toutefois, en raison de la nature décentralisée du web sémantique, les ontologies sont très hétérogènes.

L'alignement des ontologies est une solution pour résoudre l'hétérogénéité sémantique des données, en proposant des correspondances entre des entités sémantiques similaires de différentes ontologies.

L'hétérogénéité sémantique présente un défi majeur dans le processus d'alignement des ontologies. En effet, les ontologies existantes sont conçues indépendamment, par des concepteurs différents, ayant des objectifs applicatifs différents. Chacune peut donc avoir un point de vue différent sur le même concept.

Les travaux menés dans ce mémoire nous ont permis d'approfondir nos connaissances dans le domaine de l'ingénierie des connaissances (et plus particulièrement l'ingénierie Ontologique).

Notre objectif est de proposer une architecture d'alignement d'ontologies pour deux ontologies en OWL qui décrivent le même domaine, en utilisant différentes techniques reposantes sur la définition de mesures de similarité et la réutilisation de mapping afin de trouver les relations entre les entités de ces ontologies.

Nous avons combiné des techniques de base de l'alignement tel que les techniques syntaxiques (mesure de Jaccard, Levenshtein et Jaro), structurelles (mesure de Wu & Palmer), lexicales (mesure de WordNet) et à base d'instances (mesure de Jaccard pour les instances) afin de rendre l'alignement plus pertinent.

Nous avons aussi utilisé la technique de réutilisation des mapping tout en raffinant les mesures de similarité calculée préalablement. Pour ce raffinement nous avons proposé une formule afin d'améliorer la qualité des mapping résultant.

Notre but est d'appliquer les techniques de réutilisation des mapping avec les techniques de calcul des mesures de similarité pour avoir un bon résultat de l'alignement.

En résumé, nous disant que les objectifs initiaux sont atteints.

«Annexe 1»

Ontologie N°103 OWL :

```

<?xml version="1.0" encoding="iso-8859-1" ?>
<!DOCTYPE rdf:RDF [
  <!ENTITY xsd      "http://www.w3.org/2001/XMLSchema#" >
  <!ENTITY rdf      "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
  <!ENTITY rdfs     "http://www.w3.org/2000/01/rdf-schema#" >
  <!ENTITY dc       "http://purl.org/dc/elements/1.1/" >
  <!ENTITY owl    "http://www.w3.org/2002/07/owl#" >
  <!ENTITY foaf     "http://xmlns.com/foaf/0.1/" >
  <!ENTITY ical     "http://www.w3.org/2002/12/cal/ical#">]
<rdf:RDF
  xmlns="http://caei.ontologymatching.org/tests/101/onto.rdf#"
  xml:base="http://caei.ontologymatching.org/tests/101/onto.rdf#"
  xmlns:foaf ="&foaf;"
  xmlns:ical ="&ical;"
  xmlns:rdf ="&rdf;"
  xmlns:xsd ="&xsd;"
  xmlns:rdfs ="&rdfs;"
  xmlns:owl ="http://www.w3.org/2002/07/owl#"
  xmlns:dc ="&dc;"
  <!-- This ontology describes bibliographic references. -->
  <owl:Ontology rdf:about="">
    <dc:creator>Nick Khouf <mailto:nkhouf@mit.edu></dc:creator>
    <dc:contributor>Antoine Zimmermann <mailto:antoine.zimmermann@inrialpes.fr></dc:contributor>
    <dc:contributor>J r me Euzenat</dc:contributor>
    <dc:description>Bibliographic references in OWL</dc:description>
    <dc:date>08/06/2008</dc:date>
    <rdfs:label>Bibliographic references</rdfs:label>
    <rdfs:comment>Possible ontology to describe bibTeX entries.</rdfs:comment>
    <owl:versionInfo> Id: onto.rdf,v 1.30 2008/05/27 14:41:13 euzenat Exp  </owl:versionInfo>
  </owl:Ontology>
  <!-- Every entity (even external) must be typed in OWL-DL. -->
  <!-- this is for satisfying the OWL Species validator
  (which is not satisfied anyway) -->
  <owl:Class rdf:about="&rdf;List" />
  <rdf:List rdf:about="&rdf;nil" />
  <owl:ObjectProperty rdf:about="&rdf;first">
    <rdfs:domain rdf:resource="&rdf;List" />
  </owl:ObjectProperty>

```

```

<owl:ObjectProperty rdf:about="#&rdf;rest">
  <rdfs:domain rdf:resource="#&rdf;List" />
  <rdfs:range rdf:resource="#&rdf;List" />
</owl:ObjectProperty>

<owl:DatatypeProperty rdf:about="#&dc;creator" />
<owl:DatatypeProperty rdf:about="#&dc;contributor" />
<owl:DatatypeProperty rdf:about="#&dc;description" />
<owl:DatatypeProperty rdf:about="#&dc;date" />

<!-- foaf extensions -->
<owl:Class rdf:about="#&foaf;Person" />
<owl:Class rdf:about="#&foaf;Organization" />
<owl:DatatypeProperty rdf:about="#&foaf;firstName" />
<owl:DatatypeProperty rdf:ID="lastName" />
<owl:DatatypeProperty rdf:about="#&foaf;name" />

<!--
***** ENTRIES *****

Entries form the basis of a bibTeX database and are categorized by their type,
such as a book, journal article, conference proceedings, etc. Each entry type
has a specific set of fields that are required in bibTeX; as a first pass,
these fields will also be required in the OWL ontology.

Note: all rdfs:comment values for the entries come
from http://newton.ex.ac.uk/tex/pack/bibtex/btxdoc/node6.html.
-->

<owl:Class rdf:ID="Reference">
  <rdfs:label xml:lang="en">Reference</rdfs:label>
  <rdfs:comment xml:lang="en">Base class for all entries</rdfs:comment>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#&date" />
      <owl:maxCardinality rdf:datatype="#&xsd;nonNegativeInteger">1</owl:maxCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#&title" />
      <owl:maxCardinality rdf:datatype="#&xsd;nonNegativeInteger">1</owl:maxCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>

```



```

    <owl:Restriction>
    <owl:onProperty rdf:resource="#humanCreator" />
    <owl:maxCardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:maxCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:ID="Article">
  <rdfs:subClassOf rdf:resource="#Part" />
  <rdfs:label xml:lang="en">Article</rdfs:label>
  <rdfs:comment xml:lang="en">An article from a journal or magazine.</rdfs:comment>
  <rdfs:subClassOf>
    <owl:Restriction>
    <owl:onProperty rdf:resource="#author" />
    <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
    <owl:onProperty rdf:resource="#pages" />
    <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
    <owl:onProperty rdf:resource="#journal" />
    <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
    <owl:onProperty rdf:resource="#date" />
    <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
    <owl:onProperty rdf:resource="#date" />
    <owl:allValuesFrom rdf:resource="#Date" />
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
    <owl:onProperty rdf:resource="#number" />

```

```

<owl:maxCardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:maxCardinality>
| </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
| <owl:Restriction>
| <owl:onProperty rdf:resource="#volume" />
| <owl:maxCardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:maxCardinality>
| </owl:Restriction>
</rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:ID="Book">
| <rdfs:subClassOf rdf:resource="#Reference" />
| <rdfs:label xml:lang="en">Book</rdfs:label>
| <rdfs:comment xml:lang="en">A book that may be a monograph or a collection of written texts.
| </rdfs:comment>
| <rdfs:subClassOf>
| | <owl:Restriction>
| | <owl:onProperty rdf:resource="#title" />
| | <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
| | </owl:Restriction>
| </rdfs:subClassOf>
| <rdfs:subClassOf>
| | <owl:Restriction>
| | <owl:onProperty rdf:resource="#volume" />
| | <owl:maxCardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:maxCardinality>
| | </owl:Restriction>
| </rdfs:subClassOf>
| <rdfs:subClassOf>
| | <owl:Restriction>
| | <owl:onProperty rdf:resource="#publisher" />
| | <owl:maxCardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:maxCardinality>
| | </owl:Restriction>
| </rdfs:subClassOf>
| <rdfs:subClassOf>
| | <owl:Restriction>
| | <owl:onProperty rdf:resource="#series" />
| | <owl:maxCardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:maxCardinality>
| | </owl:Restriction>
| </rdfs:subClassOf>
| <rdfs:subClassOf>
| | <owl:Restriction>
| | <owl:onProperty rdf:resource="#date" />
| | <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>

```



```

    </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
<owl:onProperty rdf:resource="#date" />
<owl:allValuesFrom rdf:resource="#Date" />
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
<owl:onProperty rdf:resource="#author" />
<owl:cardinality rdf:datatype="xsd:nonNegativeInteger">1</owl:cardinality>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
<owl:onProperty rdf:resource="#edition" />
<owl:maxCardinality rdf:datatype="xsd:nonNegativeInteger">1</owl:maxCardinality>
  </owl:Restriction>
</rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:ID="Monograph">
  <rdfs:subClassOf rdf:resource="#Book" />
  <rdfs:label xml:lang="en">Monograph</rdfs:label>
  <rdfs:comment xml:lang="en">A book that is a single entity, as opposed to a collection.</rdfs:comment>
  <rdfs:subClassOf>
    <owl:Restriction>
<owl:onProperty rdf:resource="#chapters" />
<owl:allValuesFrom rdf:resource="#Chapter"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:ID="Collection">
  <rdfs:subClassOf rdf:resource="#Book" />
  <rdfs:label xml:lang="en">Collection</rdfs:label>
  <rdfs:comment xml:lang="en">A book that is collection of texts or articles.</rdfs:comment>
  <rdfs:subClassOf>
    <owl:Restriction>
<owl:onProperty rdf:resource="#chapters" />
<owl:allValuesFrom rdf:resource="#Chapter"/>
    </owl:Restriction>

```

```

</rdfs:subClassOf>
<!-- This is forbidden for transitive properties in OWL-DL -->
<!--rdfs:subClassOf>
|   <owl:Restriction>
<owl:onProperty rdf:resource="#isPartOf" />
<owl:cardinality rdf:datatype="xsd:nonNegativeInteger">1</owl:cardinality>
|   </owl:Restriction>
</rdfs:subClassOf-->
</owl:Class>

<owl:Class rdf:ID="Chapter">
  <rdfs:subClassOf rdf:resource="#Part" />
  <rdfs:label xml:lang="en">BookPart</rdfs:label>
  <rdfs:comment xml:lang="en">A chapter (or section or whatever) of a book having its own
  title.</rdfs:comment>
  <rdfs:subClassOf>
  |   <owl:Restriction>
  <owl:onProperty rdf:resource="#chapter" />
  <owl:maxCardinality rdf:datatype="xsd:nonNegativeInteger">1</owl:maxCardinality>
  |   </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
  |   <owl:Restriction>
  <owl:onProperty rdf:resource="#chapter" />
  <owl:maxCardinality rdf:datatype="xsd:nonNegativeInteger">1</owl:maxCardinality>
  |   </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:ID="InBook">
  <rdfs:subClassOf rdf:resource="#Part" />
  <rdfs:label xml:lang="en">InBook</rdfs:label>
  <rdfs:comment xml:lang="en">A subpart of a book given by a range of pages.</rdfs:comment>
  <rdfs:subClassOf>
  |   <owl:Restriction>
  <!-- author OR editor (given in the Book description) ??? -->
  <owl:onProperty rdf:resource="#author" />
  <owl:cardinality rdf:datatype="xsd:nonNegativeInteger">1</owl:cardinality>
  |   </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
  |   <owl:Restriction>
  <owl:onProperty rdf:resource="#pages" />

```



```

    <owl:minCardinality rdf:datatype="&xsd:nonNegativeInteger">1</owl:minCardinality>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#book" />
    <owl:cardinality rdf:datatype="&xsd:nonNegativeInteger">1</owl:cardinality>
  </owl:Restriction>
</rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:ID="InCollection">
  <rdfs:subClassOf rdf:resource="#Part" />
  <rdfs:label xml:lang="en">InCollection</rdfs:label>
  <rdfs:comment xml:lang="en">A part of a book having its own title.</rdfs:comment>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#author" />
      <owl:cardinality rdf:datatype="&xsd:nonNegativeInteger">1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#collection" />
      <owl:cardinality rdf:datatype="&xsd:nonNegativeInteger">1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:ID="InProceedings"> <!-- Communication -->
  <rdfs:subClassOf rdf:resource="#Part" />
  <rdfs:label xml:lang="en">InProceedings</rdfs:label>
  <rdfs:comment xml:lang="en">An article in a conference proceedings.</rdfs:comment>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#author" />
      <owl:cardinality rdf:datatype="&xsd:nonNegativeInteger">1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#proceedings" />
      <owl:cardinality rdf:datatype="&xsd:nonNegativeInteger">1</owl:cardinality>
    </owl:Restriction>

```

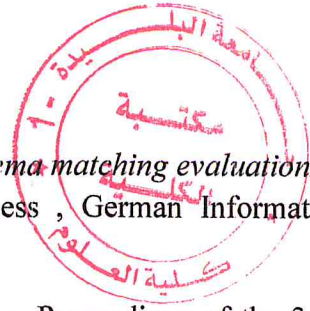
Références bibliographiques

- [1]. Neeches, Finin T ,Fikes R.E, Gruber T.R, Senator T et Swartou W.R. (1993).« Enabling technology for knowledge sharing » AI Magazine. Vol.12, no 3.
- [2]. Gruber T. (1993). A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition, 5(2), 199-220.
- [3]. Guarino N. (1997b). Understanding, building and using ontologies. International J. Human-Computer Studies, 46, 293-310.
- [4]. Borst W. N. (1997). Construction of Engineering Ontologies. Center for Telematica and Information Technology, University of Tweenty, Enschede, NL.
- [5]. Mike Uschold and Martin King.(1995) AIA1-TR-183.
- [6]. Mr. BELABED, « Introduction aux ontologies », cours de l'intelligence artificielle master 1 système d'information et de connaissance.
- [7]. BORG96, A. Borgida.(1996). "On the relative expressiveness of description logics and predicate logics". Artificial Intelligence, volume 82, number 1-2, pages 353–367.
- [8]. Blas et al. 98, BLAZQUEZ M. FERNANDEZ M., GARCIA-PINAR J. M. & GOMEZ-PEREZ A.(1998), Building Ontologies at the Knowledge Level using the Ontology Design Environment, in Proceedings of the Banff Workshop on Knowledge Acquisition for Knowledge-based Systems.
- [9]. Dien et al. 01, DIENG R., CORBY O., GANDON F., GIBOIN A., GOLEBIEWSKA J., MATTA N. et RIBIERE M.(2001), «Méthodes et outils pour la gestion des connaissances : une approche pluridisciplinaire du knowledge Management ». Dunod, 2 edition.
- [10]. Pere 99, Paterno F.(1999), Model Based Design and Evaluation of Interactive Application. Springer Verlag.
- [11]. Audrey Baneyx,2007,CONSTRUIRE UNE ONTOLOGIE DE LA PNEUMOLOGIE.Cycle de vie d'une ontologie,1 ,38.
- [12]. Gomez-Perez, A. 1999, Ontological Engineering: a State of the Art. Expert Update, 2(3), 33-43.
- [13]. Gruber, T. 1993, A translation approach to portable ontology specification. Knowledge acquisition, 5, 199-220.
- [14]. Uschold, M., & Gruninger, M. 1996, Ontologies: Principles, Methods and Applications. Knowledge Engineering Review, 11(2).

- [15]. Bachimont, B. 2000. Engagement Sémantique et Engagement Ontologique : Conception et Réalisation d'Ontologies en Ingénierie des Connaissances. (J. Z. M. Charlet, G. Kassel, & D. Bourgault, Éd.s.) Ingénierie des Connaissances, Evolutions Récentes et Nouveaux Défis, 305-323.
- [16]. webmaster-hub.com/publication/L-autre-semantique-Le-Web.html. alysis of the umls metathesaurus. AMIA 98 Annual Symposium
- [17]. Lassila et Swick, 1999, Lassila, O. et Swick, R. R. (1999). Resource description framework (rdf) model and syntax specification. World Wide Web Consortium W3C Recommendation 22 February 1999 / W3C /.
- [18]. Corcho O, Fernández-López M et al, 2003, Methodologies, tools and languages for building ontologies. Where is their meeting point?, 46, 41-64.
- [19]. McGuinness et al. 2002, Daml+oil: An ontology language for the semantic web. IEEE Intelligent Systems, 17, p72-80.
- [20]. Xu, L., & Embley, D. (2003, October 20). Using Domain Ontologies to Discover Direct and Indirect Matches for Schema Elements. In : Second International Semantic Web Conference (ISWC- 03).
- [21]. He, B., Chang, K., & Han, J. (2003, December). Automatic Complex Schema Matching across Web Query Interfaces: A Correlation Mining Approach. Technical Report UIUCDCS R-2003-2388, Department of Computer Science, UIUC.
- [22]. Y. Kalfoglou, M. Schorlemmer.: Ontology mapping: the state of the art. The Knowledge Engineering Review, 18(1):1-31, 2003.
- [23]. P. Shvaiko, J. Euzenat.: A survey of schema-based matching approaches. Journal on Data Semantics, IV: pp 146-171, 2005.
- [24]. J. Euzenat, P. Shvaiko.: Ontology Matching. ISBN 978-3-540-49611-3, SpringerVerlag Berlin Heidelberg 2007.
- [25]. Rahm, E., & Bernstein, P. A. (2001). A survey of approaches to automatic schema matching. VLDB Journal: Very Large Data Bases. 10 (4), pp. 334-350

- [26]. Do, H. H., & Rahm, E. (2002, Août). COMA - A System for Flexible Combination of Schema Matching Approaches. In : Proceedings of the 28th Intl. Conference on Very Large Databases (VLDB), Hongkong.
- [27]. David Aumuller, Hong-Hai Do, Sabine Masmann, and Erhard Rahm. (2005). Schema and ontology matching with COMA++. In *Proc. 24th International Conference on Management of Data (SIGMOD), Software Demonstration*, pages 906–908, Baltimore (MD US).
- [28]. Erhard Rahm, Hong-Hai Do, and Sabine Masmann.(2004). Matching large XML schemas. *ACM SIGMOD Record*, 33(4):26–31.
- [29]. M. Ehrig.: *Ontology alignment: bridging the semantic gap. Semantic web and beyond: computing for human experience*. Springer, New-York (NY US), 2007.
- [30]. S. Zanobini.: *Semantic coordination: the model and an application to schema matching*. PhD thesis, International Doctorate School in Information and Communication Technology, University of Trento, Trento (IT), March 2006.
- [31]. Noy, N. F., & Musen, M. A. (2000). Prompt: algorithm and tool for automated ontology merging and alignment. In *Proceeding of Seventeenth National Conference on Artificial Intelligence AAAI*.
- [32]. McGuinness, D. L., Fikes, R., Rice, J., & Wilder, S. (2000). An Environment for Merging and Testing Large Ontologies.(KR 2000). pp. 483-493.
- [33]. OAEI. (2009). <http://oaei.ontologymatching.org/>, consulte en.
- [34]. Do H. et Rahm E. : COMA - A System for Flexible Combination of Schema Matching Approaches. In *28th International Conference on Very Large Data Bases*.
- [35]. Giunchiglia, F., Shvaiko, P. et Yatskevich,2004, M. *S-Match: an algorithm and an implementation of semantic matching*. Dans *Proceedings of ESWS, Heraklion (GR)*,p 61-75.
- [36]. Giunchiglia, F. et Shvaiko, P.2003, *Semantic matching*. Dans *Proc. IJCAI 2003 Workshop on ontologies and distributed systems, Acapulco (MX)*, pages 139–146.
- [37]. Miller, A.G.1995, *WordNet: A lexical database for English*. *Communications of the ACM*, 38(11):39–41.

- [38]. Giunchiglia F. et Yatskevich M. 2004. *Element level semantic Matching*. dans Meaning Coordination and Negotiation workshop at ISWC'04. Hiroshima, Japan.
- [39]. Ehrig, M. et Staab, S. 2004. *QOM – quick ontology mapping*. Dans Proc. 3rd ISWC, Hiroshima (JP).
- [40]. Ehrig, M. and Sure, Y. 2004, “*Ontology mapping – an integrated approach*”, in Christoph Bussler, John Davis, Dieter Fensel, and Rudi Studer, editors, Proc. 1st ESWS, Hersounisous (GR), volume 3053 of Lecture Notes in Computer Science, pages 76–91.
- [41]. Doan A., Madhavan J., Domingos P. et Halevy A. : Ontology matching: A machine learning approach. *In* Staab S. et Studer R. [SS04], pages 385–404.
- [42]. Q. Umer and D. Mundy (2012). Semantically Intelligent Semi-Automated Ontology Integration. Proceedings of the World Congress on Engineering 2012 Vol II.
- [43]. C.R. Rene Robin and G.V. Uma (2010). A Novel Algorithm for Fully Automated Ontology Merging Using Hybrid Strategy. European Journal of Scientific Research ISSN 1450-216X Vol.47 No.1 (2010), pp.074-08.
- [44]. W.W. Royce, 1970, *Managing the development of large software systems*, Proc. WESTCON, Calif., USA.
- [45]. Roques P. 2007, *UML 2 : Modéliser une application web*. Les cahiers du programmeur. 4^{ème} édition. p 182.
- [46]. Z. Wu et M. Palmer. 1994. *Verb semantics and lexical selection*. In Proceedings of the 32nd Annual Meeting of the Associations for Computational Linguistics, p 133-138.
- [47]. Slimani T et al. 2007. *Une extension de mesure de similarité entre les concepts d'une ontologie*. p 4-5.
- [48]. Arnaud M et al. *Gestion du conflit dans l'appariement des ontologies*. p 4.
- [49]. Aissa Fellah, Mimoun Malki, Ahmed ZAHAF (2008). Alignement des ontologies : utilisation de WordNet et une nouvelle mesure structurelle. CORIA - Conférence en Recherche d'Information et Applications. p 404.
- [50]. Donald G. Firesmith - B. 2002, Henderson-Sellers, *The OPEN Process Framework: An Introduction*, Pearson Education.



[51]. Do H., Melnik S., Rahm E.2002, « *Comparison of schema matching evaluations*», Proceedings of the 2nd Int. Workshop on Web Databases , German Informatics Society, Erfurt.

[52]. EON W .2004, « *EON: Ontology Alignment Contest* », Proceedings of the 3r d Workshop Evaluation of Ontology-based Tools (EON).

[53]. EON W .2006, « *EON 2006: Evaluation of Ontologies for the Web* », Proceedings of the 4th International EON Workshop,

[54].Euznat J, Shvaiko P et al, 2006, « *Results of the Ontology Alignment Evaluation Initiative 2006* », in, Proceedings of the 1st ESWC 2006 international workshop on ontology matching, Athens, (GAU US)