

Université Saad DAHLAB Blida 01



**Faculté des Sciences
Département d'Informatique
Mémoire Présenté par :**

AMEYOUN Mohamed Souheib

TAKI Mounir

**En vue d'obtenir le diplôme de master
Domaine : Mathématique et informatique
Filière : Informatique
Spécialité : Informatique
Option : Ingénierie de logiciel**

Thème :

Intégration des ontologies volumineuses

Soutenu le : 29/09/2019, devant le jury composé de :

M^{me} **FAREH** Promotrice

M^{me} **ABED** Présidente

M^{me} **MANCERE** Examinatrice

2018/2019

Table des matières

Introduction Générale

Table des matières.....	2
Table des figures.....	6
1. Introduction générale	2
2. Problématique	2
3. Objectif	3

Chapitre I : Les Ontologies

1. Introduction.....	6
2. Origine des ontologies.....	6
3. Définitions des ontologies	7
4. Pour quelles raisons développer une ontologie	9
5. Les éléments constitutifs d'une ontologie	10
5.1. Concept.....	11
5.2. Relations et fonctions	12
5.3. Axiomes.....	13
5.4. Instances	13
6. Langages de représentation des ontologies.....	13
6.1. KIF [Genesereth et al., 1992].....	14
6.2. RDF/RDFschéma	14
6.3. DAML + OIL.....	15
6.4. OWL	15
7. Opérations sur les ontologies	16
7.1. Intégration des ontologies	16
7.1.1. Mapping d'ontologies	16
7.1.2. Fusion d'ontologies.....	17
7.2. Alignement d'ontologies.....	18
7.3. Partitionnement d'ontologies	19
7.4. Maintenance d'ontologies	19
8. Cycle de vie global d'une ontologie	20
9. Conclusion	21

Chapitre II : Etat de l'art sur l'intégration des ontologies

1. Introduction.....	23
2. Les techniques de mesures de similarité.....	24
2.1. Mesure de similarité terminologique	24

2.1.1.	Mesure de similarité lexicale.....	25
2.1.2.	Mesure de similarité syntaxique	25
2.2.	Mesure de similarité structurelle	25
2.3.	Mesures de similarité sémantique	25
2.4.	Mesures de similarité extensionnelle.....	26
2.5.	Mesures de similarité Combinées	26
2.5.1.	La combinaison séquentielle (hybride)	26
2.5.2.	La combinaison parallèle (composite)	26
3.	Méthodes d'intégration existantes :	27
3.1.	S-Match [Giunchiglia et al., 2004]	27
3.2.	Falcon-AO [Hu et al., 2006]	27
3.3.	TaxoMap[Hamdi et al., 2008]	29
3.4.	Thefca-ontmergemethod(Methode basée sur l'ACF) [Guan-yu et al.,2010]	29
3.5.	Méthode basée sur la classification [Maiz et al., 2010].....	30
3.6.	COMA++ [Massmann et al., 2011].....	31
3.7.	Méthode de Ding et al [Ding et al., 2013].....	31
3.8.	Fusion par enrichissement sémantique (MONSE) [Fareh et al.,2013]	31
3.9.	Méthode de Setti et al [Setti et al., 2015].....	32
4.	La comparaison des méthodes	33
4.1.	Comparaison des méthodes selon les mesures de similarité utilisées.....	33
4.1.1.	Critère de comparaison.....	33
4.1.2.	Tableau de comparaison selon les mesures de similarité utilisées	35
4.1.3.	Analyse.....	37
4.2.	Comparaison des méthodes selon les critères du passage à l'échelle utilisé.....	38
4.2.1.	<i>Critère de comparaison</i>	38
4.2.2.	Tableau de comparaison selon le passage à l'échelle utilisé	39
4.2.3.	Analyse.....	41
5.	Partitionnement à base de clustering	41
6.	Conclusion	42

Chapitre III : Approche Proposée

1.	Introduction	44
2.	Limites des approches existantes.	45
3.	L'approche proposée pour l'intégration	45
4.1.	Mapping.....	46
4.1.1.	Mapping entre blocs	46

4.1.2.	Mapping entre concepts	47
4.2.	Les mesures de similarité utilisées	48
4.2.1.	Similarité terminologique.....	49
4.2.2.	Similarité structurelle.....	51
4.2.3.	Similarité extensionnelle.....	51
4.2.4.	Similarité sémantique	52
4.3.	Fusion.....	53
4.3.1.	Principe de la fusion.....	53
4.	Conclusion	56

Chapitre IV : Implémentation du système

1.	Introduction	58
2.	Environnement de développement	58
2.1.	Matériel utilisé	58
2.2.	Langage utilisé.....	58
2.3.	Paquets utilisés dans Python	59
2.3.1.	Numpy.....	59
2.3.2.	Owlyready2	60
2.3.3.	Wordnet	60
3.	Le logiciel Protégé pour la visualisation des ontologies.....	60
4.	Présentation de l'application.....	61
5.	Les résultats	65
5.1.	Chargement des deux ontologies.....	65
5.2.	Construire l'ontologie fusionnée.....	68
6.	Conclusion	69

Chapitre V : Tests et Validation

1.	Introduction	71
2.	Evaluation	71
3.	Evaluation du mapping.....	72
3.1.	Mesures d'évaluation utilisées.....	72
3.2.	Résultats expérimentaux et discussion	72
4.	Evaluation de la Fusion.....	73
4.1.	Mesures d'évaluation utilisées pour la syntaxe	73
4.2.	Résultats expérimentaux et discussion	74
4.3.	Mesures d'évaluation utilisées pour la sémantique.....	75
4.3.1.	Les outils pour la validation de la sémantique	75

4.3.2.	Résultats expérimentaux et discussion.....	76
5.	Conclusion	78
7.	Référence	82

Table des figures

Figure 1: Composition du mot ontologie	7
Figure 2: Utilisation des ontologies.....	10
Figure 3: Triangle sémantique	12
Figure 4: Mapping des ontologies.....	17
Figure 5: Principe de la fusion d'Ontologies.....	18
Figure 6: Principe d'alignement d'ontologies	19
Figure 7: Cycle de vie global d'une ontologie [Furst., 2002].....	20
Figure 8: Mesures de calcul de similarités [Euzenat, 2008].....	24
Figure 9: Exemple de trois ontologies locales du même domaine.....	30
Figure 10: Processus d'alignement basé sur le clustering d'ontologies	46
Figure 11: processus de mapping entre blocs.....	47
Figure 12: processus de mapping entre concepts	48
Figure 13 : Extrait de l'ontologie UnivBench.....	51
Figure 14: Exemple de fusion de deux concepts.....	54
Figure 15: exemple de chargement des racines dans la nouvelle ontologie	55
Figure 16: Exemple de fusion de deux ontologies	55
Figure 17: première fenêtre de l'application.....	61
Figure 18: Interface en cours de traitement.....	62
Figure 19: fenêtre de dialogue pour choisir l'emplacement de l'ontologie	62
Figure 20: Fenêtre d'erreur 1	63
Figure 21: trois boutons d'affichage de résultat de Mapping et de la fusion	63
Figure 22: le fichier de mapping après la fin de la fusion	64
Figure 23: le fichier de fusion après la fin de l'opération	64
Figure 24: le traitement est terminé	65
Figure 25: chargement de l'ontologie cmt et Conference.....	66
Figure 26: visualisation de l'ontologie O qui contient les deux ontologies « cmt » et « Conférence»	67
Figure 27: Code Python pour le chargement de la première ontologie	68
Figure 28: la hiérarchie de l'ontologie finale après la fusion	69
Figure 29: la validation de l'ontologie edas et ekaw par RDF Validator.....	74
Figure 30: la validation de l'ontologie edas et ekaw par OWL Validator	75
Figure 31: la visualisation de l'ontologie fusionnée par protégé.....	76
Figure 32: le résultat du test de la fusion des deux ontologies « edas » et « ekaw »	77
Figure 33: la visualisation de l'ontologie fusionnée des deux ontologies « human » et « mouse »	77
Figure 34: resultat du test de la cohérence et la consistance.....	77
Tableau 1: Tableau de comparaison selon les mesures de similarité utilisées.....	36
Tableau 2: Tableau de comparaison selon les critères du passage à l'échelle utilisé	40
Tableau 3: Les ontologies de tests.....	71
Tableau 4: présente les valeurs de rappel et de précision avec différent nombre de blocs	73

Résumé

Nées des besoins de représentation des connaissances, les ontologies sont à l'heure actuelle au cœur des travaux menés en Ingénierie des Connaissances.

L'objectif de ce travail est la conception et la réalisation d'un système permettant la l'intégration des ontologies volumineuses, basé sur le partitionnement, et ce, par réaliser l'alignement et la fusion des ontologies initiales. Pour ce faire, l'accent est mis sur l'introduction d'une dimension sémantique dans le processus d'intégration, en combinant plusieurs mesures de similarité, à savoir les mesures terminologique, structurelle et sémantique, afin de trouver une valeur très fine de degrés de ressemblances entre les entités des ontologies. Les résultats d'alignement sont utilisés pour réaliser le processus de la fusion des ontologies, en construisant la nouvelle hiérarchie de l'ontologie fusionnée, cette dernière sera validée selon la syntaxe et la sémantique.

Mots-Clés : Ontologie Volumineuse, Alignement, Sémantique, Mesure de similarité, Mapping, Fusion.

ملخص

اخترعت الأنطولوجيا من الحاجة إلى تمثيل المعرفة، وهي في صميم العمل المنجز في هندسة المعرفة.

الهدف من هذا العمل هو تصميم وتنفيذ نظام يسمح بدمج الأنطولوجيا الضخمة استنادًا إلى التقسيم من خلال تحقيق التوافق والانصهار للأنطولوجيا الأولية.

للقيام بذلك، ينصب التركيز على إدخال بعد دلالي في عملية التكامل، من خلال الجمع بين العديد من تدابير التشابه، وهي التدابير المصطلحات والهيكلية والدلالية، من أجل إيجاد قيمة جيدة للغاية في درجات التشابه بين الكيانات الأنطولوجية. يتم استخدام نتائج المحاذاة لإجراء عملية دمج الأنطولوجيا، من خلال بناء

التسلسل الهرمي الجديد للأنطولوجيا المدمجة، وسيتم التحقق من صحة هذا الأخير وفقًا للجمل والدلالات.

الكلمات المفتاحية: الأنطولوجيا الضخمة، المحاذاة، الدلالات، قياس التشابه، رسم الخرائط، الانصهار.

Summary

Born from the need to represent knowledge, ontologies are currently at the heart of the work carried out in Knowledge Engineering.

The goal of this work is the design and realization of a system allowing the integration of voluminous ontologies, based on partitioning, by realizing the alignment and the fusion of the initial ontologies. To do this, the emphasis is on introducing a semantic dimension into the integration process, by combining several similarity measures, namely terminological, structural and semantic measures, in order to find a very fine value of degrees of similarities between ontology entities. Alignment results are used to perform the process of ontology merging, by building the new hierarchy of the merged ontology, the latter will be validated according to syntax and semantics.

Keywords: Voluminous Ontology, Alignment, Semantics, Similarity Measure, Mapping, Fusion.

Introduction Générale

1. Introduction générale

Contexte

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.

Certaines des principales caractéristiques des ontologies sont qu'elles garantissent une compréhension commune de l'information et qu'elles émettent des hypothèses des domaines explicites. En conséquence, l'interconnexion et l'interopérabilité du modèle en font un outil précieux pour relever les défis d'accès et d'interrogation des données dans les grandes organisations.

L'intégration des ontologies a été au cœur de nombreuses recherches afin de réduire le coût qu'incombe l'intégration manuelle d'ontologies.

La grande disponibilité de données et de connaissances a nécessité des méthodologies efficaces pour la réutilisation et la manipulation de ces connaissances.

2. Problématique

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

Dans les domaines d'applications réelles, les ontologies devenant de plus en plus volumineuses, par exemples en agronomie et en médecine, des ontologies comportant des dizaines de milliers de concepts (AGROVOC : 28 439, NALT : 42 326, NCI : 25 652), l'efficacité des méthodes d'intégration diminue considérablement, vu la complexité de ce processus, avec la prise en compte de la dimension sémantique dans l'intégration sur un très grand nombre de concepts dans les ontologies initiales.

En effet, l'hétérogénéité sémantique présente un défi majeur dans le processus d'intégration des ontologies, elle est due aux différentes interprétations des objets du monde réel.

3. Objectif

L'objectif de ce projet consiste à concevoir et réaliser un système d'intégration des ontologies de grande taille, en effet, dans plusieurs domaines les connaissances sont représentées à travers des ontologies volumineuses. Elles contiennent des milliers de concepts et de relations entre ces derniers. En conséquent le problème du passage à l'échelle pour l'intégration des ontologies a une très grande importance.

Notre objectif est de réaliser un système d'intégration d'ontologie volumineuse, l'intégration des ontologies peut être réalisée par trouver des équivalences entre les différents concepts des deux ontologies en gardant les ontologies séparées, ou réaliser une fusion des deux ontologies, afin de créer une ontologie unique et cohérente, en traitant le problème de l'hétérogénéité sémantique.

Afin d'atteindre le but de notre travail, le mémoire s'articulera autour de 5 chapitres :

Chapitre 1 : Généralités sur les ontologies

Ce chapitre présente des généralités sur les ontologies, en soulignant l'objectif de l'utilisation des ontologies et les différentes opérations sur les ontologies.

Chapitre 2 : Les travaux existants sur l'intégration des ontologies

Ce chapitre propose des généralités sur des méthodes utilisées pour l'intégration des ontologies, et les différentes techniques de passage à l'échelle des ontologies, par la suite nous réalisons une étude comparative afin d'élaborer notre approche.

Chapitre 3 : L'approche proposée

Ce chapitre propose la description de l'approche proposée et met l'accent sur les limites des approches déjà proposées, ainsi qu'une présentation des techniques de similarités utilisés dans notre approche.

Chapitre 4 : Implémentation

Ce chapitre présente les différents outils utilisés pour la réalisation de notre système, et représente le résultat de la conception pour la réalisation de notre système d'intégration des ontologies volumineuse. Il s'agit de présenter l'implémentation de notre système.

Chapitre 5 : Tests et validation

Ce chapitre présent les résultats obtenus, nous détaillons l'ensemble des métriques de qualité sur lesquels se base l'évaluation de nos résultats de mapping et de fusion obtenus.

Chapitre I : Les Ontologies

1. Introduction

L'exploitation de connaissances en informatique a pour objectif de ne plus faire manipuler en aveugle des informations à la machine mais de permettre un dialogue (une coopération) entre le système et les utilisateurs. Alors, le système doit avoir accès non seulement aux termes utilisés par l'être humain mais aussi à la sémantique qui leur est associée, afin qu'une communication efficace soit possible.

Dans ce chapitre nous illustrons quelque concept de base sur les ontologies on générale

2. Origine des ontologies

Le terme Ontologie a été utilisé pour la première fois par les philosophes grecs dans une discipline qui a plus 2300 ans, qui s'intéresse à l'existence de l'être en tant qu'être et aux catégories fondamentales de l'existant.

Étymologiquement parlant, Onto est un terme dérivé d'un mot grec et signifie l'Être (ainsi que ses manifestations) Logie vient du mot grec Logos qui veut dire Science ou Étude. Donc, l'Ontologie est la science qui s'intéresse à l'étude de l'être en tant qu'être.

Dans la philosophie classique, l'Ontologie correspond à ce qu'Aristote appelait la Philosophie première, c'est-à-dire la science de l'être en tant qu'être, par opposition aux philosophies secondes qui s'intéressent à l'étude des manifestations de l'être (les existants). Ensuite la notion d'ontologie a été abordée dans le domaine de l'intelligence artificielle (IA) par John McCarthy. Il affirmait déjà en 1980 que les concepteurs des systèmes intelligents fondés sur la logique devraient d'abord énumérer tout ce qui existe.

3. Définitions des ontologies

Ontologie est une branche de la métaphysique qui s'intéresse à l'existence, à l'être en tant qu'être et aux catégories fondamentales de l'existant. En effet, ce terme est construit à partir des racines grecques « ontos » qui veut dire ce qui existe, l'être, l'existant, et « logos » qui veut dire l'étude, le discours, d'où sa traduction par « l'étude de l'être » et par extension de l'existence comme présente la figure 1.

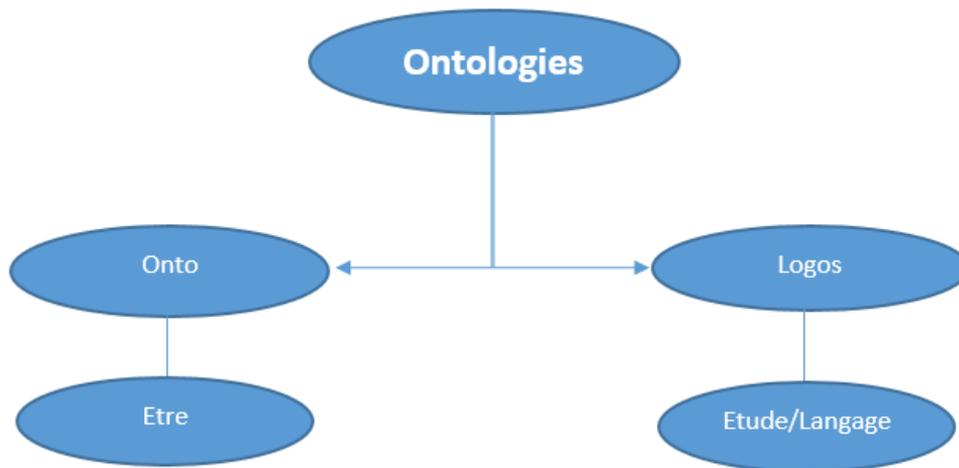


Figure 1: Composition du mot ontologie

Par la suite nous présentons quelques définitions des ontologies.

- **Définition 1**

« Une ontologie définit les termes et les relations de base du vocabulaire d'un domaine ainsi que les règles qui indiquent comment combiner les termes et les relations de façon à pouvoir étendre le vocabulaire » [Neeches et al., 1991].

- **Définition 2**

« Spécification explicite d'une conceptualisation » [Gruber, T. R. 1993].

Le terme "Conceptualisation" se réfère à un modèle abstrait d'un domaine, où nous relevons tous les concepts de ce domaine et le terme "explicite" signifie que le type de concept et les contraintes sur ce dernier sont explicitement définis.

- **Définition 3**

«Une ontologie est une théorie logique qui permet une spécification explicite et partielle d'une conceptualisation»[Guarino, 1995].

Où le terme conceptualisation désigne l'idée de la réalité qu'un individu ou un groupe d'individus peuvent avoir.

- **Définition 4**

«Une ontologie fournit le moyen pour décrire d'une manière explicite la conceptualisation des connaissances représentées dans les bases de connaissances » [Shreiber, 1995].

Cette définition propose l'extraction d'ontologie à partir d'une base de connaissances.

- **Définition 5**

« Une ontologie est un ensemble de termes hiérarchiquement structurés, conçu afin de décrire un domaine qui peut être utilisé comme un squelette de base pour les bases de connaissances»[Swartout, 1997].

- **Définition 6**

Ces deux dernières définitions sont regroupées dans celle de **Studer** comme

« *Spécification formelle et explicite d'une conceptualisation partagée* »

[Studer et al., 1998].

- ❖ **Formelle** : l'ontologie doit être lisible par une machine, ce qui exclut le langage naturel.

- ❖ **Explicite** : la définition explicite des concepts utilisés et des contraintes de leurs utilisations.

❖ *Conceptualisation* : le modèle abstrait d'un phénomène du monde réel par identification des concepts clefs de ce phénomène.

❖ *Partagée* : l'ontologie n'est pas la propriété d'un individu, mais elle représente un consensus accepté par une communauté d'utilisateurs.

4. Pour quelles raisons développer une ontologie

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 représente les dimensions scientifique et technique de la participation des ontologies dans le domaine de l'Intelligence Artificielle. Au fil des 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 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.

Aussi :

- Partager la compréhension commune de la structure de l'information entre les personnes ou les fabricants de logiciels.
- Permettre la réutilisation du savoir sur un domaine
- Expliciter ce qui est considéré comme implicite sur un domaine
- Distinguer le savoir sur un domaine du savoir opérationnel
- Analyser le savoir sur un domaine

D'après Roche : [Roche et al.,2003]

La figure 2 représente les domaines d'utilisations des ontologies.

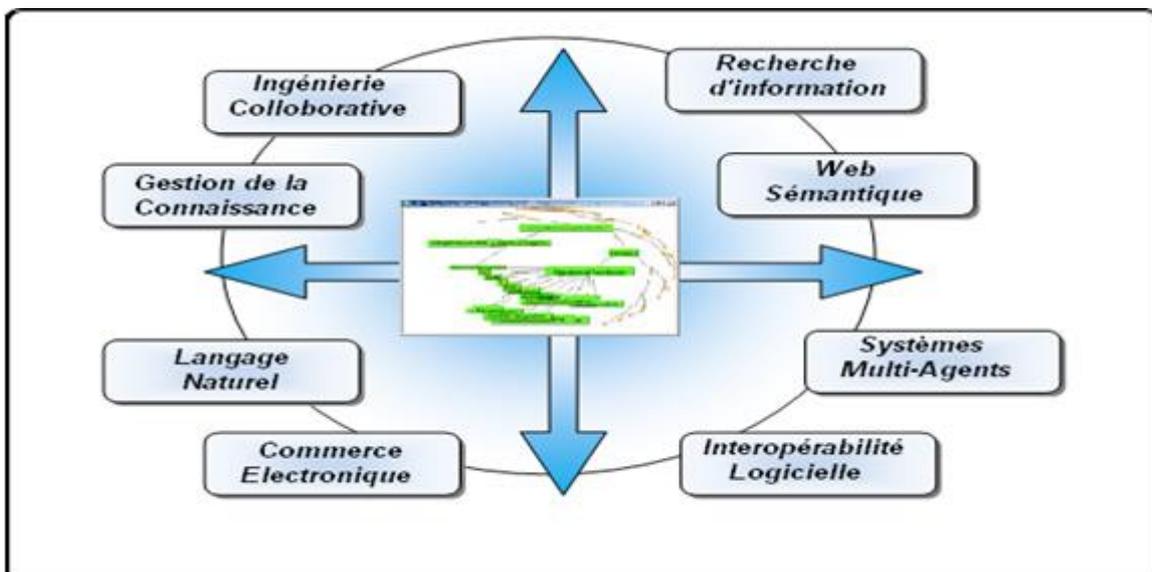


Figure 2: Utilisation des ontologies

5. Les éléments constitutifs d'une ontologie

Les ontologies définissent les termes et les relations entre eux, elles leur fournissent un vocabulaire commun. Bien 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.

Les composants d'une ontologie sont :

5.1. Concept

Un concept est une abstraction réunissant un certain nombre d'entités du « monde réel » (Aristote) qui sont ses instances.

Un concept est désigné par un label (terme, nom). Ce dernier est défini par ses différentes relations avec les autres concepts de l'ontologie, ses attributs et les contraintes qui lui sont associées, et il peut représenter plusieurs objets. Un concept peut donc décrire une tâche, une fonction, une action, une stratégie, un processus de raisonnement, etc.

Il peut se définir comme une entité composée de trois éléments distincts :

- Un terme : est un élément lexical qui permet d'exprimer le concept en langue naturelle et peut admettre des synonymes.
- Une extension : une extension est l'ensemble des objets manipulés à travers ce concept qui sont également appelés instances du concept.
- Une intension : appelée aussi notion. Elle spécifie la sémantique du concept à travers un ensemble de propriétés et des contraintes. Par exemple, le concept «personne» est caractérisé par un nom, un prénom et une adresse.

La *figure 3* représente le trigle sémantique.

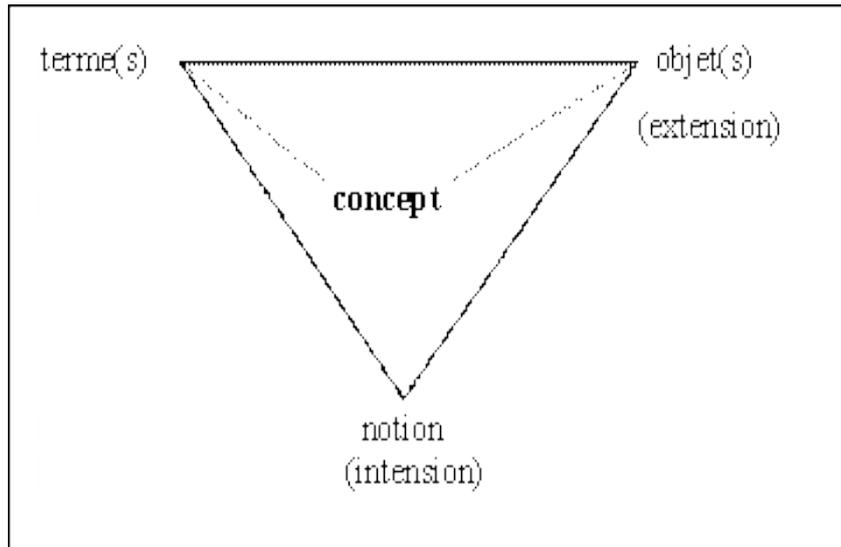


Figure 3: Triangle sémantique

5.2. Relations et fonctions

Les relations traduisent les associations (pertinentes) existant entre les concepts du domaine. Une relation est définie comme une notion de lien entre des entités, exprimée souvent par un terme, les liens sont classés en deux catégories : des liens hiérarchiques et des liens sémantique.

- ❖ Terme : représente la relation en langue.
- ❖ Extension : l'ensemble de réalisation effective d'une relation entre concepts.
- ❖ Intension : l'ensemble de propriétés et des attributs de la relation.
- ❖ Signature : est l'ensemble des concepts pouvant être liées par la relation

Par la suite nous présentons quelques propriétés représentent les intentions des concepts et des relations du domaine.

- **Les propriétés algébriques d'une relation** : Comme la symétrie, la réflexivité et la transitivité etc.
- **La propriété de subsumption entre concepts ou entre relations** : Un exemple d'une relation de subsumption : un concept C1 subsume un concept C2 si toutes les propriétés sémantiques de C1 sont aussi des propriétés sémantiques de C2. Autrement dit C2 est plus spécifique que C1.

- **La cardinalité d'une relation** : La cardinalité précise le nombre d'instances d'un concept qu'une relation peut lier.
- **La disjonction (incompatibilité)** : deux concepts sont disjoints si leurs extensions sont disjointes.
- **L'exclusivité** : Deux relations sont exclusives si quand l'une lie des instances des concepts, l'autre ne lie pas ces instances, et vice-versa. L'exclusivité entraîne l'incompatibilité.

Exemple : l'appartenance et le non appartenance.

- **L'équivalence** : deux concepts sont équivalents s'ils ont la même extension.
- **L'inverse** : Deux relations binaires sont inverses l'une de l'autre si lorsque une lie deux instances I1 et I2, l'autre lie I2 et I1. Exemple : les relations " a pour père " et " a pour enfant " sont inverses l'une de l'autre.

5.3. Axiomes

Permettent de modéliser des assertions toujours vraies, à propos des abstractions du domaine traduites par l'ontologie.

Ils permettent de combiner des concepts, des relations et des fonctions pour définir des règles d'inférences et qui peuvent intervenir, par exemple, dans la déduction, la définition des concepts et des relations, ou alors pour restreindre les valeurs des propriétés ou les arguments d'une relation.

5.4. Instances

Les individus constituent la définition extensionnelle de l'ontologie. Ils représentent des éléments singuliers véhiculant les connaissances à propos du domaine du problème. Elles sont définies en OWL.

6. Langages de représentation des ontologies

Les langages de représentation des ontologies sont des langages formels. Ils permettent l'expression d'ensemble de concepts et leurs relations conceptuelles, leur objectif est de :

- ❖ décrire les ressources et services présents sur le Web,
- ❖ exprimer des modèles abstraits (ontologies) de ces descriptions,
- ❖ échanger / partager ces descriptions sur le Web

Il existe plusieurs langages informatiques spécialisés dans la création et la manipulation des ontologies. Parmi ces langages nous citons :

6.1. KIF [Genesereth et al., 1992]

KIF est un langage basé sur les prédicats du premier ordre avec des extensions pour représenter des définitions et des méta-connaissances, la logique du premier ordre étant un langage de bas niveau pour l'expression d'ontologies.

Une extension du langage KIF, ONTOLINGUA, est utilisée dans le serveur d'édition d'ontologies, ONTOLINGUA du même nom.

6.2. RDF/RDFschéma¹

Le W3C a adopté le langage RDF (Ressource Description Framework) comme un des formalismes standards de représentation de connaissances sur le Web. Utilisant la syntaxe XML (Extended MarkupLanguage) qui constitue déjà un standard, le RDF permet de décrire des ressources Web en termes de ressources, propriétés et valeurs.

Une ressource peut être une page Web (identifiée par son URI, United Ressource Identifier) ou une partie de page (identifiée par une balise).

Les propriétés couvrent les notions d'attributs, relations ou aspects et servent à décrire une caractéristique d'une ressource en précisant sa valeur.

Les valeurs peuvent être des ressources ou des littéraux. RDF dispose d'une sémantique formelle analogue à celle des graphes conceptuels, c'est-à-dire identique à celle d'un fragment de la logique du premier ordre.

¹ <https://www.w3.org/RDF/>

Un schéma de base incluant les primitives sémantiques généralement utilisées, a ainsi été ajouté au RDF et constitue ce qu'on appelle le RDF SCHEMA (RDFS).

6.3. DAML + OIL

Dans l'optique d'une utilisation d'ontologies sur le Web, le langage RDF-Sa été enrichi par l'apport du langage OIL (OntologyInterchangeLanguage) qui permet d'exprimer une sémantique à travers le modèle des frames tout en utilisant la syntaxe de RDF-S.

OIL offre de nouvelles primitives permettant de définir des classes à l'aide de mécanismes ensemblistes issus des logiques de description (intersection de classes, union de classes, complémentaire d'une classe). Il permet également d'affiner les propriétés de RDF-S en contraignant la cardinalité ou en restreignant la portée.

6.4. OWL ²

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

Développé par le groupe de travail sur le Web Sémantique du W3C, OWL peut être utilisé pour représenter explicitement les sens des termes des vocabulaires et les relations entre ces termes.

Les activités de spécification de langages d'ontologies pour le Web sont le fruit du consortium W3C (World Wide Web) qui émet des recommandations : RDF(S), OWL.

OWL s'inscrit dans les langages du Web permettant de définir des Métadonnées sur les informations disponibles sur Internet. Basé sur RDF, OWL étend le pouvoir de description du langage de modélisation. Ainsi, si RDF-S (la partie sémantique de RDF) s'est attaché à définir des vocabulaires, le W3C s'est rapidement aperçu de la faiblesse du pouvoir d'expression de ce langage de modélisation qui ne permettait pas de pouvoir aisément mettre en relation des

² <https://www.w3.org/TR/2012/REC-owl2-overview-20121211/>

vocabulaires, des ontologies. Pour pallier cette faiblesse, OWL a été standardisé qui étend le pouvoir de RDF-S tout en lui restant compatible.

Pour cela, OWL a des moyens plus puissants pour exprimer la signification et la sémantique que XML, RDF, et RDF-S.

OWL permet donc de définir des ontologies qui sont l'association de définitions de termes d'un domaine particulier et des relations entre ces termes.

7. Opérations sur les ontologies

Dans cette section nous allons présenter quelques opérations qu'on peut appliquer sur les ontologies :

7.1. Intégration des ontologies

L'ontologie vise à constituer une représentation du monde réel qui puisse être acceptée par tous les membres d'une communauté mais les possibilités de représentation sont variables et par conséquent les ontologies diffèrent malgré les efforts de normalisation. Les différentes conceptualisations engendrent l'hétérogénéité des ontologies, une problématique de plus en plus explorée en raison de la multiplication de leur nombre et l'augmentation de leur accessibilité.

L'intégration d'ontologies représente une solution pour cette limite, elle permet la possibilité d'interopérabilité entre les ontologies.

L'intégration des ontologies signifie la construction d'une nouvelle ontologie en utilisant d'autres ontologies disponibles. Ces différentes ontologies font partie de la nouvelle ontologie.

7.1.1. Mapping d'ontologies

Le mapping entre deux ontologies « Ontologie1 ; Ontologie2 » signifie que pour chaque entité dans une ontologie « Ontologie1 », il faut trouver l'entité correspondante dans l'ontologie « Ontologie2 » avec un sens équivalent ou le sens le plus proche.

Une caractéristique importante de cette approche est qu'elle ne modifie pas les ontologies impliquées et qu'elle produit en sortie un ensemble de correspondances.

Le mapping d'ontologies a pour objectif la représentation des correspondances entre les ontologies. Cela permet, par exemple, d'interroger des bases de connaissances hétérogènes en utilisant une interface commune ou en transformant des données entre différentes représentations. [Mellal, 2007]

La figure suivante montre le principe de mapping entre les ontologies :

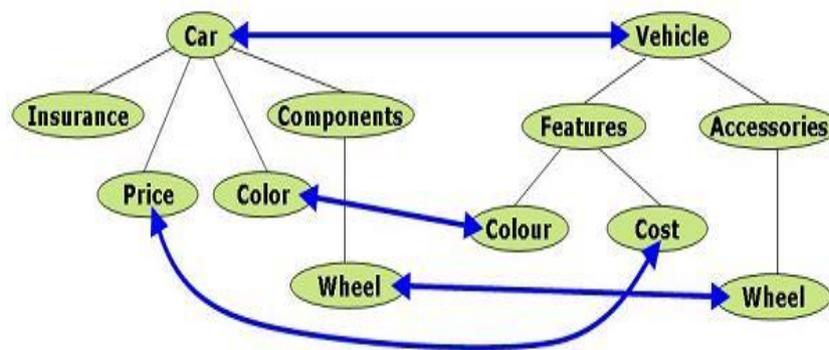


Figure 4: Mapping des ontologies

7.1.2. Fusion d'ontologies

La fusion d'ontologies permet de créer une nouvelle ontologie, appelée l'ontologie fusionnée, qui repose sur la capture des connaissances des ontologies d'origine et les concepts supplémentaires nécessaires pour réaliser cette fusion.

Le défi est alors d'assurer que toutes les correspondances et les différences entre les ontologies soient correctement prises en compte dans l'ontologie résultante [Elbyed, 2009].

Ce type d'approche est généralement utilisé, dans le cadre d'intégration de données, pour obtenir une ontologie globale qui sert d'interface pour un certain nombre d'ontologies locales.

La figure 4 montre le principe de la fusion :

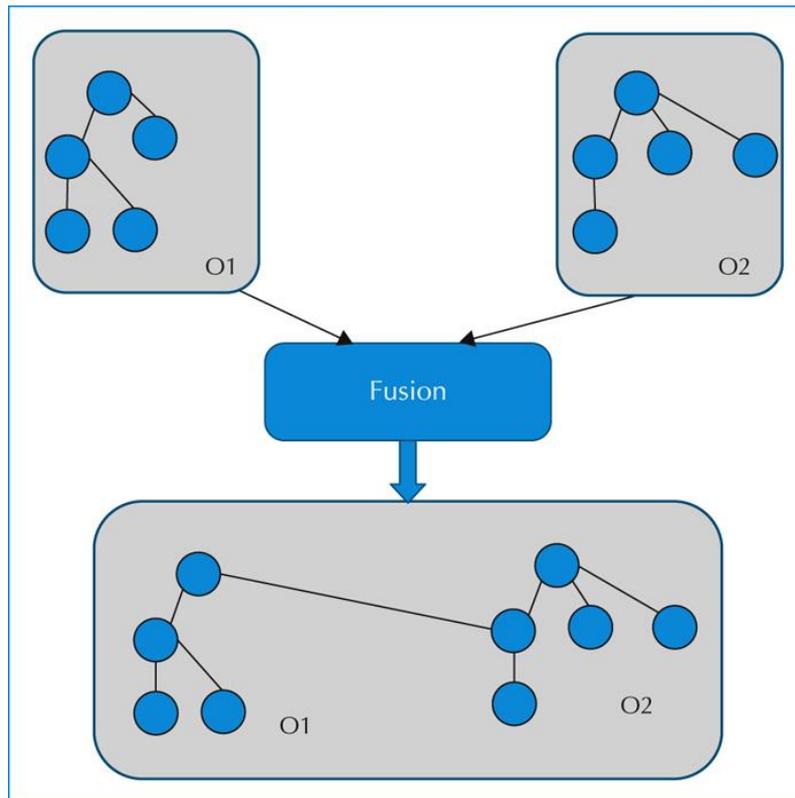


Figure 5: Principe de la fusion d'Ontologies.

7.2. Alignement d'ontologies

L'alignement des ontologies a pour objectif d'établir des relations sémantiques entre différentes ontologies.

Le principe de l'alignement est illustré dans la *figure 6*

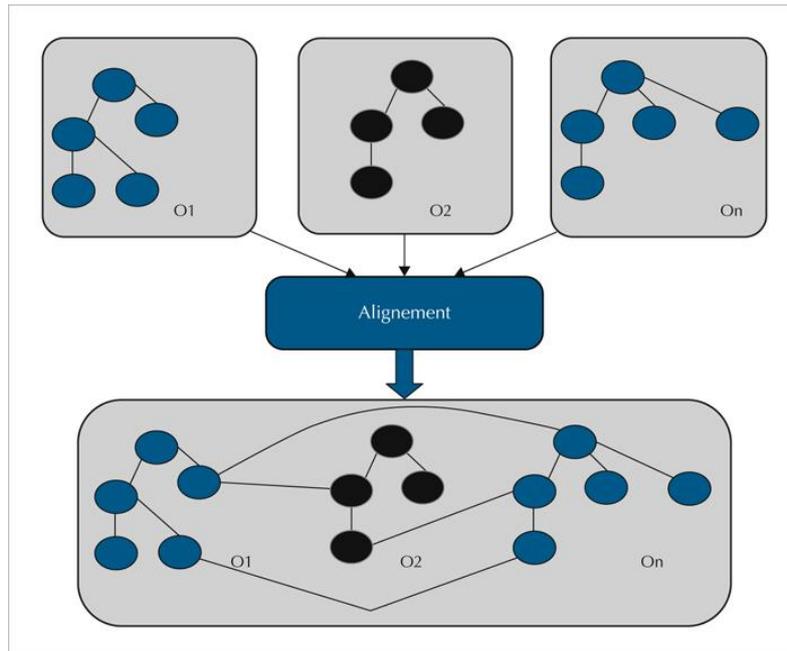


Figure 6: Principe d'alignement d'ontologies

7.3. Partitionnement d'ontologies

Le partitionnement d'ontologies divise une ontologie en un ensemble de sous-ensembles, chaque sous-ensemble étant appelé une partition.

Le partitionnement d'ontologies peut être utilisé dans des applications telles que l'alignement d'ontologies, la fusion d'ontologies et la synthèse de texte basée sur l'ontologie.

Le partitionnement d'ontologies est généralement applicable pour diviser de grandes ontologies et agir sur des sous-ontologies pour augmenter la qualité des partitions, dont le but est de rendre l'application de différentes opérations sur ces ontologies pratique et efficace.

7.4. Maintenance d'ontologies

Les ontologies sont confrontés de façon continue à un problème d'évolution, car les termes techniques relatifs à un domaine particulier changent et évoluent de façon perpétuelle. Par conséquent, les ontologies de domaine doivent être maintenues pour faire face aux incomplétudes et aux erreurs.

La maintenance de l'ontologie est une étape liée à la construction de l'ontologie, plus précisément, c'est la dernière phase des activités techniques de la construction.

La maintenance de l'ontologie est une tâche très importante, car elle permet de mieux structurer le contenu sémantique de l'ontologie, et assurer une meilleure représentation des connaissances représentées par l'ontologie à l'aide des connaissances à jour, et garantir, par conséquent, l'utilité et la fiabilité de l'ontologie. Cette tâche est d'autant plus compliquée quand les ontologies à analyser sont de grande taille.

8. Cycle de vie global d'une ontologie

Les ontologies étant 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. En particulier, les ontologies doivent être considérées comme des objets techniques évolutifs et possédants un cycle de vie qui nécessite d'être spécifié. [Blazquez et al., 1998]

La vie d'une ontologie passe par les états suivants :

- ❖ Évaluation des besoins construction, diffusion, et utilisation. Après chaque utilisation significative, l'ontologie et les besoins sont réétudiés et l'ontologie peut être étendue.
- ❖ La phase de construction peut être décomposée en 3 étapes : conceptualisation, ontologisation, opérationnalisation. L'étape d'ontologisation peut être complétée d'une étape d'intégration au cours de laquelle une ou plusieurs ontologies vont être importées dans l'ontologie à construire. [Fürst, 2002]

Le schéma du cycle de vie global d'une ontologie est présenté dans la figure 7.

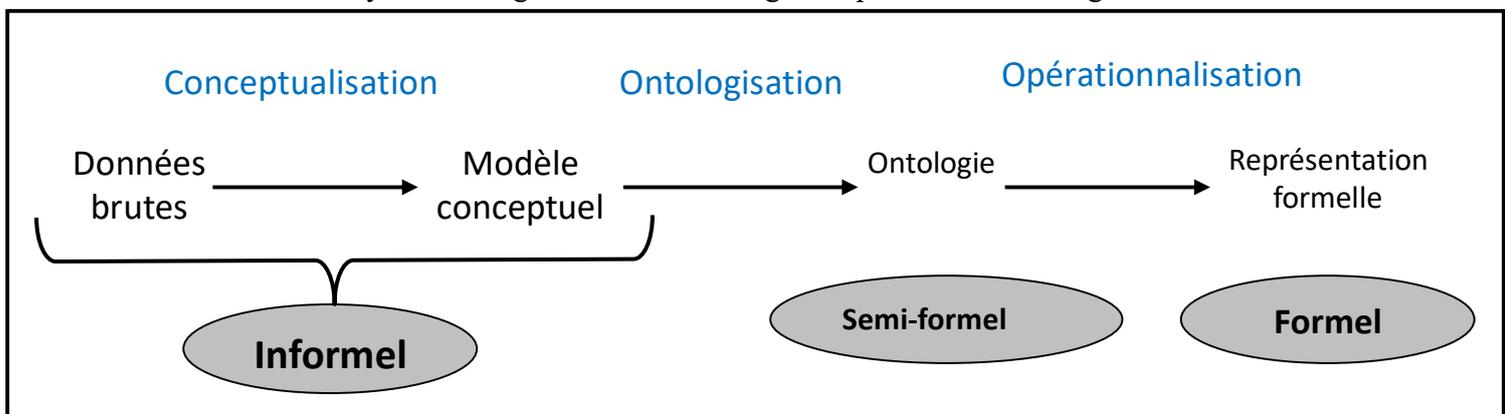


Figure 7: Cycle de vie global d'une ontologie [Fürst, 2002]

9. Conclusion

Dans ce premier chapitre, nous avons donné une vision générale sur les ontologies par différentes définitions. Dans le chapitre suivant nous allons parler des méthodes d'intégration d'ontologies qui ont été utilisées.

Chapitre II :

Etat de l'art sur l'intégration des ontologies

1. Introduction

Les ontologies ont été reconnues comme un élément essentiel pour le partage des connaissances. En définissant des concepts associés à des zones particulières, les ontologies permettent à la fois de décrire le contenu des ressources à intégrer, et à clarifier le vocabulaire utilisé dans les requêtes des utilisateurs.

Ainsi, si les connaissances et les données doivent être partagées, il est Indispensable d'établir des correspondances sémantiques entre les ontologies concernées.

L'alignement d'ontologies est une tâche importante dans l'intégration car elle permet la prise en compte conjointe de ressources décrites par différentes ontologies, elle est complexe basée sur des mesures de similarité. De nombreuses études ont été réalisées, mais la plupart du temps, une seule mesure est insuffisante pour détecter une similitude. Différentes approches combinant plusieurs mesures successives ont été proposées : cette combinaison est effectuée afin de donner des approches plus prometteuses.

Dans ce chapitre nous commençons par définir les différentes mesures de similarité, ensuite nous présentons les différentes approches et méthodes qui existent pour l'intégration des ontologies.

A la fin nous comparons et analysons ces méthodes.

2. Les techniques de mesures de similarité

Il existe plusieurs techniques de mesures de similarité dans le domaine de l'ingénierie des connaissances, nous allons faire le point sur quelques-unes.

La *figure 1* représente un résumé des différentes mesures de similarité, catégorisées selon les techniques utilisées.

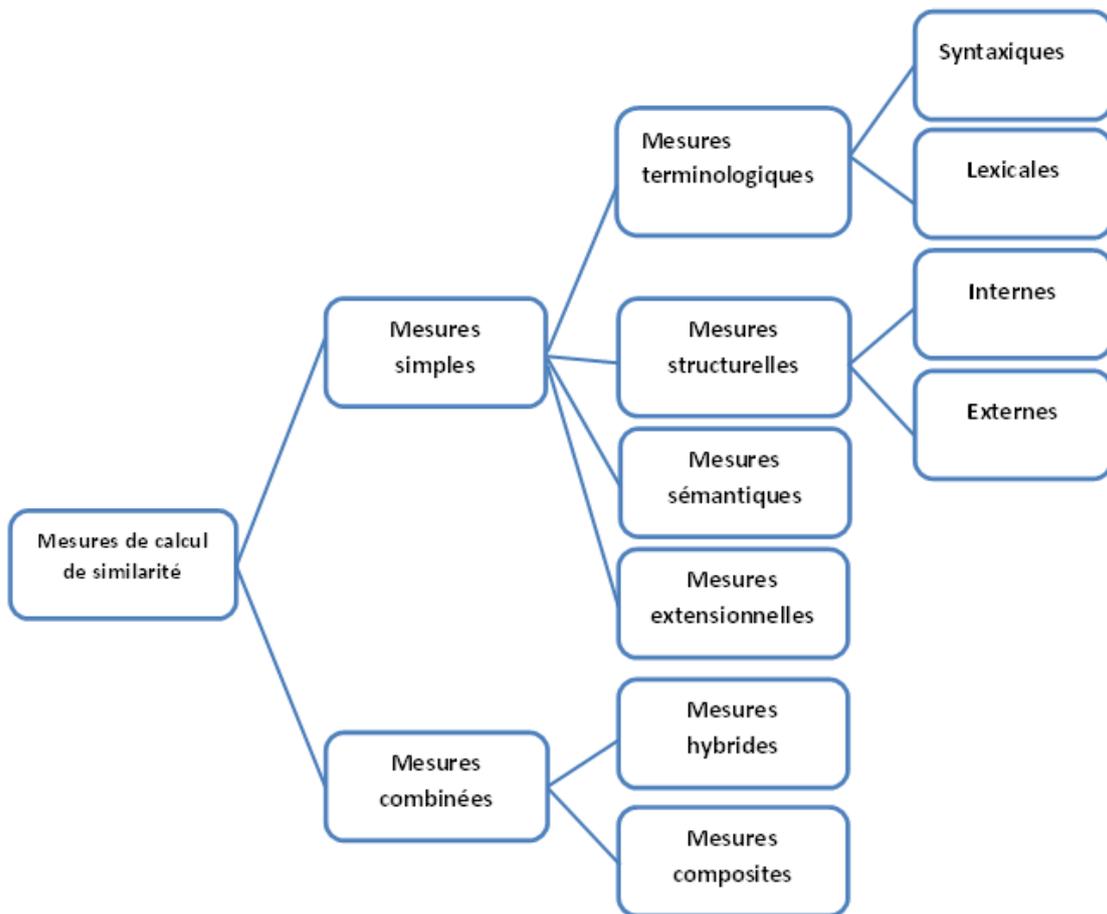


Figure 8: Mesures de calcul de similarités [Euzenat, 2008].

2.1. Mesure de similarité terminologique

Ces mesures reposent sur la comparaison des termes ou des chaînes de caractères ou bien les textes. Elles sont exploitées pour calculer la valeur de la similarité des entités textuelles, telles que des noms, des étiquettes, des commentaires, des descriptions, etc. Nous trouvons dans cette

méthode deux approches : l'approche syntaxique et l'approche lexicale, appelée aussi linguistique [Monge et al, 1996]

2.1.1. Mesure de similarité lexicale

En linguistique, la similarité lexicale est la mesure de degré de ressemblance entre des séries de mots appartenant à deux termes donnés. Une similarité lexicale de 1 (ou 100%) correspondrait à un recouvrement total entre les vocabulaires, tandis que 0 signifie qu'il n'y a pas de mots communs.

2.1.2. Mesure de similarité syntaxique

Les techniques de similarité syntaxique consistent à effectuer des mesures de similarité en se basant sur les étiquettes et les labels correspondants aux couples de concepts à aligner en comparant caractère par caractère. Deux éléments sont considérés comme sémantiquement proches si les termes qui les désignent sont syntaxiquement proches.

2.2. Mesure de similarité structurelle

La similarité structurelle entre deux éléments est calculée en fonction de leurs informations structurelles au sein de la hiérarchie ontologique.

2.3. Mesures de similarité sémantique

La similarité sémantique est une évaluation du lien sémantique entre deux concepts dont le but est d'estimer le degré par lequel les concepts sont proches dans leur sens.

La définition donnée par Lin de la similarité sémantique repose sur trois suppositions. La similarité entre deux concepts est liée aux caractéristiques qu'ils ont en commun (plus ils ont de caractéristiques communes, plus les concepts sont similaires) et à leurs différences (plus deux concepts sont différents, moins ils sont similaires). La similarité maximale est obtenue lorsque deux concepts sont identiques [Mellal, 2007].

2.4.Mesures de similarité extensionnelle

Elles résultent de la similarité entre deux entités qui sont notamment des concepts ou des classes tout en analysant ainsi leurs extensions (leurs ensembles d'instances). Chaque instance peut être représentée par un vecteur de noms et/ou de valeurs. Des calculs de similarités entre vecteurs permettent de comparer les instances [Ziani et al., 2010].

2.5.Mesures de similarité Combinées

Ces mesures combinent plusieurs mesures lorsqu'une seule est insuffisante [Leacock et al., 1998].

Il existe deux types de combinaison :

2.5.1. La combinaison séquentielle (hybride)

La méthode la plus simple pour combiner les mesures est l'utilisation séquentielle de ces dernières en choisissant un ordre d'exécution.

Par exemple, nous choisissons de lancer une mesure terminologique avant de lancer une autre mesure structurelle ou sémantique [Elbyed, 2009].

2.5.2. La combinaison parallèle (composite)

Une autre manière de combiner les résultats des différentes mesures (c.-à-d. les valeurs de similarité) consiste tout d'abord à lancer parallèlement plusieurs mesures, puis par la suite à combiner leurs résultats [Elbyed, 2009].

3. Méthodes d'intégration existantes :

Plusieurs méthodes de mapping et de fusion ont été créées, nous avons sélectionné quelques unes afin de faire une comparaison entre elles selon plusieurs critères de comparaison.

3.1.S-Match [Giunchiglia et al., 2004]

S-Match est une correspondance sémantique open source offrant plusieurs correspondances sémantiques algorithmes et installations pour le développement de nouvelles unes. Il comprend des composants pour transformer des structures d'arbre en ontologies légères, où chaque étiquette du nœud dans l'arbre est traduite en formule propositionnel de description logique (DL), qui codifie de manière univoque la signification du nœud.

S-Match contient la mise en œuvre de la correspondance sémantique de base, la correspondance sémantique minimale, et la structure préservant la correspondance sémantique Algorithmes (SPSM). L'algorithme d'appariement sémantique de base est un algorithme d'appariement d'usage général, très personnalisable et adapté à de nombreuses applications. L'algorithme minimal d'appariement sémantique exploite des connaissances supplémentaires encodées dans la structure de l'entrée et il est capable de produire un mapping minimal et un mapping maximale. Le SPSM est un type d'appariement sémantique qui produit un score de similarité et un mapping tout en préservant les propriétés structurelles

- (i) correspondances individuelles entre les nœuds sémantiquement liés.
- (ii) les fonctions sont appariées à des fonctions et les variables à des variables.

3.2.Falcon-AO [Hu et al., 2006]

Une méthode basée sur le clustering d'ontologies, appelée Falcon-AO a été proposée dans [Hu et al., 2006]. La première étape du processus se base sur le partitionnement d'ontologie en un ensemble de blocs en utilisant la notion des liens pondérés à l'aide de deux techniques de similarité : la technique syntaxique (basée sur la comparaison des labels des éléments des ontologies) et la technique structurelle (basée sur la comparaison de similarités structurelles qui

exploite les alignements générés par la technique syntaxique et/ou fournis en entrée de la technique). Le calcul des liens pondérés entre deux concepts s'opère comme suit :

$$poids(c_i; c_j) = \begin{cases} f = \alpha * sim_{struct}(c_i, c_j) + (1 - \alpha)sim_{syn}(c_i, c_j) & si f > \varepsilon \\ 0 & \text{sinon} \end{cases}$$

Où : $sim_{struct}(c_i, c_j)$ est la similarité structurelle entre $c_i; c_j$ basée sur la mesure de Wu et Palmer [Wu et al.1994] ; sim_{syn} est la similarité syntaxique ; ε et α sont deux valeurs fixées par l'utilisateur pour la variation de poids attribués aux liens de pondération.

L'algorithme commence avec n blocs comportant chacun un seul concept, ensuite une méthode itérative est appliquée. A chaque itération, l'algorithme choisit le bloc qui a la cohésion maximale (la cohésion désigne la somme des poids des liens reliant les concepts du même bloc) et le bloc qui a la valeur de couplage maximale (Le couplage désigne la somme des poids des liens reliant les concepts appartenant à deux blocs distincts) avec ce premier bloc.

$$Cohésion(B_i) = goodness(B_i, B_i)$$

$$Couplage (B_i, B_j); B_i \neq B_j$$

$$goodness(B_i, B_j) = \frac{\sum_{c_i \in B_i, c_j \in B_j} poids(c_i, c_j)}{Taille(B_i) * Taille(B_j)}$$

L'algorithme fusionne ces deux blocs, les remplace par le cluster résultat et met à jour les nouvelles valeurs de couplage et de cohésion. Le critère d'arrêt est satisfait, lorsque les blocs construits ont atteint une taille maximale fixée au préalable par l'utilisateur.

Dans la phase d'alignement, il faut sélectionner les paires des blocs contenant plus d'entités en commun, la valeur de proximité entre deux blocs doit être supérieure à un seuil. La similarité des entités est calculée en utilisant une technique terminologique. La dernière étape consiste à aligner les entités des clusters similaires ou proches en utilisant les techniques structurelles et linguistiques.

Cette méthode réussit à clusteriser les ontologies de grande taille. Cependant la phase de traitement des blocs similaires est très coûteuse en termes de temps vu que le calcul de proximité

des blocs qui est basé sur les éléments partagés par les deux ontologies, doit être procédé sur toutes les paires de blocs possibles des deux ontologies.

3.3. TaxoMap[Hamdi et al., 2008]

Deux méthodes ont été proposées dans [Hamdi, et al., 2008]. La première méthode commence par partitionner l'ontologie cible en utilisant le même algorithme de partitionnement adopté par Falcon-OA. Par conséquent, le partitionnement de l'ontologie source est réalisé en suivant celui de l'ontologie cible. En effet, la méthode détermine pour chaque bloc dans l'ontologie cible, l'ensemble des ancres (les entités syntaxiquement similaires partagées par les deux ontologies) qui constitueront le futur cluster dans l'ontologie source. Finalement, les entités des clusters similaires sont mises en correspondance. La deuxième méthode se distingue par le Co-clustering des ontologies, c.à.d. le clustering des deux ontologies en même temps. Pour simuler le parallélisme, l'ontologie cible est partitionnée en favorisant la fusion des blocs partageant des ancres avec la source, et on partitionne la source en favorisant la fusion des blocs partageant des ancres avec un même bloc généré pour la cible. L'algorithme de partitionnement est celui proposé par Falcon-OA.

3.4. Thefca-ontmergemethod(Methode basée sur l'ACF) [Guan-yu et al.,2010]

Stumme et Maedche ont proposé une méthode nommée FCA-Merge d'abord, elle extrait les instances basées sur la technologie de traitement du langage naturel et génère deux contextes formels en calculant, puis utilise le FCA-Merge algorithme de base pour dériver un contexte commun et calculer un réseau de concept, finalement il génère la dernière ontologie fusionnée basée sur le réseau de concept. Cette approche a résolu le problème de la sémantique manquante dans les systèmes actuels, mais à chaque fois qu'il fusionne des ontologies, il doit collecter les documents pertinents du domaine, et la flexibilité est peu forte, l'intervention de l'homme dans le processus de fusion est importante et le processus est assez compliqué. Le travail de [Gu

Tao.,2003]met en avant une méthode pour construire un contexte formel, il utilise les attributs et les objets dans un contexte formel, respectivement pour exprimer les attributs et les concepts de l'ontologie. Cette approche peut considérablement réduire la complexité de la construction du contexte formel à partir des ontologies existantes. Ainsi, le travail de **[Guan-yu et al.,2010]**propose une nouvelle méthode FCA-OntMerge de la fusion d'ontologies basée sur une analyse de concept formel, convertit d'abord les ontologies pour être exprimer dans le même langage, puis convertit ces ontologies en contexte formel et faire le mappages attributs, Enfin, la fusion des contextes formels et la génération de réseau de concepts ultime.

3.5.Méthode basée sur la classification **[Maiz et al., 2010]**

L'idée principale de cette approche est de combiner la puissance de l'approche statistique représentée par l'algorithme de classification hiérarchique avec le mécanisme d'inférence offert par le langage sémantique OWL-DL. Pour générer une ontologie globale fusionnée automatiquement, il faut appliqué l'algorithme de classification sur différentes catégories d'entités ontologiques (concepts, propriétés, instances) afin de trouver des classes d'entités équivalentes appartenant à différentes ontologies locales, comme illustré à la **figure (9)**.Pour chaque classe, il y aura une déduction pour découvrir les nouveaux axiomes représentant les nouvelles relations entre entités de la même classe ou entre différentes classes de la même catégorie. Après cela, il faut utilisés différentes classes et axiomes pour construire l'ontologie globale. La méthodologie commence par aligner les ontologies locales en recherchant des entités similaires appartenant à des ontologies différentes. Ensuite, l'utilisation des résultats de l'alignement de l'ontologie pour fusionner automatiquement des ontologies locales.

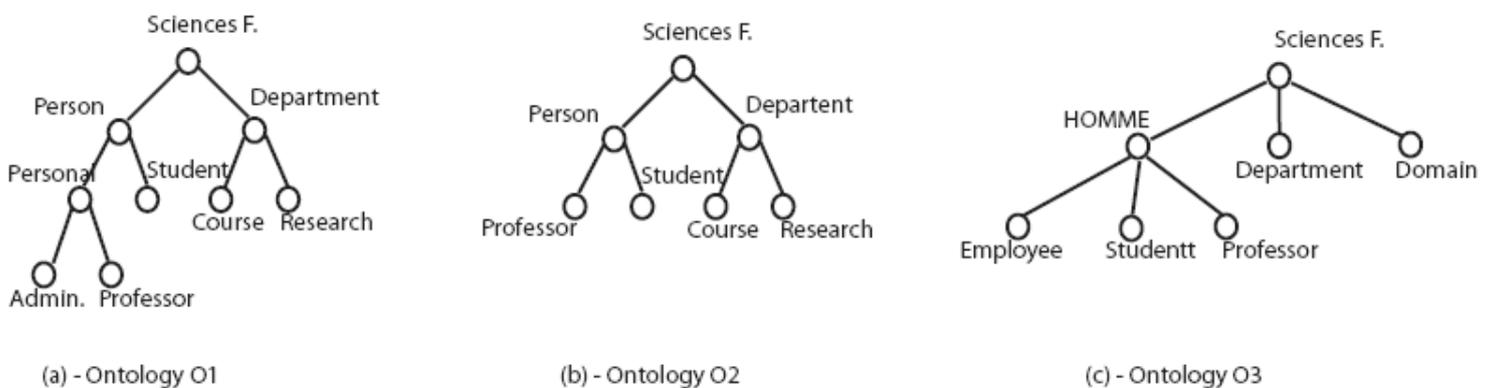


Figure 9: Exemple de trois ontologies locales du même domaine

3.6.COMA++ [Massmann et al., 2011]

Un système dédié à l'alignement des graphes de schémas larges a été proposé dans [Massmann et al., 2011]. La méthode proposée se base sur l'idée Diviser pour régner. Une ontologie est décomposée en des clusters (appelés fragments) qui seront, par la suite, mis en correspondance. Le clustering de l'ontologie est basé sur un ensemble de règles heuristiques prédéfinies pour la production d'un ensemble de clusters de taille réduite. Chaque cluster est identifié par un concept racine qui sera utilisé pour la comparaison des clusters dans la phase d'alignement de clusters. En effet, deux clusters dont les deux concepts racine sont sémantiquement similaires, ceci implique la similarité des deux clusters. L'alignement sera ensuite mené sur les entités appartenant aux clusters jugés similaires en utilisant la technique structurelle et terminologique.

3.7.Méthode de Ding et al [Ding et al., 2013]

Dans ce travail, les auteurs utilisent l'algorithme K-means pour réaliser l'alignement entre plusieurs attributs.

En premier lieu, les auteurs convertirent les attributs en points, ensuite exécutent l'algorithme K-means pour partitionner les attributs à plusieurs clusters. Les attributs dans le même cluster ont la même sémantique.

Dans cet algorithme, le nombre de K objets est choisi aléatoirement comme des centres de clusters ensuite la méthode TF/IDF est utilisée pour calculer le poids. Aussi le modèle de l'espace vectoriel est appliqué comme métrique de calcul de distance entre les points d'attributs.

3.8.Fusion par enrichissement sémantique (MOnSE) [Fareh et al.,2013]

Les auteurs ont proposé et développé le système MOnSE qui se concentre sur :

- L'ajout d'une dimension sémantique par enrichissement préalable des ontologies, ce qui se fait en définissant un ensemble d'annotations de métadonnées, aux concepts originaux de deux ontologies.
- La méthode est basée sur le calcul de la similarité sémantique entre les concepts d'ontologies. Elle est basée sur une combinaison de méthodes pondérée pour calculer les similitudes existantes avec des spécificités différentes.
- L'automatisation de l'algorithme pour fusionner les ontologies d'origine afin de décharger le plus possible l'utilisateur.

Avant que l'ontologie puisse être fusionnée via MOnSE, elle doit faire l'objet d'un enrichissement par un expert. Au cours de l'enrichissement, les concepts de l'ontologie seront situés dans leur contexte. L'expert fournit une définition et des synonymes qui collent autant que possible sur le contexte à laquelle se rapportent leurs ontologies, à l'aide de la terminologie lexicale WordNet, ou en faisant référence aux connaissances de l'expert. Cet enrichissement aura un impact majeur sur la similarité entre des concepts des ontologies différentes.

Le processus de fusion selon MOnSE est le suivant :

- Etape 1 : Construction d'un thésaurus global - la première étape consiste à construire, à partir des ontologies initiales, un thésaurus incluant ces différents concepts et leurs attributs, instances et relations afin de poursuivre ces différents traitements conduisant à la fusion.
- Etape 2 : Calcul des mesures de similarité - Deuxièmement, calculez la similarité entre les concepts de deux ontologies afin de mesurer leur degré de correspondance et de faciliter le regroupement et la fusion de ces concepts.
- Etape 3 : Changer le thésaurus global - après avoir calculé les similitudes entre les ontologies, une modification du thésaurus aura lieu afin qu'il représente un aperçu complet de la nouvelle ontologie, en créant une nouvelle hiérarchie entre les différents concepts.
- Etape 4 : Validation et génération d'ontologies fusionnées- la dernière étape consiste à construire l'ontologie finale à partir du thésaurus modifié.

3.9.Méthode de Setti et al [Setti et al., 2015]

Dans le but de partitionner une ontologie volumineuse donnée en modules de haute qualité, Setti et al, ont proposé une approche de partitionnement qui introduit une amélioration de

l'algorithme de classification k-means, en se basant sur une nouvelle mesure de similarité sémantique « Dennai ». C'est une amélioration de la mesure de similarité structurelle : Wu et Palmer.

L'approche consiste à extraire d'abord tous les constructeurs de l'ontologie (classes, axiomes, relations et propriétés), puis à convertir l'ontologie représentée en OWL sous forme d'un graphe. Après, différentes mesures de similarité sont calculés d'un nœud donné vers tous les nœuds du graphe. Il s'agit de déterminer le chemin le plus court entre un nœud donné et un autre nœud du graph. Cette étape génère différentes matrices de mesures de similarité. Ensuite, deux algorithmes de clustering sont utilisés. Le premier est introduit pour adapter l'algorithme traditionnel au clustering d'ontologies. Ensuite il subit des modifications pour intégrer les différentes mesures de similarité. Le deuxième algorithme a pour but d'éliminer les clusters inutiles générés auparavant. Enfin, un processus de validation du partitionnement est programmé comme une perspective.

Deux séries d'expérimentations ont montré que cette approche réduit le temps et l'espace requis pour traiter une ontologie, et produit des partitions de haute qualité. L'amélioration de l'algorithme k-means proposée et l'utilisation de la nouvelle mesure de similarité Dennaidonnent de meilleures partitions et des clusters significatifs que l'algorithme k-means traditionnel.

4. La comparaison des méthodes

Nous allons faire deux comparaisons, la première selon les mesures de similarité utilisées et la deuxième selon les critères du passage à l'échelle.

4.1. Comparaison des méthodes selon les mesures de similarité utilisées

Dans ce qui suit nous définissons d'abord les critères de comparaison qui sont basés sur les mesures de similarité, ensuite faire une illustration d'un tableau de comparaison qui traite et compare les méthodes choisies par rapport aux critères, ensuite nous analysons le tableau pour déterminer les points forts et les points faibles de chaque méthode afin d'arriver à une conclusion.

4.1.1. Critère de comparaison

Dans ci-dessous une brève explication des critères utilisés dans le Tableau 3.

❖ **Langage utilisé**

Ce critère représente le langage utilisé en entrée du système (RDF, OWL ...etc.), il existe plusieurs langages de représentation des ontologies qui offrent des structures de données adaptées à la représentation des concepts.

❖ **Mesure de similarité**

Représente les types de mesures de similarité utilisées pour faire la comparaison entre les concepts ontologies.

❖ **Techniques de mesure de similarité**

Ce critère nous indique les techniques utilisées par chaque méthode, et il existe un grand nombre de technique dans la littérature.

❖ **Ressources externes**

Ce critère représente l'utilisation des ressources externes dans le processus d'alignement, les ressources externes sont les supports utilisés pour faire les tests.

❖ **Intégration**

Ce critère représente le type d'intégration utilisé par le système (mapping ou fusion).

❖ **Automatisation**

C'est un critère qui nous montre si le système fonctionne automatiquement, manuellement par l'intervention de l'utilisateur, ou un fonctionnement semi-automatique ; intervention de l'utilisateur dans quelques taches

4.1.2. Tableau de comparaison selon les mesures de similarité utilisées

	Langage utilisé	Mesure de similarité	Technique de similarité	Ressource externe	Intégration	Automatisation
S-Match [Giunchiglia et al., 2004]	XML schéma, OWL	Syntaxique, Sémantique	String-Based, Language-based	Wordnet	Mapping	Auto
Falcon-AO [Hu et al.,2006]	RDF,OWL	Syntaxique Structurelle	StringBased	Wordnet	Mapping	Auto,
Méthode basée sur l'ACF [Guanyu et al.,2010]	OWL	Syntaxique, Linguistique, Instances	Treillis de Galois, Jena : outil d'analyse d'ontologie	Wordnet	Fusion	Semi-Auto
Méthode basée sur la classification [Maiz et al.,2010]	OWL-DL	Sémantique, Terminologique	Inférence-logique	aucun	Fusion	Semi-Auto
Coma++[Massmann et al.,2011]	XML Schéma, RelationalSchema OWL	Syntaxique, Structurelle (descendants, ascendants)	String-Based, Language-Based, Data-Types	Thesaurus	Mapping	Semi-Auto

	Langage utilisé	Mesure de similarité	Technique de similarité	Ressource externe	Intégration	Automatisation
TaxoMap [Hamdi et al.,2008]	OWL	Syntaxique Structurale	String-Based, Language- Based	-	Mapping	Auto,
Méthode de Ding et al [Ding et al.,2013]	Schéma synthétique	Syntaxique	Cosine	-	Mapping	Auto
Méthode d'enrichissement sémantique [Fareh et al.,2013]	OWL	Structurelle Terminologique Sémantique	String Based, Wu et Palmer	Wordnet	Fusion	Auto
Méthode de Setti et al [Setti et al.,2015]	OWL	Syntaxique, Structurelle	Tbk,Jaccard, Cosine, Dice, Euclidienne, Wu &Palmer,Mesured'édition, Denna	-	Mapping	-

-le critère n'est pas cité dans l'article de base

Tableau 1: Tableau de comparaison selon les mesures de similarité utilisées

4.1.3. Analyse

❖ La plupart des méthodes utilisent le langage de représentation des ontologies, le standard OWL, qui exprime beaucoup de contraintes ce qui fait de lui un langage riche par rapport à RDF qui ne permet de représenter que les concepts et les propriétés, on peut dire donc que RDF est moins riche que OWL, et il est utilisé dans un certain arbre de système.

❖ L'utilisation de plusieurs mesures augmente la précision du système, les mesures terminologiques ne se basent pas sur la sémantique ce qui les donne une faible précision, tandis que les mesures structurelles se basent beaucoup plus sur les données externes (propriétés) et internes (concepts) pour avoir plus d'exactitude, certaines méthodes utilisent une combinaison de plusieurs mesures dites complémentaires afin d'aboutir à des résultats plus précis.

❖ La majorité des méthodes utilisent le Wordnet comme ressource externe (S-Match [Giunchiglia et al., 2004], Falcon-AO[Hu et al.,2006]), d'autres utilisent un thesaurus (COMA++)

❖ La majorité des méthodes proposées dans la littérature traitent le problème de mapping mais pour la fusion nous trouvons un nombre limité de méthodes lorsqu'il s'agit de manipuler des ontologies assez volumineuses.

❖ La plupart des méthodes récentes sont automatiques mais il y a des méthodes qui sont semi-automatiques qui nécessitent l'intervention de l'expert dans une tâche de l'opération (calcul de mesure de similarité, Fusion).

❖ Dans la méthode de [Ding et al., 2013], les auteurs ont basé sur la mesure syntaxique, ils ont négligé l'utilisation des mesures structurelles et sémantiques qui permettent une forte similarité entre les concepts.

4.2. Comparaison des méthodes selon les critères du passage à l'échelle utilisé

Dans ce qui suit nous procéderons par faire une explication des critères de comparaison que nous allons utiliser, ensuite nous ferons un tableau comparatif de certaines méthodes trouvées dans la littérature, et nous finirons par faire une analyse du comparatif, et faire une conclusion.

4.2.1. Critère de comparaison

Nous allons faire une brève présentation sur les critères utilisés dans le Tableau 4

❖ **Operations**

Ce critère représente le type d'intégration réalisé par le système, pour déterminer s'il s'agit d'un mapping ou d'une fusion d'ontologie.

❖ **Passage à l'échelle**

Ce critère nous indique si la méthode citée traite des ontologies volumineuses.

❖ **La technique utilisée**

Ce critère représente la technique utilisée pour le traitement de l'intégration des ontologies volumineuses.

❖ **Ontologies de test**

Ce critère détermine la taille des ontologies sur lesquelles le travail a été réalisé.

❖ **Critère d'évaluation**

Ce critère précise les critères utilisé pour l'évaluation de la méthode d'intégration tel que : la précision et le rappel.

4.2.2. Tableau de comparaison selon le passage à l'échelle utilisé

	Operations	Passage à l'échelle	La technique utilisée	Ontologies de test	Critère de test
S-Match [Giunchiglia et al., 2004]	Mapping	Non	-Exécution séquentielle des techniques de similarité	lightweight	-
Falcon-AO [Hu et al.,2006]	Mapping	Oui	Clustering Agglomératif	-large-scale ontologies -Russia12. Et TourismAB,	-
Méthode basée sur l'ACF [Guanyu et al.,2010]	Fusion	Non	-Treillis de Galois -Jena : outil d'analyse d'ontologie	-	-
Méthode basée sur la classification [Maiz et al.,2010]	Fusion	Oui	classification hiérarchique et inférence	-	-
Coma++ [Massmann et al.,2011]	Mapping	Oui	-Clustering basé sur la structure. -Utilisation des règles heuristiques	Cinq XML shemas : -CIDX -Excel -Noris -Paragon -Apartume	Précision

	Operations	Passage à l'échelle	La technique utilisée	Ontologies de test	Critère de test
TaxoMap [Hamdi.,2012]	Mapping	Oui	-Pattern-Based Mapping Refinement -Clustering Agglomératif	Large ontologies	
Methode de Ding et al [Ding et al.,2013]	Mapping	Non	Clustering	05 librairies en ligne	Précision et rappel
Methode d'enrichissement sémantique [fareh et al.,2013]	Fusion	Non	-	Benchmarks de l'OAEI	Précision et rappel
Methode de Setti et al [Setti et al.,2015]	Mapping	Oui	Clustering : k-meansamélioré	TAMBIS ontology (393 classes+relations+597 axiomes) Photographyontology (185 classes+268 relations + différent types d'objets + propriétés)	- Cohesion - density

-le critère n'est pas cité dans l'article de base

Tableau 2: Tableau de comparaison selon les critères du passage à l'échelle utilisé

4.2.3. Analyse

❖ Un très grand nombre de méthodes qui sont proposées dans la littérature traitent le problème de mapping mais pour le problème de fusion on trouve peu de méthodes qui la traitent. Cependant la plupart de ces méthodes sont récentes et ne traitent pas le passage à l'échelle.

❖ La plupart des méthodes de mapping traite le passage à l'échelle contrairement aux méthodes de fusion qui non seulement ne sont pas nombreuses mais aussi une seule parmi les trois méthodes citées traite le problème de passage à l'échelle, sans donner de détails sur les ontologies testées.

❖ Les ontologies de test diffèrent dans chaque méthode, pour les méthodes de mapping qui traitent le passage à l'échelle le test a été fait sur des ontologies de grande taille. à. d. des ontologies de milliers de concepts et relations mais dans les méthodes de mapping les ontologies de test sont des ontologies de l'OAEI et qui sont des ontologies de base pas nécessairement de grande taille.

❖ Dans la majorité des travaux, les mesures de test sur les ontologies ne sont pas mentionnées.

5. Partitionnement à base de clustering

Dans notre système, nous avons utilisé le partitionnement qui a été fait par [Benkaddour, 2018] comme une première partie.

Nous présentons le schéma global du partitionnement à base de clustering, Le partitionnement est composé des phases suivantes :

Phase 1 : Extraction des composants de l'ontologie OWL

Dans la première phase, en entrée il y a une ontologie OWL, après il y aura un parcours de tout la hiérarchie de l'ontologie, pour extraire tous les composants qui se trouvent dans cette dernière (concepts, instances, attributs et propriétés).

Phase 2 : Calcul de distances entre concept

Après l'extraction des composants de l'ontologie, vient le calcul de distances qui s'effectue entre les concepts et les instances de l'ontologie afin de mesurer leur degré de correspondance.

Phase 3 : Processus de partitionnement

Après le calcul de toutes les mesures de similarités, comme troisième étape il y a processus de partitionnement, il s'agit d'implémenter un algorithme de clustering amélioré, qui se compose de trois algorithmes combiné : k-means, k-medoide et le mal classé. A la fin, le résultat sera un de partitions.

6. Conclusion

Dans ce chapitre, nous avons présenté un aperçu des différents axes de recherche sur lesquels s'appuie l'intégration des ontologies. Nous avons présenté les méthodes d'intégration d'ontologies, particulièrement le processus d'alignement qui constitue une étape préliminaire pour les méthodes de fusion. Notre objectif est de faire une fusion de deux ontologies volumineuses, afin d'avoir une seule ontologie fusionnée qui englobe les connaissances des ontologies fusionnées.

Dans le prochain chapitre, nous présentons l'approche que nous avons proposée

Chapitre III : Approche Proposée

1. Introduction

Dans le chapitre précédent nous avons fait des comparaisons sur les différentes approches proposées dans la littérature, et dans ce chapitre nous proposerons une approche qui traite la fusion des ontologies volumineuses en combinant les mesures de similarité et en prenant en considération les points forts et les points faibles de chaque méthode déjà analysée pour obtenir un bon résultat .

Nous commencerons d'abord par citer les limites des approches existantes, ensuite nous introduirons le processus de partitionnement à base de clustering, ensuite nous présentons notre approche avec ses différentes étapes de mapping et fusion.

2. Limites des approches existantes.

La majorité des travaux existants que nous avons déjà analysé dans le chapitre précédent utilisent peu de mesure de similarité ce qui diminue la précision de la similarité sémantique, nous avons distingué aussi que la majorité des travaux sont basés sur le mapping et non pas la fusion tandis qu'il y'a d'autre approches qui ne traite pas le passage à l'échelle.

Nous avons mis au point un système d'intégration qui sert à réaliser le mapping et la fusion des ontologies volumineuses, il se résume comme suit :

- ❖ Mettre un système d'intégration sert à réaliser le mapping et la fusion des ontologies volumineuses.
- ❖ Le calcul d'une similarité sémantique entre les concepts et les instances de l'ontologie, il s'agit de combiner plusieurs mesures de calcul de similarité existante (terminologique, structurelle et intentionnelle).
- ❖ L'utilisation des ressources externe pour avoir un mapping sémantique avec l'utilisation de Wordnet.
- ❖ L'automatisation du processus d'alignement et de la fusion
- ❖ Utilisation du standard OWL comme un langage de représentation d'ontologie
- ❖ Réaliser des tests du système sur des ontologies volumineuses

3. L'approche proposée pour l'intégration

Notre approche consiste à faire l'intégration entre deux ontologies volumineuses, d'abord nous abordons le mapping ensuite nous réalisons la fusion en se basant sur les résultats du mapping.

4.1.Mapping

Le mapping entre deux ontologies « o_1 ; o_2 » signifie que pour chaque entité dans une ontologie « o_1 », il faut trouver l'entité correspondante dans l'ontologie « o_2 » avec un sens équivalent ou le sens le plus proche.

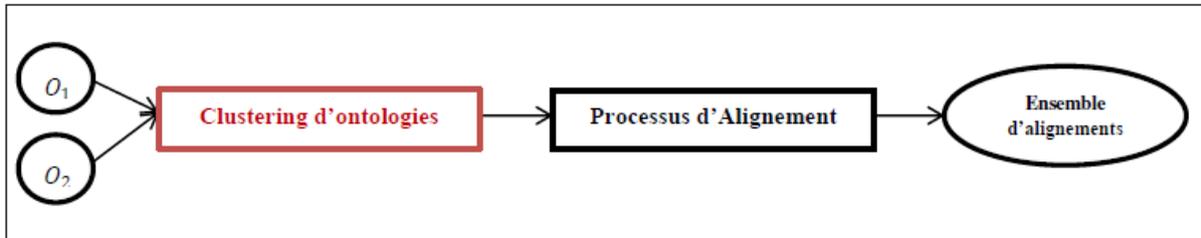


Figure 10: Processus d'alignement basé sur le clustering d'ontologies

Dans notre approche nous réalisons un mapping entre bloc dans le but de diminuer le nombre de calculs à cause de la taille de l'ontologie après nous feront un mapping entre concept pour retirer l'ensemble d'alignement

4.1.1. Mapping entre blocs

En partant d'un partitionnement d'ontologies et la désignation de centres de chaque cluster de l'ontologie partitionné ces derniers se sont les entrées du mapping nous calculons les mesures de similarités de chaque centre de la première ontologie avec tous les autres centres de la deuxième ontologie pour afin de trouver l'entité correspondante avec un sens équivalent ou le sens le plus proche pour produire comme sortie un ensemble de correspondances entre blocs

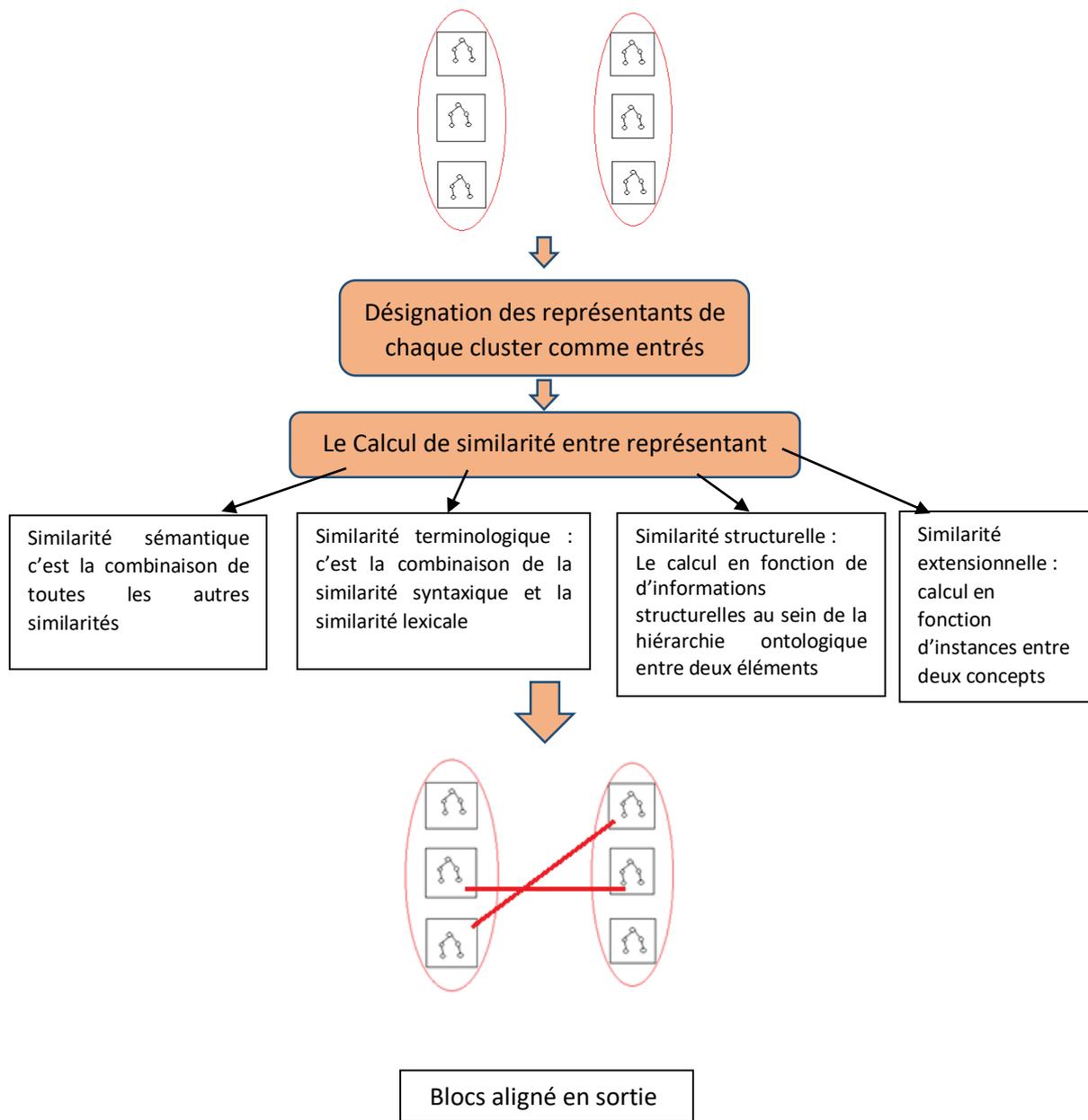


Figure 11: processus de mapping entre blocs

4.1.2. Mapping entre concepts

Après avoir trouver les correspondances avec les centres des clusters (blocs) de chaque ontologie, nous lançons le mapping entre concepts, cela veut dire de calculer la similarité entre chaque concept qui se trouve dans des blocs similaire c.à.d. faire un calcul de similarité sémantique entre les concepts du bloc A et les concepts du bloc B qui sont similaires, chaque

concept du bloc A va avoir toutes les valeurs de similarité avec les autres concepts du bloc B pour ensuite déterminer le concept le plus proche.

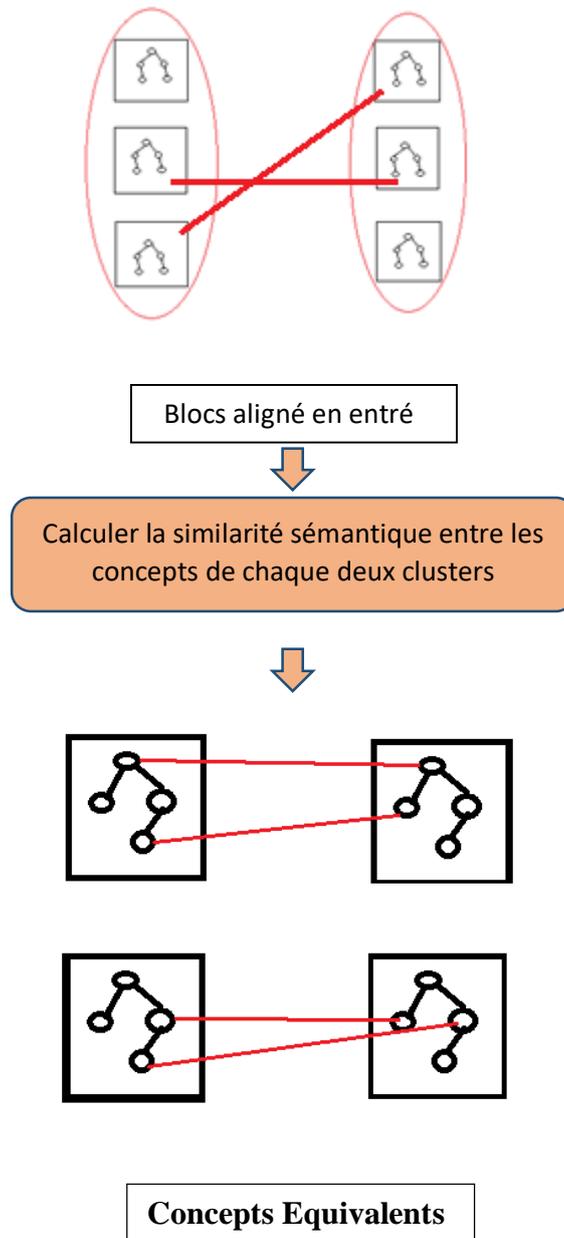


Figure 12: processus de mapping entre concepts

4.2. Les mesures de similarité utilisées

Il existe plusieurs mesures de similarité dans la littérature, on cite ceux qu'on a utilisés dans notre travail :

4.2.1. Similarité terminologique

Afin de mesurer la similarité terminologique entre les concepts, nous allons utiliser les deux mesures suivantes :

4.2.1.1. Similarité lexicale

Les méthodes basées sur un langage se fondent sur des techniques de traitement du langage naturel afin de trouver des associations entre les entités ou les classes. Ces méthodes exigent l'utilisation de ressources externes. Plusieurs types de ressources peuvent être employés, notre choix s'est porté sur WordNet.

WordNet est une ressource lexicale de langue anglaise, disponible sur internet, 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.

WordNet contient des liens entre les synsets qui représentent plusieurs relations : is-a (est-un(e)), part-of (fait-partie-de), synonymie, antonymie, hyponymie, homonymie etc.

Pour le calcul de la similarité linguistique, la fonction $\text{Syn}(c)$ calcule l'ensemble des Synsets de WordNet du concept c retenus après l'enrichissement de l'ontologie soit

$S = \text{Syn}(c1) \cap \text{Syn}(c2)$ l'ensemble des sens communs entre $c1$ et $c2$ à comparer, la cardinalité de S est :

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

Soit $\min(|\text{Syn}(c1)|, |\text{Syn}(c2)|)$ le minimum entre les cardinalité des deux ensembles $\text{Syn}(c1)$ et $\text{Syn}(c2)$ alors la mesure de similarité lexicale entre deux concepts $c1$ et $c2$ est défini comme suit :

$$\text{Simlex}(c1, c2) = \frac{\lambda(s)}{\min(\text{Syn}(c1), \text{Syn}(c2))}$$

4.2.1.2. Similarité syntaxique :

Dans notre système nous allons utiliser la distance de JARO, cette dernière mesure la similarité entre deux chaînes de caractères. Plus la distance de Jaro entre deux chaînes est élevée, plus elles sont similaires.

Cette mesure est particulièrement adaptée au traitement de chaînes courtes comme des noms ou des mots de passe. Le résultat est normalisé de façon à avoir une mesure entre 0 et 1, le zéro représente l'absence de similarité.

La distance de Jaro entre les deux chaînes s_1 et s_2 est définie par :

$$SymSin = \frac{1}{3} \left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m} \right)$$

Où :

- m est le nombre de caractères correspondants.
- t est le nombre de transpositions.

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 :

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

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.

DSJaro : $S \times S \in [0, 1]$ telle que :

$$SymSin(Marie, Maires) = \frac{1}{3} \left(\frac{5}{|5|} + \frac{5}{|6|} + \frac{5-1}{5} \right) = 0.87$$

Nous avons ensuite, combiné ces deux mesures, on ajoutant un poids à la similarité Lexicale le résultat est la formule suivante :

$$Sim_{ter}(c_1, c_2) = \frac{(SimLex(c_1, c_2) * 3) + SimSyn(c_1, c_2)}{4}$$

4.2.2. Similarité structurelle

Cette dernière sera calculée par la mesure de Wu et Palmer :

$$Simwp(c1, c2) = \frac{2 \times Profondeur(c)}{Profondeur(c1) + Profondeur(c2)}$$

Exemple

Soit l'ontologie de la Figure 13, on dénote par C1, C2 et C3 les concepts «Person», «PostDoc» et «AdministrativeStaff». En appliquant la mesure de Wu et Palmer, la valeur de similarité est calculée comme suit :

$$Simwp(C1, C2) = 2 \times 1 / (1 + 4) = 0.4$$

$$Simwp(C2, C3) = 2 \times 2 / (4 + 3) = 4 / 7 = 0.57$$

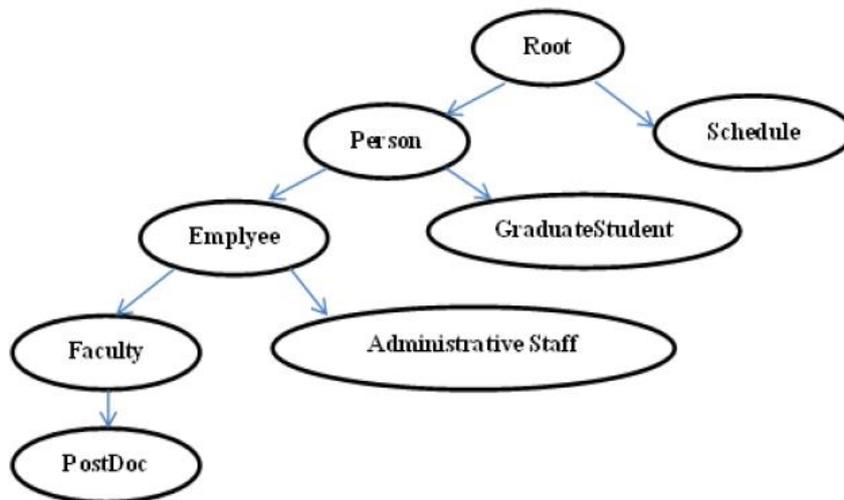


Figure 13 : Extrait de l'ontologie UnivBench.

4.2.3. Similarité extensionnelle

Pour calculer la mesure de similarité extensionnelle, nous allons utiliser la mesure terminologique syntaxique de Jaro, que nous avons appliqué précédemment sur les concepts, adaptée sur les termes des instances.

Soit C1, C2 deux concepts. Pour calculer la mesure de similarité entre les instances appartenant aux C1 et C2, on suit les étapes suivantes :

Etape 1 :

- Définir un seuil **S1**
- Calculer la mesure de similarité entre toutes les instances appartenant à C1 et C2.
- **Si** la similarité **SimJaro** entre Instances Inst1 et Inst2 (appartenant à C1, C2 successivement)

est supérieure ou égale au seuil ($SimJaro \geq S1$)

Alors Inst1 et Inst2 sont similaires, Stocker **SimJaro**

Sinon Sim =0

Etape 2 :

- Définir un autre seuil **S2**, tel que **S2** est le nombre d'instances similaires.
- Calculer la formule suivante : $F = \frac{\sum \text{instances similaires (C1,C2)}}{\sum \text{nombre d'instances(C1,C2)}}$
- **Si F** \geq **S2**

Alors

$$SimIns = \frac{\sum SimJaro(C1, C2)}{\sum \text{nombre d'instances}(c1, c2)} \text{ ou } Sim_{Jaro} \geq S1$$

C.à.d. (**SimIns** = La moyenne des mesures qui dépassent S1)

- **Sinon SimIns = 0**

4.2.4. Similarité sémantique

La similarité sémantique sera calculée par la combinaison hybride des similarités terminologique, structurelle et extensionnelle, en ajoutant une condition sur la similarité extensionnelle car si la classe ne contient pas des instances nous éliminons la similarité extensionnelle et nous avons aussi ajouté un poids à la similarité structurelle grâce à ça grande importance, ce poids a été désigné grâce à des tests afin d'obtenir des meilleurs résultat de mapping, La similarité sémantique définis selon la formule suivante :

Si un concept n'a pas d'instances :

$$SimSem(c1, c2) = \frac{SimTer(c1, c2) + (SimStr[c1, c2] * 4)}{5}$$

Sinon :

$$SimSem(c1, c2) = \frac{SimTer(c1, c2) + (SimStr[c1, c2] * 4) + SimExt(c1, c2)}{6}$$

4.3.Fusion

Dans cette section nous abordons la fusion des ontologies. Nous commençons par une simple définition ensuite nous présentons le principe de la fusion.

La fusion d'ontologies est le processus de création d'une seule ontologie rassemblant les connaissances de deux ou plusieurs ontologies existantes et différentes qui décrivent le même sujet ou appartiennent au même domaine d'application. C'est donc la création d'une nouvelle ontologie, appelée l'ontologie fusionnée capturant les connaissances des ontologies d'origines. Le défi est alors d'assurer que toutes les correspondances et les différences entre les ontologies soient correctement prises en compte dans l'ontologie résultante.

Les approches les plus courantes utilisent l'union ou l'intersection. Dans l'approche par union, l'ontologie résultante contient l'union des entités provenant des ontologies originales et suppose résolues les différences de représentation d'un même concept. Dans l'approche de type intersection, l'ontologie résultante ne contient que les parties communes des ontologies originales.

4.3.1. Principe de la fusion

L'utilisation conjointe de deux ontologies peut nécessiter soit un simple alignement dans le cas où il n'y a aucune ressemblance entre les deux ontologies, et que le résultat de mapping est vide, soit une véritable fusion. L'alignement est suffisant dans le cas de l'utilisation d'ontologies portant sur des domaines de connaissance complémentaire, ou sur des domaines de niveaux sémantiques différents. Par exemple au lieu d'utiliser deux ontologies du domaine « Médecine » on peut les fusionner, et obtenir une seule ontologie englobant les connaissances des deux ontologies.

- **Préparation de la fusion**

Pour préparer la fusion nous procédons à extraire chaque deux concepts déterminés similaires par le mapping, et nous utilisons le Wordnet qui nous offre la possibilité de récupérer le plus bas hyperonyme commun dans la structure du Wordnet, cet hyperonyme sera déterminé comme le nouveau concept qui est la fusion des deux concepts initiaux. (*Figure 5*)

Exemple : [Author,Co_Author] → [Person]

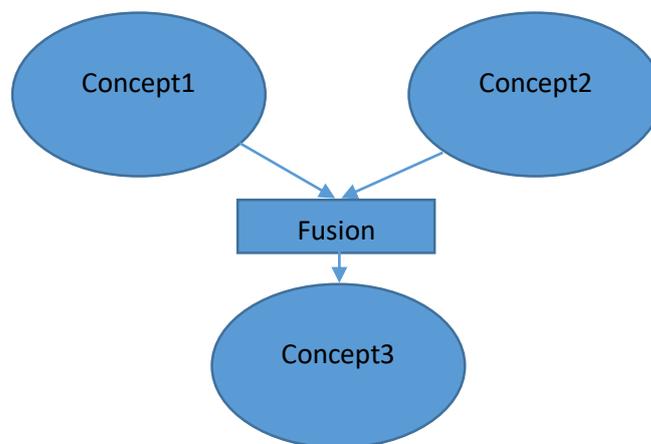


Figure 14: Exemple de fusion de deux concepts

Ensuite nous avons procéder à la fusion par union comme suit :

Tout d'abord nous commençons par copier les deux ontologies dans notre nouvelle ontologie, par le chargement des concepts racines qui ont comme superclasse la classe « Thing » des deux ontologies initiales en premier lieu Ensuite nous chargeons les autres concepts dans notre ontologie finale de façon automatique afin d'obtenir une union entre les deux ontologies initiales.

La *figure 15* représente un exemple de chargement des racines des deux ontologies initiales dans la nouvelle ontologie.

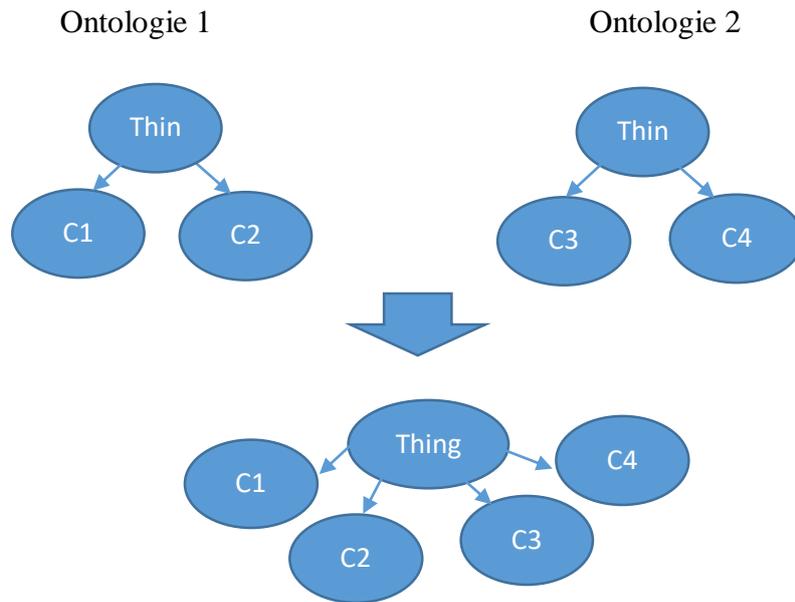


Figure 15: exemple de chargement des racines dans la nouvelle ontologie

Ensuite, nous plaçons les concepts fusionnés pendant la préparation dans notre nouvelle ontologie.

La *figure 7* représente un exemple de fusion de deux ontologies où on peut voir que les deux concepts C4 et C5 ont été fusionnés dans le concept C4,5.

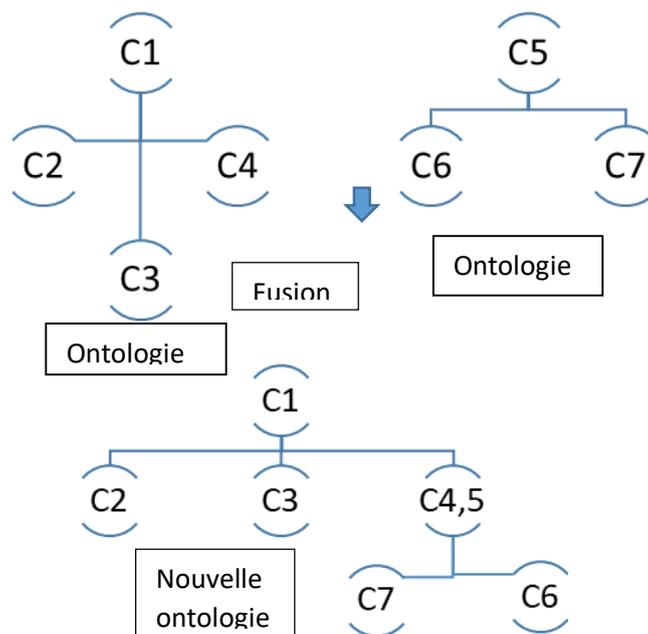


Figure 16: Exemple de fusion de deux ontologies

Finally, we proceed to the extraction of DataProperty and ObjectProperty for each class and insert them into the new ontology, by joining the properties of similar concepts into the new fused concepts.

4. Conclusion

In this chapter we have talked about different similarity measurement techniques and the combination we used in our work. We have presented the different processes present in our system, first we presented the partitioning, then we talked about the mapping, and finally we described the fusion process which will be presented in the next chapter, with the conception and the global architecture of our system.

Chapitre IV : ***Implémentation du système***

1. Introduction

Nous avons présenté dans le chapitre précédent l'approche proposée dans notre travail, dans ce chapitre on va expliquer l'implémentation de cette approche. Tout d'abord on va Présenter l'environnement de développement ainsi que les différents outils utilisés dans ce travail, ensuite on fera une description de notre système à travers quelques captures de l'interface de notre programme.

2. Environnement de développement

Tout d'abord nous allons spécifier les différents outils utilisés dans notre travail

2.1. Matériel utilisé

Tout le travail et les tests présents dans ce mémoire ont été réalisés sur une machine avec les spécifications suivantes :

CPU : AMD Ryzen™ 5 2400G avec processeur graphique Radeon™ RX Vega 11

Nombre de cœurs : 4

Nombre de threads : 8

Fréquence de base : 3.6Ghz

Fréquence Boost maximale : 3.9Ghz

RAM : 16 Gb /DDR4

Système d'exploitation : Windows 10 - Édition 64 bits

2.2. Langage utilisé

Pour l'implémentation nous avons utilisé le langage de programmation Python.

- **Définition**³:

Python est un langage de programmation, dont la première version est sortie en 1991. Créé par **Guido van Rossum**, il a voyagé du Macintosh de son créateur, qui travaillait à cette époque au *Centrum voor Wiskunde en Informatica* aux Pays-Bas, jusqu'à se voir associer une organisation à but non lucratif particulièrement dévouée, la **Python Software Foundation**, créée en 2001. Ce langage a été baptisé ainsi en hommage à la troupe de comiques les « Monty Python ».

Python est un langage de programmation puissant et facile à apprendre. Il dispose de structures de données de haut niveau efficaces et d'une approche simple mais efficace de la programmation orientée objet. La syntaxe élégante et le typage dynamique de Python, associés à sa nature interprétée, en font un langage idéal pour la création de scripts et le développement rapide d'applications dans de nombreux domaines sur la plupart des plates-formes.

L'interpréteur Python et la bibliothèque standard étendue sont librement disponibles sous forme de source ou de binaire pour toutes les principales plates-formes à partir du site Web Python, et peuvent être distribués librement. Le même site contient également des distributions et des pointeurs vers de nombreux modules, programmes et outils Python gratuits, ainsi que de la documentation supplémentaire.

L'interpréteur Python est facilement étendu avec de nouvelles fonctions et de nouveaux types de données implémentés en C ou C++ (ou d'autres langages pouvant être appelés à partir de C). Python convient également comme langage d'extension pour les applications personnalisables.

2.3. Paquets utilisés dans Python

Dans l'implémentation de notre système, on a utilisé les paquets suivants dans python :

2.3.1. Numpy

Natural Language Toolkit (NLTK) est une bibliothèque logicielle en Python permettant un traitement automatique des langues, développée par Steven Bird et Edward Loper du département d'informatique de l'Université de Pennsylvanie. En plus de la bibliothèque,

³ <https://www.python.org>

NLTK fournit des démonstrations graphiques, des données-échantillon, des tutoriels, ainsi que la documentation de l'interface de programmation (API).

2.3.2. Owlready2⁴

Owlready est un module de programmation orientée ontologie en Python, incluant un quadre RDF optimisé.

Owlready peut :

- Importer les ontologies OWL 2.0 au format NTriples, RDF/XML ou OWL/XML.
- Exporter les ontologies OWL 2.0 vers NTriples ou RDF/XML.
- Manipuler de manière transparente les classes, les instances et les propriétés d'ontologie, comme s'il s'agissait d'objets Python normaux.
- Ajouter les méthodes Python aux classes d'ontologie.

2.3.3. Wordnet

WordNet est une base de données lexicale pour la langue anglaise, créée par Princeton et faisant partie du corpus NLTK.

WordNet peut être utilisé avec le module NLTK pour trouver la signification des mots, des synonymes, des antonymes, etc.

3. Le logiciel Protégé pour la visualisation des ontologies

Protégé est un système auteur pour la création d'ontologies. Il a été créé à l'université Stanford et est très populaire dans le domaine du Web sémantique et au niveau de la recherche en informatique.

Protégé est développé en Java. Il est gratuit et son code source est publié sous une licence libre (la Mozilla Public Licence).

⁴ <https://pythonhosted.org/Owlready2/>

Protégé peut lire et sauvegarder des ontologies dans la plupart des formats d'ontologies : RDF, RDFS, OWL, etc.

4. Présentation de l'application

Afin de faciliter l'interaction et l'utilisation de notre système, nous avons implémenté une interface graphique permettant de choisir les ontologies à fusionner, et lancer le programme.

Une barre de progression indique l'avancement du traitement et un texte qui s'affiche pour indiquer l'étape du travail en cours.

Les résultats seront enregistrés dans le bureau par défaut.

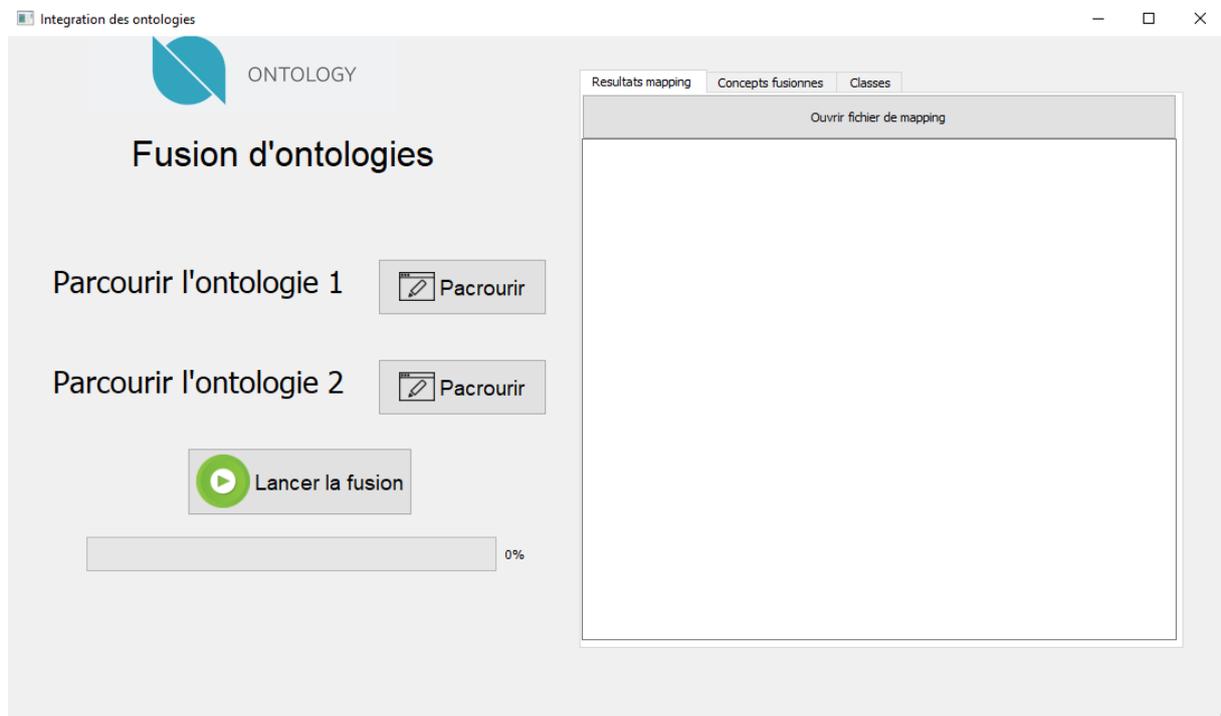


Figure 17: première fenêtre de l'application

La figure 17 est la première fenêtre de notre programme, le premier bouton ouvre une fenêtre de dialogue pour choisir l'emplacement de la première ontologie, et le deuxième bouton permet de parcourir la deuxième ontologie.

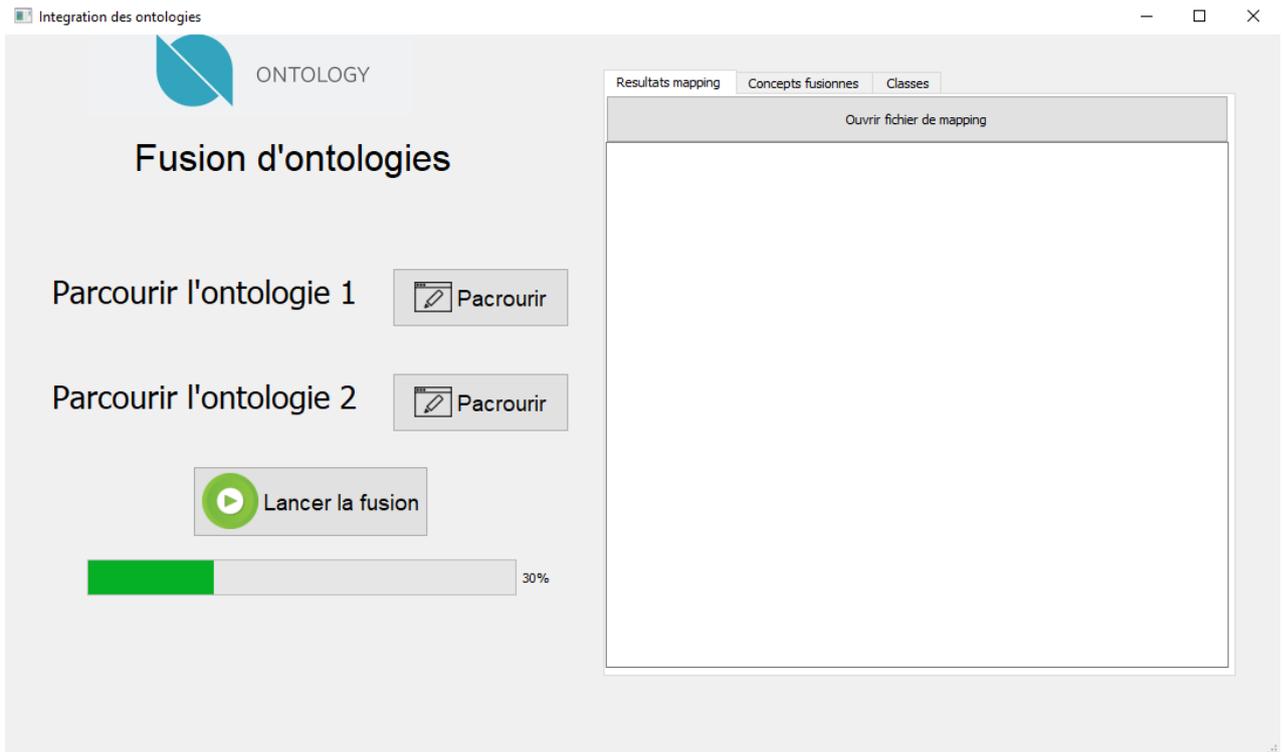


Figure 18: Interface en cours de traitement

La *figure 18* représente l'état de l'interface après l'appuyé sur le bouton « Lancer la fusion »

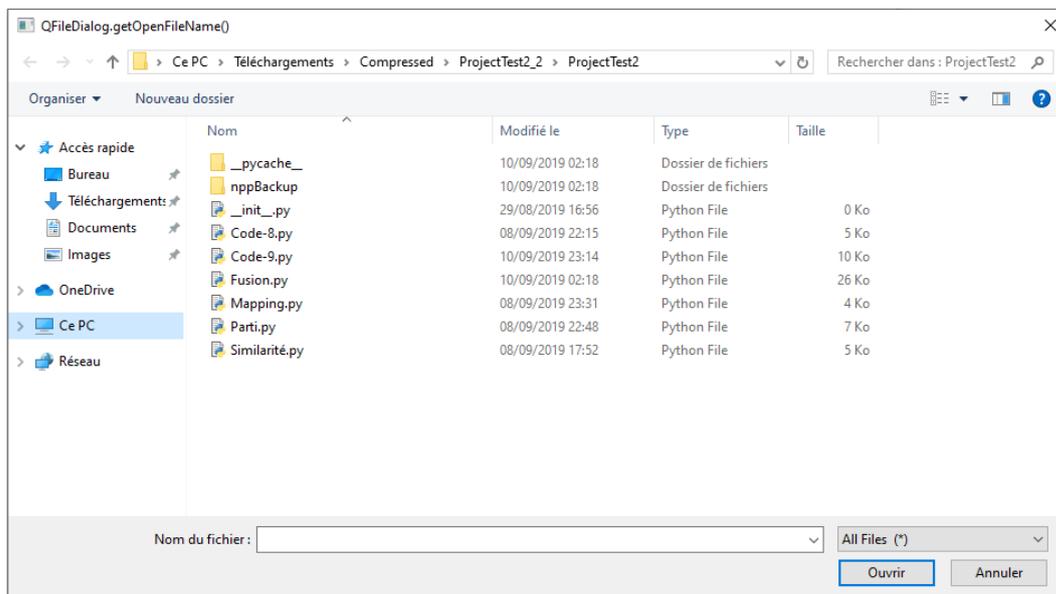


Figure 19: fenêtre de dialogue pour choisir l'emplacement de l'ontologie

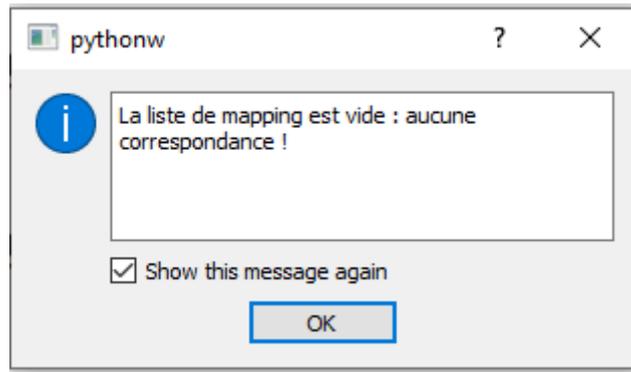


Figure 20: Fenêtre d'erreur 1

La *figure 20* représente une fenêtre d'erreur lorsque notre programme ne trouve aucune correspondance pour le mapping entre les deux ontologies.

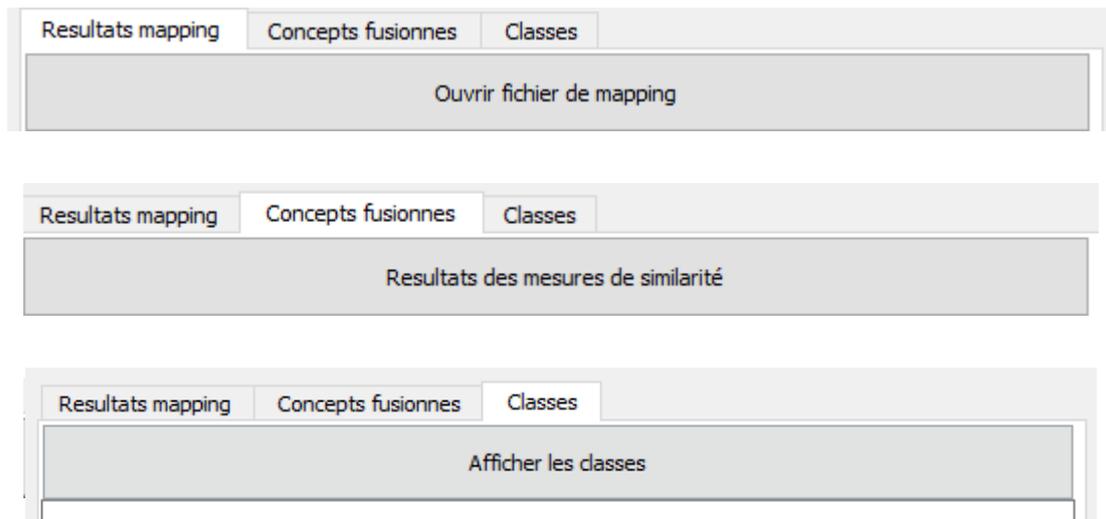


Figure 21: trois boutons d'affichage de résultat de Mapping et de la fusion

La *figure 21* représente trois onglets pour afficher les résultats de mapping et de la fusion.

Nous expliquons l'utilité de chaque bouton dans ce qui suit :

Le bouton numéro 1 affiche les concepts correspondants et leur similarité

Integration des ontologies

ONTOLGY

Fusion d'ontologies

Parcourir l'ontologie 1 Pacourir

Parcourir l'ontologie 2 Pacourir

Lancer la fusion

100%

Concepts fusionnés	Classes
Conference---->Social_Event	la similarité = 0.5547619047619048
Conference---->Session	la similarité = 0.5580498866213152
Conference---->University	la similarité = 0.6069190069190069
Conference---->Conference	la similarité = 0.7936507936507937
Conference---->Organisation	la similarité = 0.5292876124808897
Topic---->Session	la similarité = 0.5315803244374673
Topic---->Review	la similarité = 0.5432037815126051
Topic---->Proceedings	la similarité = 0.6094128094128094
Topic---->Tutorial	la similarité = 0.6297619047619047
Topic---->Conference	la similarité = 0.5768620268620268
Topic---->Organisation	la similarité = 0.5586324389895819
Topic---->Document	la similarité = 0.6028887778887779
Country---->Location	la similarité = 0.5421555428698286
Sponsorship---->Workshop	la similarité = 0.6724598930481283
Document---->Document	la similarité = 0.7068995284594024
Review---->Social_Event	la similarité = 0.5870634920634921
Review---->Review	la similarité = 0.8202262715416184
Person---->Presenter	la similarité = 0.5372568970608187
Person---->Paper	la similarité = 0.6221811957106074
Person---->Flyer	la similarité = 0.6107839509800294
Person---->Person	la similarité = 0.753006253006253
Person---->Web_Site	la similarité = 0.5287503890445067
Reviewer---->Presenter	la similarité = 0.573733576351554
Reviewer---->Web_Site	la similarité = 0.6318181818181818
Attendee---->Presenter	la similarité = 0.5373015873015873
Attendee---->Web_Site	la similarité = 0.5952380952380952
Attendee---->Student	la similarité = 0.5423469387755102
Paper---->Paper	la similarité = 0.8846747884731342
Paper---->Web_Site	la similarité = 0.5339411983116633
Author---->Flyer	la similarité = 0.5515172546751493
Author---->Person	la similarité = 0.5400472802433587
Presenter---->Presenter	la similarité = 0.8810916179337231
Presenter---->Paper	la similarité = 0.5674211474963355
Presenter---->Person	la similarité = 0.5372568970608187

Figure 22: le fichier de mapping après la fin de la fusion

Integration des ontologies

ONTOLGY

Fusion d'ontologies

Parcourir l'ontologie 1 Pacourir

Parcourir l'ontologie 2 Pacourir

Lancer la fusion

100%

Concepts fusionnés	Classes
Conference---->Social_Event	le plus bas hyperonyme = abstraction
Conference---->Session	le plus bas hyperonyme = abstraction
Conference---->University	le plus bas hyperonyme = social_group
Conference---->Conference	le plus bas hyperonyme = conference
Conference---->Organisation	le plus bas hyperonyme = social_group
Topic---->Session	le plus bas hyperonyme = communication
Topic---->Review	le plus bas hyperonyme = abstraction
Topic---->Proceedings	le plus bas hyperonyme = abstraction
Topic---->Tutorial	le plus bas hyperonyme = communication
Topic---->Conference	le plus bas hyperonyme = abstraction
Topic---->Organisation	le plus bas hyperonyme = abstraction
Topic---->Document	le plus bas hyperonyme = communication
Country---->Location	le plus bas hyperonyme = entity
Sponsorship---->Workshop	le plus bas hyperonyme = entity
Document---->Document	le plus bas hyperonyme = document
Review---->Social_Event	le plus bas hyperonyme = psychological_feature
Review---->Review	le plus bas hyperonyme = reappraisal
Person---->Presenter	le plus bas hyperonyme = person
Person---->Paper	le plus bas hyperonyme = physical_entity
Person---->Flyer	le plus bas hyperonyme = entity
Person---->Person	le plus bas hyperonyme = person
Person---->Web_Site	le plus bas hyperonyme = whole
Reviewer---->Presenter	le plus bas hyperonyme = person
Reviewer---->Web_Site	le plus bas hyperonyme = whole
Attendee---->Presenter	le plus bas hyperonyme = person
Attendee---->Web_Site	le plus bas hyperonyme = whole
Attendee---->Student	le plus bas hyperonyme = person
Paper---->Web_Site	le plus bas hyperonyme = physical_entity
Author---->Flyer	le plus bas hyperonyme = entity
Author---->Person	le plus bas hyperonyme = person
Presenter---->Presenter	le plus bas hyperonyme = presenter
Presenter---->Paper	le plus bas hyperonyme = physical_entity
Presenter---->Person	le plus bas hyperonyme = person
Presenter---->Student	le plus bas hyperonyme = person

Le bouton numéro 2 affiche les concepts correspondants et leur plus bas hyperonyme qui va les remplacés

Le bouton numéro 3 affiche les classes de la nouvelle ontologie

La *figure 24* représente la fin de la fusion c.à.d. la génération de l'ontologie fusionnée

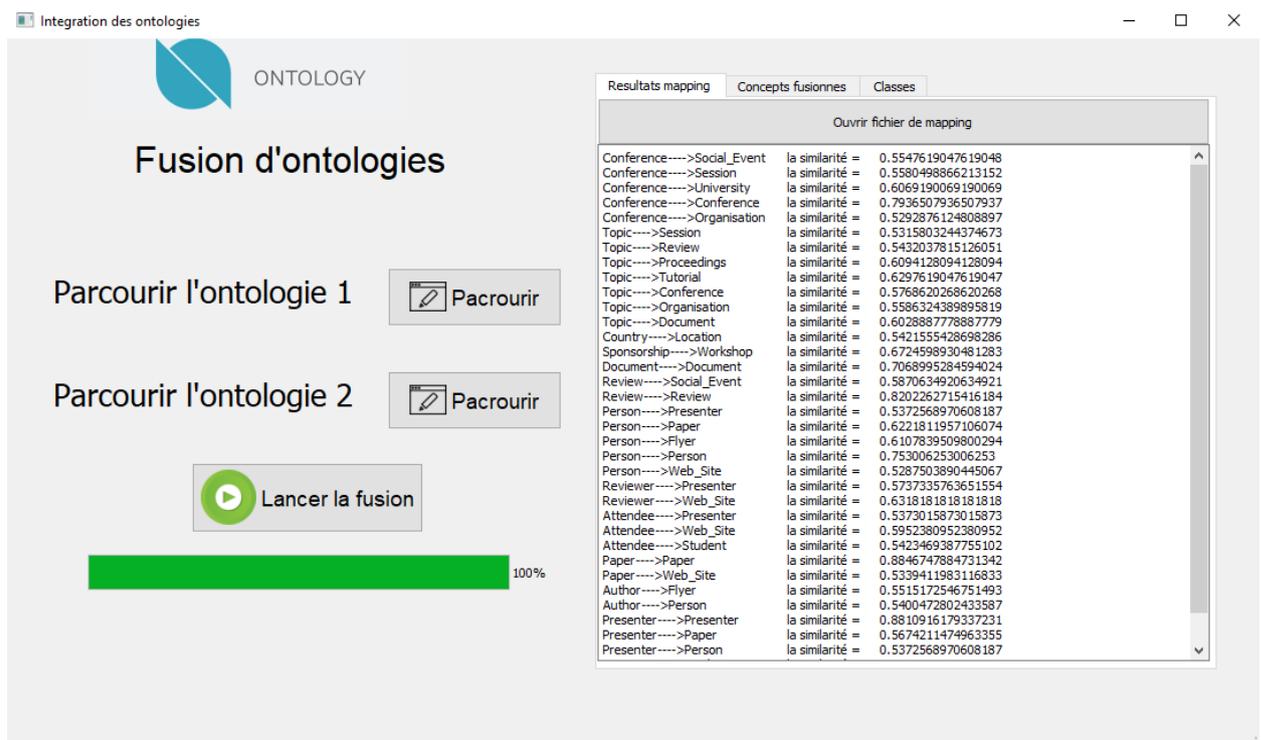


Figure 24: le traitement est terminé

5. Les résultats

Le résultat de la fusion est un fichier owl qui représente l'ontologie final mais pour arriver à ce fichier en cite comme étape

5.1.Chargement des deux ontologies

Dans cette étape le programme prépare la fusion en chargeant les deux ontologies en copiant tous les concepts dans l'ontologie final

Par exemple :

On veut fusionner l'ontologie « **cmt.owl** » l'ontologie« **conference.owl** »

Premièrement on les charge dans l'ontologie qu'on va créer (voir *figure 25*)

Dans cette étape le chargement des deux ontologies consiste à copier l'ontologie concept par concept dans l'ontologie finale.

La *figure 25* représente un exemple de chargement de l'ontologie « **cmt** » et «**Conférence**» dans l'ontologie finale.



Figure 25: chargement de l'ontologie cmt et Conférence

La *figure 26* représente une visualisation par Protégé de l'ontologie finale après le chargement des deux ontologies « **cmt** » et «**Conférence**».

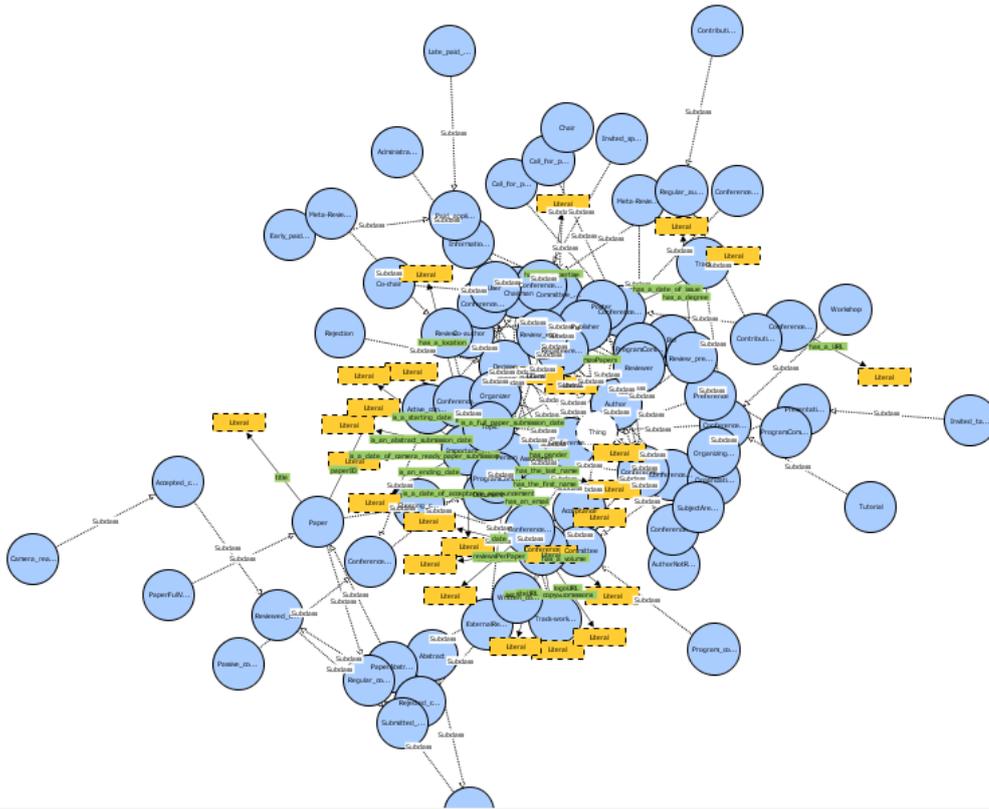


Figure 26: visualisation de l'ontologie O qui contient les deux ontologies « cmt » et «Conférence»

La figure 27 représente le chargement de la première ontologie on deux étapes :

- Etape 1 :
Sélectionner les racines de l'ontologie et les supprimer de la liste des classes.
Pour tester si la classe est une racine il faut tester si ça class mère est la classe «Thing»
If (self.listClassesDeux[i].is_a[0] == owl.Thing) :
- Etape 2 :
Chargé les subclasses (classe fille) de chaque classes créé de façon continue et automatique c.a.d chaque classe créé rentre dans la listes des classes qui seront après lié avec leur subclasses.(La même chose pour la deuxième ontologie.)

```

with ontolo:
    cls = []
    k = 0
    b = 0
    clsracine = []
    indice = len(self.listClasses)
    listOfIndex = []
    for i in range(len(self.listClasses)):
        # print(i.name)
        if (not self.listClasses[i].is_a):
            print("cc")
        elif(self.listClasses[i].is_a[0] == owl.Thing):
            clsracine.append(self.listClasses[i])
            listOfIndex.append(i)

    cpt = 0
    for i in listOfIndex:
        i = i - cpt
        del self.listClasses[i]
        cpt = cpt + 1

    listOfIndex.clear()
    for i in range(len(clsracine)) :
        cls.append(types.new_class(clsracine[i].name, (Thing,)))
        laclass = cls[k]
        k = k + 1
        with self.ontoUN:
            listtmp=list(clsracine[i].subclasses())
            pass
        for j in range(len(listtmp)):
            cls.append(types.new_class(listtmp[j].name, (laclass,)))
            k = k + 1

    for i in cls:
        for j in self.listClasses :
            if(i.name == j.name):
                with self.ontoUN:
                    listtmp=list(j.subclasses())
                    pass
                for h in listtmp :
                    cls.append(types.new_class(h.name, (i,)))
    ontolo.save("C:/Users/psyco/OneDrive/Bureau/Partitionnement dontologies/Partitionnement dontologies/Application/Ontologies/test/ontologies/o.owl")
pass

```

Sélectionné les racines

Chargement des subclasses

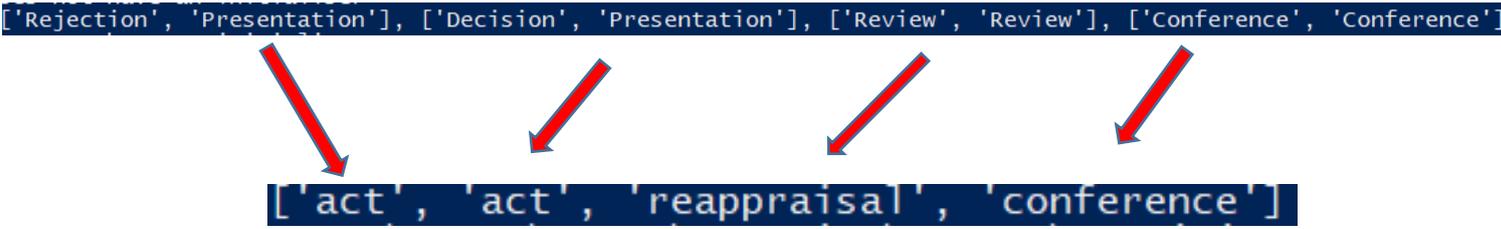
Figure 27: Code Python pour le chargement de la première ontologie

5.2. Construire l'ontologie fusionnée

Cette étape consiste à remplacer les concepts qui sont proches par un seul concept qui aura les subclasses des deux concepts et sera le fils de la superclasse de chaque concept

Par exemple :

Les ontologies que nous avons chargées précédemment vont être modifiées, en supprimant les concepts qui résultent du mapping par des concepts qui représentent chaque couple.



Le résultat :

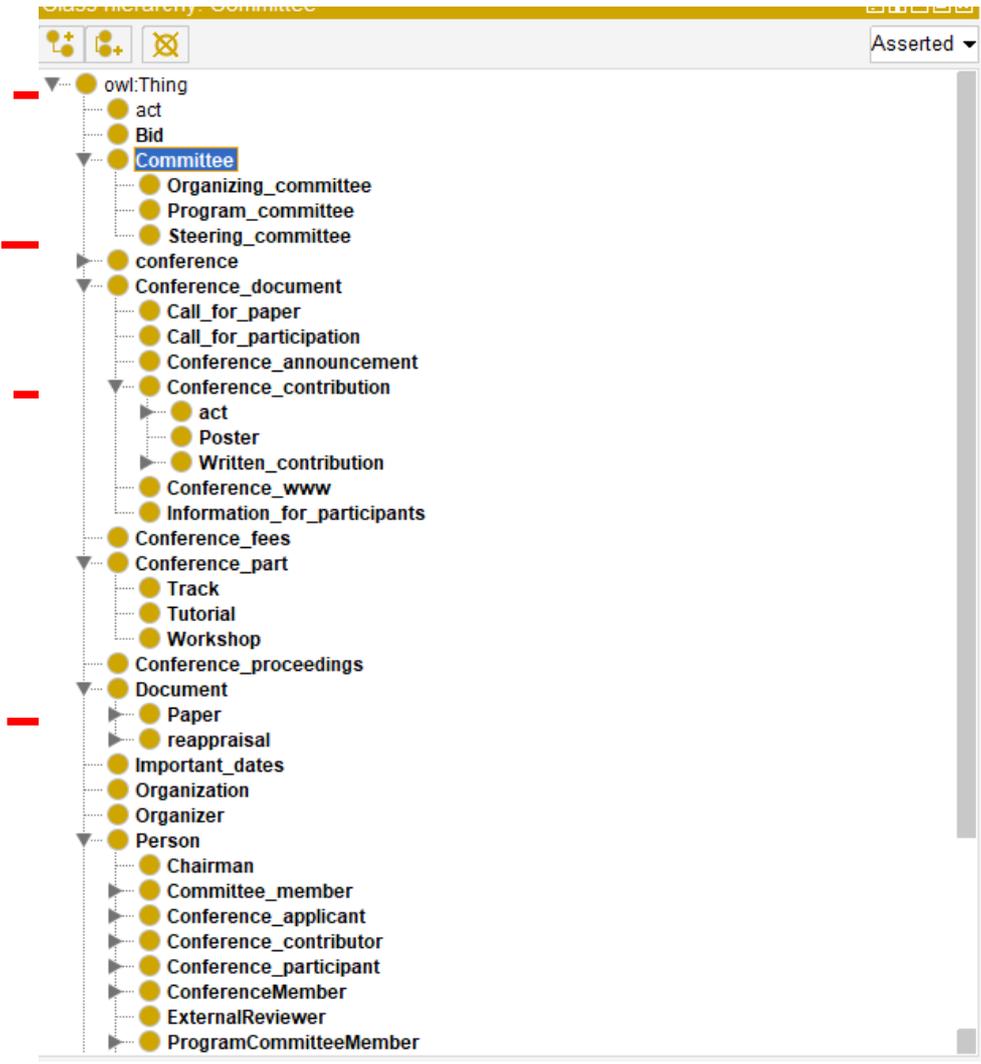


Figure 28: la hiérarchie de l'ontologie finale après la fusion

6. Conclusion

Dans ce chapitre nous avons présenté l'environnement de travail sur lequel les tests ont été effectués, ensuite nous avons défini le langage utilisé avec les différents paquets que nous avons utilisé. Finalement nous avons pris quelques captures de notre application pour illustrer son fonctionnement et son exécution. Dans le chapitre suivant, nous présentons les résultats des différents tests effectués, et la méthode de validation de notre ontologie résultante

Chapitre V :
Test et validation

1. Introduction

Dans le chapitre précédent, nous avons présenté notre environnement de travail hardware et software sur lequel tout le programme et les tests se sont effectués, ensuite nous avons présenté l'interface de notre système avec différentes captures, nous avons montré aussi les cas d'erreurs possibles qui peuvent se produire lors de l'exécution de notre système. Nous avons aussi montré l'affichage des résultats dans notre interface. Dans ce chapitre nous passons à la validation et les tests de notre système, nous commençons par montrer l'évaluation du mapping avec quelques résultats et discussions, ensuite nous abordons la validation de la nouvelle ontologie fusionnée de notre système syntaxiquement et sémantiquement.

2. Evaluation

Afin de valider notre approche, nous avons effectué des tests sur des ontologies de différentes tailles, petite et grande, selon le tableau suivant :

Taille	Ontologie (OWL)	Nombre de classes	Nombre d'instances
Petite	edas	104	
	ekaw	74	
Grandes	Human	3304	
	Mouse	2744	

Tableau 3: Les ontologies de tests

Nous rappelons que le calcul de similarité entre les différents concepts de l'ontologie se fait en combinant différentes mesures de similarité, y compris la mesure extensionnelle qui se base sur les instances des classes.

3. Evaluation du mapping

Dans cette section on procède à l'évaluation de mapping avec les deux ontologies de petites tailles « edas » et « ekaw »

3.1. Mesures d'évaluation utilisées

Pour mesurer la qualité d'un mapping, différentes mesures d'évaluation existent. Les plus connues et les plus utilisées sont :

- **Précision**

La précision est une mesure qualitative d'un système de mapping qui se calcule par rapport à un alignement de référence (ensemble des correspondances considérées correctes généralement par un humain)

Soit H l'alignement de référence. La précision d'un alignement A produit par un système S est une fonction qui renvoie une valeur appartenant à l'intervalle [0,1].

La précision est le rapport entre les correspondances correctes (True positives) sur la totalité des correspondances retournées par le système S (True positives et False positives). Sa formule est donnée Comme suit :

$$\text{Precision} = \frac{|A \cap H|}{|A|}$$

- **Rappel :**

Le Rappel est le rapport des correspondances correctes produites par le système sur le nombre total de correspondances considérées correctes (Alignements de référence H)

Il est considéré comme étant un paramètre de performance.

Une valeur élevée de Recall signifie qu'un nombre important de correspondance a été trouvé, mais aucune information sur les mappings incorrects trouvés n'est donnée.

Le calcul de Rappel est donné par la formule suivante :

$$\text{Rappel} = \frac{|H \cap A|}{|H|}$$

3.2. Résultats expérimentaux et discussion

Dans cette section nous présentons les résultats obtenus. La première série de tests a été faite sur les petites ontologies : *ekaw* et *edas*, avec différentes valeurs de K.

Dans le cadre des expérimentations menées, les tests de comparaison ont été réalisés avec l'aide d'un expert du domaine.

Nous avons comparé nos résultats avec le résultat des experts du domaine pour calculer les mesures de rappel et de précision.

La précision :

$$\text{La précision} = \frac{17}{25} = 0.68$$

Il faut prendre en compte que le mapping de test ne précise pas le nombre de clusters lors de partitionnement et ne précise pas les mesures de similarités utilisé.

Le rappel :

$$\text{le rappel} = \frac{17}{23} = 0.73$$

Le tableau ci-dessous présente les valeurs de rappel et de précision avec différent nombre de blocs « K »

	K=6	K=5	K=4	K=3
La précision	0.71	0.65	0.62	0.67
Le rappel	0.65	0.73	0.73	0.82

Tableau 4: présente les valeurs de rappel et de précision avec différent nombre de blocs

4. Evaluation de la Fusion

Pour mesurer la qualité de la fusion et sa validité, il faut la valider syntaxiquement et sémantiquement.

4.1. Mesures d'évaluation utilisées pour la syntaxe

Pour évaluer le système syntaxiquement c.à.d valider la syntaxe du fichier de sortie OWL il existe des validateurs fiables et crédibles qui proposent leurs travaux gratuitement pour valider la syntaxe d'un fichier OWL.

- **RDF Validator[1]** : est un validateur de syntaxe des fichier RDF/XML et qui est fait par le W3C
- **OWL Validator[2]** : est un validateur de syntaxe des fichier RDF/XML, OWL/XML, OWL FunctionalSyntax, Manchester OWL Syntax, OBO Syntax, et KRSS Syntax.et qui est fait par le OWL API

[1] : <https://www.w3.org/RDF/Validator/>

[2] : <https://visualdataweb.de/validator/>

4.2.Résultats expérimentaux et discussion

Nous avons validé la syntaxe avec les deux validateur OWL et RDF validator qui sont les validateurs les plus fiables et connus par les experts.

La figure ci-dessous représente la validation de la syntaxe de notre fichier OWL par RDF Validator après la fusion de « *edas* » et « *ekaw* ».



[Skip Navigation](#) [Home](#)
[Documentation](#)
[Feedback](#)

Jump To:

- [Source](#)
- [Triples](#)
- [Messages](#)
- [Graph](#)
- [Feedback](#)
- [Back to Validator Input](#)

Validation Results

Your RDF document validated successfully.

Triples of the Data Model

Number	Subject	Predicate	Object
1	http://cmt	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Ontology
2	http://cmt#Acceptance	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
3	http://cmt#Acceptance	http://www.w3.org/2000/01/rdf-schema#subClassOf	http://cmt#act
4	http://cmt#Document	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
5	http://cmt#Document	http://www.w3.org/2000/01/rdf-schema#subClassOf	http://www.w3.org/2002/07/owl#Thing
6	http://cmt#Paper	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
7	http://cmt#Paper	http://www.w3.org/2000/01/rdf-schema#subClassOf	http://cmt#Document
8	http://cmt#Paper	http://www.w3.org/2000/01/rdf-schema#subClassOf	http://cmt#Regular_contribution
9	http://cmt#Bid	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
10	http://cmt#Bid	http://www.w3.org/2000/01/rdf-schema#subClassOf	http://www.w3.org/2002/07/owl#Thing
11	http://cmt#Preference	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
12	http://cmt#Preference	http://www.w3.org/2000/01/rdf-schema#subClassOf	http://www.w3.org/2002/07/owl#Thing
13	http://cmt#SubjectArea	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
14	http://cmt#SubjectArea	http://www.w3.org/2000/01/rdf-schema#subClassOf	http://www.w3.org/2002/07/owl#Thing

Figure 29: la validation de l'ontologie edas et ekaw par RDF Validator.

OWL 2 Validation Report

Summary

The ontology and all of its imports are in the OWL 2 profile

Imports Closure

Ontology IRI	Physical URI
OntologyID(OntologyIRI(<http://cmt>))	

Figure 30: la validation de l'ontologie edas et ekaw par OWL Validator

La *figure 30* représente la validation de la syntaxe de notre fichier owl après la fusion de « *edas* » et « *ekaw* » par OWL Validator

4.3. Mesures d'évaluation utilisées pour la sémantique

La validation de la sémantique est un sujet nettement moins riche dans la littérature. Elle s'apparente à la validation de la sémantique des connaissances, la validation du modèle conceptuel en adéquation avec la réalité qu'il modélise. Cette étape oblige à une communication accrue entre acteurs de domaines d'expertises différents : les ontologies d'un côté et les spécialistes du domaine modélisé dans l'ontologie de l'autre. Alors que la validation structurelle est aisément réalisée à l'aide de méthodes automatiques, qui ne demandent quasiment aucune intervention humaine.

4.3.1. Les outils pour la validation de la sémantique

- Protégé comme un outil de développement d'ontologie, il est très populaire dans le domaine du Web sémantique, nous l'avons utilisé pour valider l'ontologie grâce à la visualisation et le graphe de la hiérarchie.
- Le reasoner HermiT est un raisonneur pour les ontologies écrites à l'aide du langage OWL (Web Ontology Language). Avec un fichier OWL, HermiT peut déterminer si

l'ontologie est cohérente, identifier les relations de subsomption entre les classes et bien plus encore.

4.3.2. Résultats expérimentaux et discussion

L'outil de développement d'ontologie protégé nous a aidé dans notre projet grâce à ces plugins de visualisation VOWL et OWLviz et témoin aussi que l'ontologie est valide pour la lire.

- **Ontologies de petites tailles**

Dans ce test nous avons utilisé les deux ontologies « edas » et « ekaw ».

Le premier test c'est la visualisation par protégé qui confirme la bonne structure de notre ontologie, cette dernière a été faite sans aucune erreur déclaré par protégé (voir *figure 32*)

La figure ci-dessous représente la visualisation de l'ontologie fusionnée par protégé.

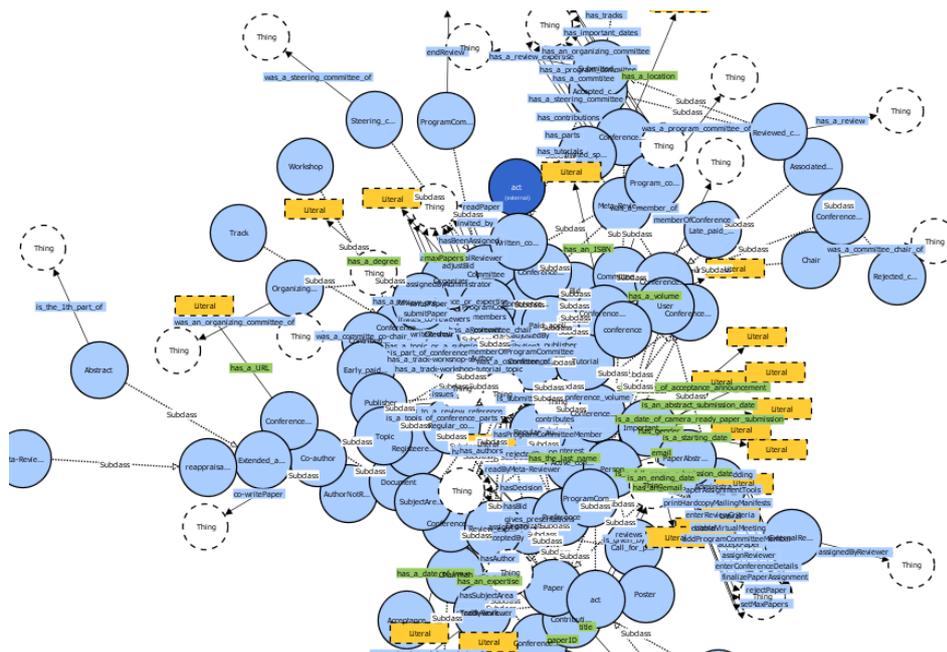


Figure 31: la visualisation de l'ontologie fusionnée par protégé.

Pour le deuxième test, nous avons utilisé le reasoner Hermit qui prend en entrée un fichier OWL et détermine si l'ontologie est cohérente et consistante.

La figure 5 représente le résultat du test de la fusion des deux ontologies « edas » et « ekaw »



Figure 32: le résultat du test de la fusion des deux ontologies « edas » et « ekaw »

- **Ontologies de Grandes tailles**

Dans ce test nous avons utilisé les deux ontologies « human » et « mouse ».

Pour ces deux ontologies volumineuses la qualité de la visualisation n'est pas très visible à cause du nombre important de concepts et relations

La figure ci-dessous présente la visualisation de l'ontologie fusionnée

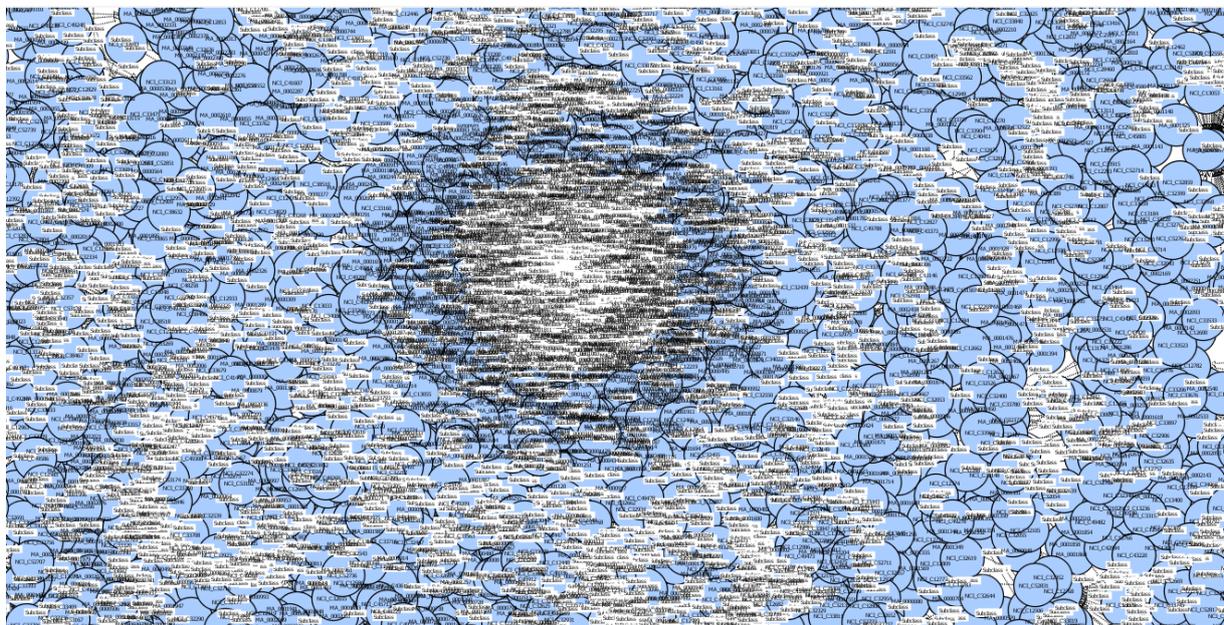


Figure 33: la visualisation de l'ontologie fusionnée des deux ontologies « human » et « mouse »

Nous avons effectué le test de cohérence et consistance sur l'ontologie résultante de la fusion des deux ontologies volumineuses « human » et « mouse »



Figure 34: resultat du test de la cohérence et la consistance

5. Conclusion

Dans ce chapitre nous avons testé l'efficacité notre système sur quelques ontologies afin de pouvoir visualiser et assurer l'exactitude des résultats obtenues par rapport aux expert de domaine, Pour la validation de notre mapping, nous avons calculé le rappel et la précision de nos résultats par la comparaison avec le résultat de l'expert du domaine. Nous avons aussi validé la syntaxe en utilisant OWL Validator. Ensuite Nousavons testé la cohérence et la consistance de notre nouvelle ontologie résultante après la fusion.

Notre système a été prouvé sémantiquement cohérent et consistant, et syntaxiquement correct.

Conclusion et Perspectives

Conclusion générale

Les travaux menés dans ce mémoire nous permis d'approfondir nos connaissances dans le domaine de l'ingénierie de connaissances et plus particulièrement l'ingénierie ontologiques. Notre objectif était de tirer profit des travaux menés dans cette voie et nous nous sommes intéressés plus particulièrement aux méthodes d'intégration d'ontologie, qui sert à réaliser l'alignement, et la fusion des ontologies en une nouvelle ontologie qui englobe les connaissances des deux ontologies.

L'alignement d'ontologies, qui a pour but la découverte des correspondances sémantiques entre des ontologies, devient une tâche cruciale pour résoudre ce problème d'hétérogénéité dans les applications du web sémantique.

Nous avons utilisé le Wordnet comme ressource externe afin d'améliorer notre alignement pour obtenir un résultat de mapping plus riche sémantiquement.

Nous avons utilisé plusieurs mesures de similarités afin d'obtenir de meilleurs résultats dans la similarité sémantique, nous avons combiné des mesures de similarité avec des poids pour chaque mesure, ces poids ont été définis après plusieurs tests.

Nous avons effectué une fusion par union, le processus est complètement automatique et ne requière aucune intervention de l'utilisateur.

La nouvelle ontologie fusionnée contient toutes les propriétés des données (Data-Properties) ainsi que les propriétés des objets (Object-Properties) des deux ontologies.

Notre nouvelle ontologie a été prouvé cohérente et consistante, ainsi qu'elle a été validé syntaxiquement.

Perspectives

Notre approche est basée sur l'alignement et le partitionnement en utilisant la combinaison de plusieurs mesures de similarité. Afin d'améliorer notre approche différentes perspectives sont envisagées :

- Utiliser le threading dans la fusion et le calcul des mesures de similarité peut améliorer le temps d'exécution.
- Création d'un outil de navigation et de visualisation d'ontologies qui peut être implémenté dans le langage de programmation python.

Références
Bibliographiques

7. Référence

[Benkaddour, 2018] : Benkaddour A, Partitionnement d'ontologies volumineuses en utilisant le datamining, (2018).

[Blazquez M et al, 1998]: BLAZQUEZ M., FERNANDEZ M., GARCIA-PINAR J. M. & GOMEZ-PEREZ A, Building Ontologies at the Knowledge Level using the Ontology Design Environment, in Proceedings of the Banff Workshop on Knowledge Acquisition for Knowledge-based System, (1998).

[Borst, 1997]:Borst, W. N., &Borst, W. N. (1997). Construction of engineering ontologies for knowledge sharing and reuse.

[Chechik M et al., 2012]: Chechik, M., Nejati, S., &Sabetzadeh, M. (2012). A relationship-based approach to model integration. *Innovations in Systems and Software Engineering*, 8(1), 3-18.

[Ding et al., 2013]: Ding, G., Sun, T., &Xu, Y. (2013, July). Multi-schema matching based on clustering techniques. In *2013 10th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)* (pp. 778-782). IEEE.

[Elbyed, 2009]:Elbyed, A. (2009). ROMIE, une approche d'alignement d'ontologies à base d'instances (Doctoral dissertation, Evry, Institut national des télécommunications).

[Euzenat et al., 2007]: Euzenat J, ShvaikoP.Ontology matching, Library of CongressControl Number: 2007926257 ACM Computing Classification(1998): H.3, H.4, I.2, F.4, ISBN: 978-3-540-49611-3, Springer Berlin :Heidelberg New York, 2007.

[Fareh et al., 2013]: Fareh, M., Boussaid, O., Chalal, R., Mezzi, M., &Nadji, K. (2013). Merging ontology by semantic enrichment and combining similarity measures. *International Journal of Metadata, Semantics and Ontologies*, 8(1), 65-74.

[Fürst, 2002] :Fürst, F. (2002). L'ingénierie ontologique. Rapport de recherche, Institut de recherche en informatique de Nates –France- (02-07).

[Garcia et al., 2012]: Garcia, A. C., Tiveron, L., Justel, C. M., &Cavalcanti, M. C. (2012). Applying Graph Partitioning Techniques to Modularize Large Ontologies. In *ONTOBRAS-MOST* (pp. 72-83).

[Giunchiglia et al., 2003]:Giunchiglia, F., Shvaiko, P., &Yatskevich, M. (2004, May). S-Match: an algorithm and an implementation of semantic matching. In *European semantic web symposium* (pp. 61-75). Springer, Berlin, Heidelberg.

[Gruber, 1993]: Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2), 199-220.

[Guan-yu et al., 2010]: Guan-yu, L., &Shu-peng, L. (2010, July). Formal concept analysis based ontology merging method. In *2010 3rd International Conference on Computer Science and Information Technology* (Vol. 8, pp. 279-282). IEEE.

[Guarino, 1997]:Guarino, N. (1997). Understanding, building and using ontologies. *International Journal of Human-Computer Studies*, 46(2-3), 293-310.

[Gu Tao, 2003]: Tao, G. (2003). Using formal concept analysis for ontology structuring and building. *ICIS, Nanyang Technological University*, 15-23.

[Hamdi et al., 2008]: Hamdi, F., Zargayouna, H., Safar, B., & Reynaud, C. (2008). Taxomap in the oaei alignment contest. In *OM, CEUR workshop proceedings* (Vol. 431).

[Hu et al., 2006]: Hu, W., Zhao, Y., & Qu, Y. (2006, September). Partition-based block matching of large class hierarchies. In *Asian Semantic Web Conference* (pp. 72-83). Springer, Berlin, Heidelberg.

[Maiz et al., 2010]: Maiz, N., Fahad, M., Boussaid, O., & Bentayeb, F. (2010). Automatic ontology merging by hierarchical clustering and inference mechanisms. In *Proceedings of I-KNOW* (pp. 1-3).

[Massmann et al., 2011]: Massmann, S., Raunich, S., Aumüller, D., Arnold, P., & Rahm, E. (2011, October). Evolution of the COMA match system. In *Proceedings of the 6th International Conference on Ontology Matching-Volume 814* (pp. 49-60). CEUR-WS. org.

[Mellal, 2007] : Mellal, N. (2007). Réalisation de l'interopérabilité sémantique des systèmes, basée sur les ontologies et les flux d'information (Doctoral dissertation, Université Savoie Mont Blanc).

[Neeches et al., 1991]: Neches, R., Fikes, R. E., Finin, T., Gruber, T., Patil, R., Senator, T., & Swartout, W. R. (1991). Enabling technology for knowledge sharing. *AI magazine*, 12(3), 36-36.

[Roche et al., 2003]: Roche, M., Heitz, T., Matte-Tailliez, O., & Kodratoff, Y. (2004, March). EXIT: Un système itératif pour l'extraction de la terminologie du domaine à partir de corpus spécialisés. In *Proceedings of JADT* (Vol. 4, pp. 946-956).

[Setti et al., 2015]: Ahmed, S. S., Malki, M., & Benslimane, S. M. (2015). Ontology partitioning: Clustering based approach. *International Journal of Information Technology and Computer Science*, 7(6), 1-11.

[Shreiber, 1995]: Schreiber, G., Wielinga, B., & Jansweijer, W. (1995, August). The KACTUS view on the 'O' word. In *IJCAI workshop on basic ontological issues in knowledge sharing* (pp. 159-168).

[Studer et al., 1998]: Studer, R., Benjamins, V. R., & Fensel, D. (1998). Knowledge engineering: principles and methods. *Data & knowledge engineering*, 25(1-2), 161-197.

[Swartout, 1997]: Schreiber, G., Wielinga, B., & Jansweijer, W. (1995, August). The KACTUS view on the 'O' word. In IJCAI workshop on basic ontological issues in knowledge sharing (pp. 159-168).

[Wu et al., 1994]: Wu, Z., & Palmer, M. (1994, June). Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics* (pp. 133-138). Association for Computational Linguistics.

[Ziani et al., 2010]: Ziani, M., Boulanger, D., & Talens, G. (2010, May). Système d'aide à l'alignement d'ontologies métier. In *INFORSID* (pp. 345-360).