Université Djillali Liabès de Sidi Bel Abbès
Faculté des Sciences Exactes
Département d'Informatique

N° d'ordre: .....

Spécialité : Informatique

Option: Systèmes d'information et de Connaissances

Thèse de Doctorat en Sciences

Contribution à l'intéropérabilité des Systèmes d'information d'entreprise à base d'alignement des ontologies

Présenté par

Mr. Zerhouni Mourad

Composition du Jury:

FARAOUN Kamel Mohamed	Professeur	UDL SBA	(Président)
BENSLIMANE Sidi Mohamed	Professeur	ESI-SBA	(Directeur de Thèse)
MALKI Mimoun	Professeur	ESI-SBA	(Examinateur)
GAFOUR Abdelkader	MCA	UDL-SBA	(Examinateur)
TOUMOUH Adil	MCA	UDL-SBA	(Examinateur)
AMAR BENSABER Djamel	MCA	ESI-SBA	(Examinateur)

Année Universitaire : 2019 – 2020

Dédicaces

Je remercie Allah de m'avoir donné la force, la patience et la volonté d'arriver au terme de ce travail.

Je dédie ce travail à ma Mère, mon Père, mon Epouse et à mes enfants chéris : Fatma-Zohra, Asma Chérifa, Mustapha, Khadidja, Aya et Djawed et à tous mes beaux parents.

A mes collègues Enseignants,

A mes étudiants.

Remerciements

Je présente mes respects et mes remerciements aux membres du jury qui m'ont fait'in l'honneur d'avoir accepté d'évaluer cette thèse .

Je remercie vivement mon directeur de thèse, Pr Benslimane Sidi Mohamed, d'avoir accepté de diriger ce travail et pour ses conseils et orientations.

Je remercie mes collègues Boukli Sofiane et Benhamouda Mohamed pour leurs encouragements et l'aide matérielle qu'ils m'ont apporté tout au long de la réalisation de cette thèse.

Mes meilleurs remerciements pour le Professeur Yousfat Abderrahmane pour ses encouragements et son aide.

Je remercie ma fille Asmaa Chérifa pour avoir accepter de saisir la présente thèse.

Mes amitiés s'adressent à mes collègues Enseignants du département d'Informatique.

Table des matières

Table des figures

Liste des tableaux

INTRODUCTION GENERALE 0

Sommaire

0.1 Introduction

De nos jours, les ontologies sont devenues l'une des plus importantes orientations de recherche notamment avec l'avènement du Web Sémantique. Une ontologie est définie comme la conceptualisation des objets reconnus comme existant dans un domaine, de leurs propriétés et des relations les reliant [Webster,2004]. Elles jouent un rôle primordial pour l'annotation de pages ou de services web puisqu'elles modélisent les concepts, attributs et relations utilisés pour annoter le contenu des ressources.

Dans de nombreux contextes applicatifs, plusieurs ontologies couvrant un même domaine ou des domaines connexes sont développées indépendamment les unes des autres par des communautés différentes, d'où la problématique de pouvoir échanger, intégrer et transformer les données. A ce stade se pose le problème d'intéropérabilité permettant à des systèmes hétérogènes de pouvoir communiquer et coopérer. Pour ce faire, les liens sémantiques entre les entités appartenant à deux ontologies différentes doivent être établis. Le passage à l'échelle du web est un véritable défi qui demande des efforts aux chercheurs pour optimiser la gestion du contenu qui peut faire l'objet d'un enrichissement et évolution permanente. A cette fin, il est nécessaire d'améliorer la qualité de l'organisation, la structuration, la recherche, l'identification, l'accès, l'utilisation, la réutilisation des ressources, l'intégration, et les traitements automatisés de ce contenu. Toutes les techniques d'alignement sont appelées à passer à l'échelle pour traiter des ontologies volumineuses. Mais cela, n'est pas toujours évident parce que la création de multiples ontologies, parfois pour un même domaine conduit à une hétérogénéité entre les connaissances exprimées au sein de chacune d'entre elles qui doit être résolue : c'est la problématique d'interopérabilité.

Bien que phénoménalement réussi en termes de taille et de nombre d'utilisateurs, le World Wide Web d'aujourd'hui est fondamentalement un artefact relativement simple. Le contenu web se compose principalement de l'hypertexte distribué qui est accessible via une recherche basée sur une combinaison de mots clés et la navigation sur les liens. Cette simplicité a été l'une de ses grandes forces et elle a été un facteur important dans sa popularité et sa croissance : les utilisateurs naifs sont en mesure de l'utiliser, et peuvent eux-mêmes créer leurs propres contenus.

L'explosion à la fois de la gamme et de la quantité du contenu web a toutefois mis en évidence de graves lacunes dans le paradigme de l'hypertexte. En premier lieu, le contenu requis devient de plus en plus difficile à localiser en utilisant la recherche et le parcours. Par exemple, trouver des informations sur des personnes avec des noms très communs(ou avec des homonymes célèbres) peut être une expérience frustrante.

Répondre à des requêtes plus complexes telles que, la recherche d'informations générales,l'intégration et le partage d'informations peut être difficile ou même impossible. La question qui se pose est de savoir comment trouver et intégrer l'information sur la demande et sans intervention humaine. Tel est l'objectif du web sémantique [Berners-Lee and all,2001], le but ultime étant de permettre aux données d'être partagées efficacement par des communautés plus larges, et d'être traitées automatiquement par des outils ainsi que manuellement. L'exemple classique

d'une application web sémantique est un agent de voyage automatisé qui compte tenu de diverses contraintes et préférences, offrirait à l'utilisateur des voyages ou des vacances appropriés. Une caractéristique clé d'un tel agent logiciel est qu'il n'exploiterait pas simplement un ensemble prédéterminé de sources d'information, mais il recherche sur le web pour obtenir des informations pertinentes de la même manière qu'un utilisateur humain pourrait la faire lors de la planification des vacances.

Une difficulté majeure dans la réalisation de cet objectif est que le contenu du web est principalement destiné à la présentation et à la consommation par les utilisateurs humains : les balisages HTML sont principalement concernés par la mise en page, la taille, la couleur et d'autres questions de présentation. De plus, les pages web utilisent de plus en plus d'images, y compris souvent des liens actifs, pour présenter des informations, et même lorsque le contenu est annoté, les annotations prennent généralement la forme de chaînes de langage naturel. Les utilisateurs humains sont (généralement) capables d'interpréter la signification de ses caractéristiques, et ainsi comprendre l'information présentée, mais cela peut ne pas être facile pour un agent logiciel. Une idée clé derrière le web sémantique est de résoudre ce problème en donnant la sémantique accessible par la machine aux annotations. Ceci doit être réalisé en utilisant des ontologies pour donner des significations formellement définies aux termes utilisés dans les annotations, c'est à dire, transformer ces termes en annotations sémantiques.

Les ontologies ont été reconnues comme une composante essentielle pour le partage des connaissances et la réalisation de la vision du web sémantique, où elles vont décrire la structure et la sémantique des données. L'idée est que les ontologies permettent à des utilisateurs d'organiser l'information en taxonomie des concepts, chacune avec leurs attributs,et décrivent des relations entre ces concepts.Quand des données sont présentées ou annotées par des ontologies, les logiciels peuvent mieux comprendre leurs sémantiques, ce qui facilite la localisation et l'intégration des données pour des objectifs divers.

Bien que l'idée du web sémantique reste encore largement latente, un développement important a été réalisé dans le domaine de l'ingénierie de l'ontologie [Euzenat and Valtchev, 2004] ; [Shabolt and al.,2006] . Les normes comme RDF [1] et OWL [2] ont été présentées,ainsi qu'un certain nombre d'outils de traitement d'ontologies.Des domaines tels que les bibliothèques, les soins de santé ou la biotechnologie ont adopté l'utilisation des ontologies. Cependant, dans cet environnement ouvert et hétérogène, il est peu probable qu'une ontologie globale couvrant l'ensemble des systèmes distribués puisse être développée. Dans la pratique, les ontologies de différents systèmes sont développées indépendamment les unes des autres par des communautés différentes. Les agents logiciels (ou des outils en général) doivent apprendre à traduire entre ces ontologies connexes.

Ainsi, si les connaissances et les données doivent être partagées, il est essentiel d'établir des correspondances sémantiques entre les ontologies qui les décrivent. La tâche d'alignement d'ontologies (recherche de mappings, appariements ou mises en correspondance entre les concepts des deux ontologies) est particulièrement

1. RDF : Ressource Description Framework www.w3.org
2. OWL : Ontology Web Language www.w3.org

importante puisqu'elle autorise la prise en compte conjointe de ressources décrites par des ontologies différentes [Euzenat and Shvaiko,2013].

0.2 Problematique :

Pour demeurer compétitives, de plus en plus d'entreprises sont amenées à collaborer, de manière opportuniste ou stabilisée, d'une façon durable ou éphémère, avec d'autres entreprises dans l'objectif d'optimiser les coûts et les délais de production. Dans ce cadre d'applications distribuées, l'évolution des récents développements des systèmes à base de connaissances fait apparaitre la nécessité de spécifier une interprétation commune des informations échangées. Cela peut être effectué en utilisant des standards ou en adaptant ces systèmes pour qu'ils interprètent les informations sans ambiguïté, on parle alors d'interopérabilité. Par ailleurs, pour assurer l'interopérabilté entre les SI à base d'ontologies, deux facteurs déterminants conditionnent la réussite de celle-ci : l'interopérabilité syntaxique et l'interopérabilité sémantique. L'interopérabilité syntaxique n'aura lieu que lorsque les informations et les services qui les traitent ont la même syntaxe. Cela fait naitre l'interopérabilité sémantique qui vise à donner une sémantique aux informations échangées et représentées dans des syntaxes différentes et à s'assurer que cette sémantique soit commune à tous les systèmes entre lesquels des échanges doivent être mis en oeuvre. L'alignement des ontologies qui est un domaine de recherche actif est considéré comme une solution clé pour résoudre le problème d'hétérogéneité sémantique. Un handicap majeur lors de l'exploitation des SI à base d'ontologies est la taille de celles-ci, ainsi que les temps de traitement associés durant l'opération d'alignement. En effet la performance d'un système d'appariement des SI à base d'ontologies conditionne la réussite de nombreuses applications sémantiques.

L'alignement des ontologies est une tâche cruciale dans plusieurs domaines d'application. Étant donné deux ontologies, l'alignement produit un ensemble de correspondances chacune liant deux entités (par exemple, des concepts, des instances, des propriétés, des termes, etc.) par une relation sémantique (équivalence, subsomption, incompatibilité, etc.), éventuellement munie d'un degré de confiance [Kengue and al.,2008].

Le résultat du processus d'alignement entre deux ontologies est un ensemble de correspondances. Cet ensemble est généralement réutilisé par la suite pour différents objectifs tels que : la création de nouvelles ontologies, fusion d'ontologies, migration des données entre ontologies, coordination des ontologies pour réaliser une tâche donnée ou traduire des requêtes formulées en fonction d'une ontologie vers une autre. Un système d'alignement d'ontologie est un système logiciel qui effectue le processus d'alignement. Le degré d'automatisation est une caractéristique importante de ce système.

Les systèmes d'appariement existants peuvent être classés comme suit :

- Les systèmes d'appariement interactifs :l'utilisateur est impliqué dans le processus d'extraction de l'alignement. Une façon de mettre en œuvre cette approche consiste à afficher toutes les paires d'entités avec leurs mesures de confiance et celles qui sont jugées les plus pertinentes par l'utilisateur sont sélectionnées.
- Les systèmes d'appariement semi-automatiques : présentent un ensemble de propositions d'alignement à l'utilisateur. L'utilisateur les approuve et peut in-

clure des applications supplémentaires manuellement,ceci est répété jusqu'à ce que l'alignement soit complet.

- Les systèmes d'appariement automatiques : les techniques de cette catégorie permettent de surmonter les difficultés inhérentes à l'approche interactive et réduisent la charge de l'utilisateur quant à l'identification et la maintenance de l'alignement entre ontologies. Les recherches réalisées dans le cadre de cette thèse se concentrent principalement sur l'alignement entièrement automatique.

Dans la plupart des scénarios, l'alignement d'ontologie devrait être fait sans la présence d'un utilisateur qualifié, donc l'alignement automatique est nécessaire. Plusieurs défis dans le domaine de l'alignement d'ontologies ont été mis en évidence dans des recherches récentes [Euzenat and Shvaiko,2013]. En effet, la création de multiples ontologies, parfois pour un même domaine conduit à une hétérogénéité entre les connaissances exprimées au sein de chacune d'entre elles qui doit être résolue : c'est la problématique d'interopérabilité.

Pour assurer celle-ci, la sélection des mesures de similarité appropriées ainsi que l'ajustement de la configuration de leur combinaison sont connus comme des questions fondamentales que la communauté devrait traiter. En effet, différents scénarios de correspondance peuvent nécessiter des ensembles de mesures de similarité différents, ce qui nécessite des réglages différents dans la fonction de combinaison.

De plus, la vérification de la cohérence sémantique de l'alignement découvert est également connue comme une tâche cruciale.Par ailleurs, la difficulté du problème croît avec la taille des ontologies.

En effet , l'alignement des ontologies larges conduit à une explosion de l'espace de calcul ; et par la suite, il nécessite beaucoup d'espace mémoire et de temps de calcul. Il est important de souligner à ce niveau que notre objectif principal de départ de notre travail était de contribuer à trouver des améliorations permettant aux systèmes d'information d'entreprises d'interopérer, nous nous sommes trouvé à un moment de nos recherches confrontés au problème du volume des ontologies qui peuvent être utilisées par ces systèmes, et qui représentait une limite sérieuse à la mise en œuvre de toute interopérabilité . A ce stade, nous nous sommes posé une question fondamentale, si on pouvait faire interopérer des systèmes d'information d'entreprise avec ce type d'ontologies, alors nous aurons apporté la solution idéale á la problématique d'intéropérabilité de ces systèmes. Au vu du nombre de recherches important effectuées dans ce domaine, et aux différentes stratégies mises en œuvre, nous nous sommes intérrogés pourquoi ces recherches apportaient des solutions limitées car elles n'arrivaient que difficilement à traiter l'intéropérabilité des systèmes d'information à base d'ontologies volumineuses ou larges. En effet, les différentes solutions à base d'alignement mettaient en œuvre des techniques de calcul de similarité qui se surpassaient pour atteindre de meilleurs résultats mais qui souffraient d'une anomalie incontournable : « la perte de sémantique » car toutes basées sur les stratégies de partitionnement ou de modularisation. A ce

niveau, nous nous sommes posé la question non pas celle relative à l'amélioration de l'intéropérabilité des systèmes d'information d'entreprise à base d'ontologies mais, plutôt celle liée à l'existence d'une nouvelle stratégie autre que le partitionnement ou la modularisation.

0.3 Objectif

Avant de présenter les contributions de notre travail de recherche en détail, nous clarifions le principal objectif que nous allons suivre dans cette thèse. En effet l'objectif principal de notre travail de recherche est de présenter une nouvelle stratégie d'intéropérabilité des systèmes d'information à base d'alignement d'ontologies. Pour ce faire, nous montrons que notre approche diffère des stratégies actuelles : le partitionnement et la modularisation .

Pourquoi ? Ces deux stratégies ont prouvées leurs limites lorsqu'il s'agissait du passage à l'échelle. Notre étude approfondie de l'état de l'art du domaine concernant tout ce qui touche à l'alignement nous a conduit à la conclusion que le problème n'était pas dû aux techniques d'alignement qui se basent en général sur le calcul de mesures de similarité et ayant leurs limites, mais plutôt sur la manière d'aborder la problématique qui se pose. Nos différentes recherches effectuées nous ont confortées dans notre approche. En effet, la manière de penser de l'être humain lorsqu'il s'agissait de rapprocher deux ensembles d'objets ne se fait pas par calcul, mais plutôt par rapprochement sémantique. A partir de ce constat, on s'est posé la question : « peut-on rapprocher deux ensembles d'objets sans techniques de calcul ? » ou tout simplement « est-ce que la machine est capable de rapprocher deux ensembles d'objets sans techniques de calcul ?». Si on arrivait à apporter une réponse à cette question, alors celle-ci ne sera pas spécifique à l'intéropérabilité des systèmes d'information d'entreprise à base d'ontologies seulement, mais à l'intéropérabilité en général de tout système d'information à base d'ontologies.

L'objectif de notre travail est de proposer une nouvelle stratégie capable de rapprocher deux systèmes d'information à base d'ontologies hétérogènes du même domaine en essayant d'imiter la manière de penser de l'être humain. Les résultats obtenus nous ont confortés dans notre approche et nous sommes arrivés à mettre au point une nouvelle stratégie d'alignement appelée ONTEM (ONTology Extraction Method) renforcée d'un prototype de même nom que nous développons dans notre présente thèse.

0.4 Contributions

L'intéropérabilité des systèmes d'information à base d'ontologies représente un challenge à relever car il conditionne le travail coopératif de plusieurs entreprises ou organisations à la recherche le plus-value.

L'alignement des ontologies est une tâche très importante dans de nombreuses applications. Au cours de ces dernières années, le processus d'alignement d'ontologies devient une étape nécessaire pour permettre l'interopérabilité sémantique au sein des systèmes distribués et des applications Web. Par conséquent, la conception d'un processus d'alignement d'ontologies automatique et efficace a une importance cruciale dans la mesure où celui-ci va assurer l'intéropérabilité entre des ontologies conçues par des communautés différentes. En détail, les contributions de cette thèse sont les suivantes :

- **Contribution** 1 **:** au cours des dernières années, en raison de la nature complexe du processus d'alignement d'ontologies, en particulier lorsque les ontologies considérées sont larges caractérisées par un nombre important d'entités, des méthodes approximatives ont été largement utilisées pour calculer un alignement optimal. Sachant que les stratégies existantes reposent essentiellement sur le partitionnement et la modularisation, nous avons défriché une nouvelle fois ces stratégies et nous sommes arrivés au résultat inattendu : l'existence d'une troisième piste était-t-elle possible ? Pour cela, il fallait donc étudier différents travaux de recherche dans le domaine de l'alignement d'ontologies. Elles sont toutes liées à la notion de calcul et de définition des mesures de similarité, alors que peu d'attention a été accordée à la question de savoir si d'autres stratégies pouvaient exister pour contourner les problèmes techniques de manque de mémoire et de temps longs d'éxécution. Partant de ces considérations, dans notre première contribution, nous avons proposé une nouvelle stratégie à base d'extraction d'entités ontologiques pour résoudre le problème de l'alignement.

  Notre première idée, a consisté à reformuler le problème de l'alignement des ontologies larges comme étant un problème dépendant de la théorie des ensembles. Le principe de notre approche a consisté à exploiter les opérations algébriques pour résoudre efficacement le problème d'extraction des entités communes aux deux ontologies. Nous évitons dans notre stratégie toute décomposition ainsi que tout calcul de mesure de similarité. Toute perte de sémantique est donc évitée.

- **Contribution** 2 **:** Nous proposons un prototype nommé ONTEM basé sur les opérations algébriques permettant d'obtenir automatiquement les alignements. Nous soulignons que ONTEM peut être utilisé par des utilisateurs qui ne sont pas nécessairement des experts.

0.5 Plan du Manuscrit

Ce manuscrit comporte 6 chapitres.

Chapitre 1 : Ontologies

Ce chapitre commence par une introduction générale des terminologies principales et des définitions utilisées au cours de notre travail. Nous présentons la vision du web sémantique et nous expliquons le terme ontologie et les règles utilisées dans ce contexte.

Chapitre 2 : Intéropérabilité des systèmes d'information à base d'ontologies

Nous avons commencé ce chapitre par la description des différents types d'hétérogénéité et les différents domaines d'application.Nous avons expliqué les différents types d'intéropérabilité. Ensuite, nous avons donné, la définition formelle du processus d'alignement d'ontologie. Enfin nous avons décrit les différentes mesures de similarité. Ce chapitre présente aussi l'état de l'art des systèmes et des outils développés.

Chapitre 3 : Stratégies d'alignement

Dans ce chapitre nous présentons un état de l'art correspondant aux travaux sur le partitionnement et la moduralisation. Nous décrivons ensuite trois méthodes de partitionnement qui sont adaptées à l'alignement des ontologies de très grande taille. Ces méthodes prennent en compte au plus tôt l'information fournie par les ancres (les concepts des deux ontologies qui ont exactement les mêmes labels) pour l'utiliser dès le départ dans la phase de partitionnement, ce qui en fait des approches orientées alignement.

De même nous présentons les différentes méthodes de modularisation ainsi que les différentes approches d'extraction correspondantes.

Chapitre 4 : ONTEM, une nouvelle stratégie d'alignement des ontologies larges

Dans ce chapitre, nous décrivons ONTEM, une méthode d'extraction des entités ontologiques pour l'alignement des ontologies larges dans le cadre du processus d'intéropérabilité. L'architecture de notre méthode est présentée et détaillée. Les entrées sont les ontologies qui doivent être alignées. La sortie est un alignement entre les ontologies qui consiste en un ensemble de mappings qui sont acceptés après validation. Nous étudions ensuite l'apport des techniques d'alignement de ONTEM. Le processus d'alignement comprend quatre étapes principales : le prétraitement, l'identification des entités communes, la génération des mappings et la génération de l'alignement.

Chapitre 5 : Expérimentations

Dans ce chapitre nous décrivons le Benchmark OAEI LargeBio Track 2018. Nous étudions ensuite l'apport des techniques d'alignement du prototype ONTEM. Nous détaillons pour chaque tracks du benchmark les résultats obtenus par ONTEM et son classement par rapport aux meilleurs systèmes d'alignement des ontologies larges actuels. -

Chapitre 6 : Conclusion et Perspectives

Dans ce dernier chapitre, nous présentons des conclusions en récapitulant les avantages de notre stratégie d'alignement d'ontologies larges et son apport pour régler les problèmes d'hétérogéneité afin de permettre l'intéropérabilité entre différents systèmes, ainsi que les travaux futurs.

Ontologies

1

Sommaire

1.1 Introduction

De nos jours, la modélisation formelle joue un rôle central pour les problèmes, pour lesquels on dispose uniquement des connaissances de nature linguistique. Le domaine de l'intelligence artificielle a été introduit à cet effet. Son objectif primordial est la représentation des connaissances en utilisant un langage formel.

Une branche de l'intelligence artificielle, appelée l'ingénierie des connaissances s'intéresse, notamment, en l'étude des systèmes experts. Ces derniers avaient comme objectif la résolution automatique de problèmes. Les systèmes à base de connaissances ont été développés par la suite pour permettre le stockage, la consultation et le raisonnement automatique sur les connaissances stockées.

Actuellement, les ontologies constituent un enjeu stratégique dans la représentation et la modélisation des connaissances. Récemment, elles ont été introduites pour formaliser les connaissances dans les systèmes experts. Elles définissent les primitives indispensables pour leur représentation, ainsi que leur sémantique dans un contexte particulier.

Dans une première partie de ce chapitre, nous nous attachons à décrire les différentes définitions du concept d'ontologie en fonction du contexte dans lequel il est utilisé, ainsi que les différents éléments constituant l'ontologie. Un rapide aperçu des formalismes de représentation d'ontologies est ensuite donné. Puis, nous passons en revue les différentes étapes intervenant dans la construction des ontologies. Un résumé des principaux langages utilisés est présenté. Finalement, nous détaillons les différents critères permettant d'établir une typologie des ontologies. Cette classification fait ressortir l'intérêt de produire un modèle de description d'applications s'appuyant conjointement sur des ontologies de domaine. Nous évoquons, enfin, les apports essentiels des ontologies dans le cadre des applications de l'intelligence artificielle.

1.2 Notion d'ontologie

Historiquement, l'ontologie est un concept philosophique. Il désigne la science de l'être en général. L'ontologie décrit une théorie à propos de la nature de l'existence selon le paradigme " On ne cherche pas à comprendre le monde mais à le représenter " [Roche,2005] . Plus tard, l'ontologie est apparue en pleine lumière dans le domaine de l'intelligence artificielle, afin de résoudre les problèmes de modélisation des connaissances et, plus précisément, en ingénierie des connaissances. Ceci a engendré de nombreuses définitions que nous allons résumer dans les paragraphes suivants en présentant les points de vue philosophique et informatique.

Ces points de vue sont reliés car les entités que l'on peut modéliser et représenter en intelligence artificielle (en informatique) sont des entités qui existent dans le monde. Elles peuvent être décrites dans un langage naturel par des linguistes, par exemple. Elles peuvent être ensuite représentées formellement afin de les utiliser par des machines. En conséquence, les ontologies permettent de confronter les points de vue des informaticiens et des linguistes, les besoins en ingénierie des connaissances et les acquis de la sémantique linguistique. D'une manière gé-

nérale, elles permettent de développer des méthodologies pour la modélisation de données linguistiques. Les ontologies se construisent dans un domaine donné afin d'atteindre un objectif spécifique.

1.2.1 Définitions issues de la philosophie

Dans le domaine de la philosophie, l'ontologie est considérée comme une branche de la métaphysique qui s'intéresse à la nature et l'organisation de la réalité. Elle a une signification plus large, celle de « science de ce qui existe » dans laquelle on ne cherche pas à expliquer le monde, mais à le représenter. Elle s'applique à « l'être en tant qu'être», indépendamment de ses déterminations particulières. Cette définition philosophique est tirée de deux définitions issues respectivement du dictionnaire de l'Académie française et du dictionnaire Webster.

— En tant que que discipline de la philosophie, intéressons nous à la toute dernière définition de l'Académie française. " ONTOLOGIE n.f.XV IIe siècle. Emprunté du latin scientifique Ontologie, lui-même composé à l'aide de Onto-, tiré du grec ôn, ontos , « étant, ce qui est », et Logia, tiré du grec logos, « discours, traité ». Partie de la philosophie qui a pour objet l'être en tant qu'être, qui étudie les propriétés générales de l'être" [Academie française,2005] .
— Dans le dictionnaire Webster, l'ontologie est définie comme, "a branch of metaphysics concerned with the nature and relations of being ", " a particular theory about the nature of being or the kinds of existents ", et " a theory concerning the kinds of entities and specifically the kinds of abstract entities that are to be admitted to a language system" [Webster,2004].

La définition de l'Académie a le mérite d'être claire. Mais un sens nouveau est apparu depuis une vingtaine d'années. Il s'agit d'une représentation formalisée d'ensembles de concepts d'un domaine donné, assortis de leurs relations, utilisée en informatique. Le terme s'utilise le plus souvent au pluriel, des ontologies. Au singulier, l'ontologie désigne la discipline qui traite de la modélisation des concepts et de leurs relations en vue de leur exploitation par des machines. Cette définition est plus proche du point de vue de l'intelligence artificielle, comme on le verra dans ce qui suit.

1.2.2 Définitions issues de l'intelligence artificielle

L'intelligence artificielle a permis de représenter les connaissances d'un domaine sous forme de base, dite base de connaissances, et d'automatiser leur utilisation et la résolution de problèmes autour, à travers des inférences de données. Néanmoins, les bases de connaissances ne sont, globalement, pas réutilisables, ce qui limite leur utilisation. La notion d'ontologie a été introduite, entre autres, pour pallier cette limite.

D'une manière générale, une ontologie est vue comme un ensemble de concepts permettant de modéliser un ensemble de connaissances dans un domaine donné.

Un concept peut présenter plusieurs sens thématiques. Les concepts sont liés entre eux par des relations sémantiques, des relations de composition et d'héritage. Afin de préciser cette notion, de nombreux chercheurs ont proposé des définitions qu'il est utile d'examiner :

- Une définition générale a été donnée par Thomas R. Gruber où décrit il une ontologie comme une spécification explicite d'une conceptualisation modélisant des concepts et les relations entre concepts. Thomas R. Gruber défend ce point de vue comme suit : " An ontology is a specification of a conceptualization. That is, an ontology is a description (like a formal specification of a program) of the concepts and relationships that can exist for an agent or a community of agents. This definition is consistent with the usage of ontology as set-of-concept-definitions, but more general " [Gruber,1993] .
- Plus tard, John F. Sowa a spécifié de façon plus précise cette otion. Dans sa définition, l'ontologie est vue comme un catalogue de types issus de l'étude des catégories d'entités abstraites et concrètes qui existent ou peuvent exister dans un domaine. Elle est définie comme suit "The subject of ontology is the study of the categories of things that exist or may exist in some domain. The product of such a study, called an ontology, is a catalogue of the types of things that are assumed to exist in a domain of interest D from the perspective of a person who uses a language L for the purpose of talking about D. The types in the ontology represent the predicates, word senses, or concept and relation types of the language L when used to discuss topics in the domain D" [Sowa,2000].
- Après, afin de compléter le sens philosophique originel, Guarino a introduit la notion d'ontologie formelle, qui est définie en tant que modélisation conceptuelle, ou une représentation de cette modélisation. "Une ontologie est un accord sur une conceptualisation partagée et éventuellement partielle" [Guarino and Giaretta,1995] .
- De même, Uschold définit une ontologie comme une description formelle d'entités et de leurs propriétés, relations, contraintes et comportements. De plus, les auteurs ont introduit, dans [Ushold and Grüninger,1996] , la notion de l'ontologie explicite "An explicit ontology may take a variety of forms, but necessarily it will include a vocabulary of terms and some specification of their meaning".
- Enfin, Christophe Roche a donné une définition générique et simple "Une ontologie est une conceptualisation d'un domaine à laquelle sont associés un ou plusieurs vocabulaires de termes. Les concepts se structurent en un système et participent à la signification des termes. Une ontologie est définie pour un objectif donné et exprime un point de vue partagé par une communauté. Une ontologie s'exprime dans un langage (représentation) qui repose sur une théorie (sémantique) qui garantit des propriétés de l'ontologie en termes de consensus, cohérence, réutilisation et partage" [Roche,2005] .

Nous retenons cette dernière définition car elle englobe et résume les définitions précédentes. Dans ce qui suit, nous détaillons, d'une manière non exhaustive, les éléments qui constituent l'ontologie.

1.3 Éléments constitutifs de l'ontologie

Les ontologies sont, à l'heure actuelle, au cœur des travaux menés en ingénierie des connaissances. Elles permettent de représenter les connaissances et les manipuler automatiquement, tout en gardant leur sémantique. Les connaissances sont définies à travers des concepts. Les liens entre concepts sont appelés relations. Afin de relier les concepts, l'ontologie se présente, généralement, sous forme d'une organisation hiérarchique des concepts.

1.3.1 Concepts

Selon Uschlod et Grüninger [Ushold and Grüninger,1996] , un concept peut représenter un objet matériel, une notion ou une idée. C'est une représentation de l'esprit qui abrège et résume une multiplicité d'objets empiriques ou mentaux par abstraction et généralisation des traits communs identifiables.

L'ensemble des propriétés d'un concept s'appelle sa compréhension ou son intension, et l'ensemble des objets ou êtres qu'il englobe, son extension. L'intension du concept peut généraliser un ensemble de propriétés qualitatives ou fonctionnelles communes aux individus auxquels il s'applique. L'extension du concept est représentée par un ensemble d'objets qui sont appelés instances du concept.
Exemple. : Une bibliothèque scolaire est caractérisée par un bibliothécaire, des livres de différents domaines, des magazines, etc. On peut définir un concept " C " pour désigner les livres, par exemple. L'intension de ce concept peut être LIVRE, et son extension pourra être son domaine : livre de programmation, livre d'ingénierie des connaissances, etc.

Les concepts sont organisés en taxonomie au sein d'un réseau de concepts et peuvent être structurés hiérarchiquement. Un concept est caractérisé par un ensemble de propriétés. Dans [Fürst,2002], l'auteur s'inspirait des propriétés proposées par Guarino. Nous donnons les plus intéressantes :

- Un concept est générique s'il n'admet pas d'extension. La vérité, par exemple, n'a pas d'extension.
- Un concept porte une propriété d'identité si cette propriété permet de différencier deux instances de ce concept. Par exemple, dans un système de gestion de fichier, un nom désigne d'une manière unique un fichier ou un répertoire. Le nom est une identité du fichier ou répertoire.
- Un concept est rigide s'il ne peut pas être une instance d'autres concepts. Par exemple, l'être vivant est un concept rigide, mais un " être humain " n'est pas un concept rigide, car l'humain est une instance du concept " être vivant ".
- Un concept est anti-rigide s'il peut être une instance pour d'autres concepts. Comme le cas de l'être humain dans l'exemple précédent.

Les concepts peuvent être équivalents s'ils ont la même extension. Ils peuvent être disjoints ou incompatibles si leurs extensions sont disjointes. Ils peuvent aussi être dépendants : un concept C1 est dépendant de C2 si, pour toute instance de

C1, il existe une instance de C2.

Pour la suite de notre étude, on emploie, tout au long du manuscrit, le mot concept pour désigner l'intension de concept et le mot instance pour désigner un élément de l'ensemble constituant l'extension de concept.

1.3.2 Relations entre concepts

Les concepts (respectivement, les instances) peuvent être reliés entre eux par des relations au sein d'une ontologie. Une relation est définie comme une notion de lien entre des entités, exprimée souvent par un terme ou par un symbole littéral ou autre. Généralement, les liens sont classés en deux catégories : des liens hiérarchiques et des liens sémantiques. La relation hiérarchique reprend la structuration d'Hyperonymie/Hyponymie, tandis que la relation sémantique lie les concepts par un lien, dit partie-tout ou (whole-part), qui correspond à la structuration d'H olonymie/Meronymie.

Une relation hiérarchique lie un élément supérieur, dit l'hyperonyme, et un élément inférieur, dit l'élément hyponyme, ayant les mêmes propriétés que le premier élément avec au moins une en plus. Dans certains cas, le couple (Hyperonymie, Hyponymie) s'interprète par (Type, Sous-Type). L'hyperonyme englobe l'hyponyme. On pourra alors écrire « HYPONYME est une sorte de HYPERONYME ». Exemple : Le chat est une sorte d'animal, donc, " animal " est l'hyperonyme de chat.

Une relation sémantique, comme la relation partie-tout, parfois appelée " Partie-de ", ou (holonyme / méronyme) [1] est une relation hiérarchique qui existe entre un couple de concepts dont l'un dénote une partie et l'autre dénote le tout. La relation partie-tout est différente de celle d'hyponymie par le fait qu'un hyperonyme impose ses propriétés à ses hyponymes, par contre le TOUT dispose des propriétés qui ne sont pas obligatoirement transmises à ses parties. Exemple : Dans le corps humain, la tête et les jambes font partie du corps mais elles ne disposent pas des mêmes propriétés. Tête n'est pas une sorte de Corps (voir la figure 1.1).

Comme les concepts, les relations peuvent aussi avoir des propriétés. Ces dernières peuvent être algébriques (symétrie, réflexivité, transitivité). Elles peuvent être des propriétés de cardinalité, comme par exemple, un ordinateur qui dispose, d'au moins, un disque dur. En général,ces relations sont binaires.

1. X est un méronyme de Y si X est une partie de Y.
Y est un holonyme de X si X est une partie de Y

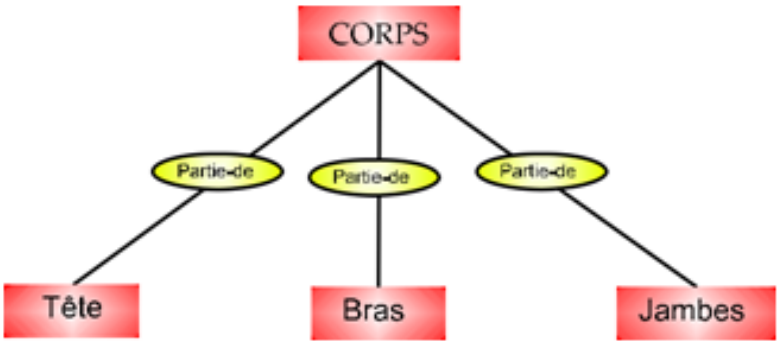

FIGURE 1.1 – *Exemple de la relation Partie-de*

Afin de décrire les concepts et relation d'une ontologie, celle-ci s'exprime dans un langage et repose sur un formalisme. Ceci fera l'objet des prochaines sections.

1.4 Formalismes de représentation :

Une ontologie, telle qu'elle est décrite dans la section précédente (1.2), a besoin d'être représentée formellement. Plus encore, elle doit représenter l'aspect sémantique des relations liant les concepts. A cet effet, de nombreux formalismes ont été développés [Gaëlle,2002].

1.4.1 Les réseaux sémantiques

Un réseau sémantique est une structure de graphe qui encode les connaissances ainsi que leurs propriétés. Les nœuds du graphe représentent des objets (concepts, situations, événements, etc) et les arcs expriment des relations entre ces objets. Ces relations peuvent être des liens " sorte - de " exprimant la relation d'inclusion ou des liens " est-un " représentant la relation d'appartenance. Par exemple, on peut dire que Volkswagen est une marque de voiture (voir la figure 1.2), comme on peut dire que le busard est un rapace qui est une sorte d'oiseau (Voir la figure 1.3).

En fait, une ontologie est considérée comme un réseau sémantique. Elle regroupe un ensemble de concepts décrivant complètement un domaine. Ces concepts sont liés les uns aux autres par des relations, d'une part, taxonomiques (hiérarchisation des concepts) et, d'autre part, sémantiques.

Parmi les réseaux sémantiques, très répandus pour la conceptualisation des ontologies, on trouve les graphes conceptuels dont le but fondamental est d'être "un système de logique hautement expressif, permettant une correspondance directe avec la langue naturelle " [Sowa,1992]. Ce type de graphes constitue un formalisme général de représentation de connaissances fondé sur la logique. Il s'inscrit dans la

continuité des graphes existentiels de Charles Sanders Peirce [2].

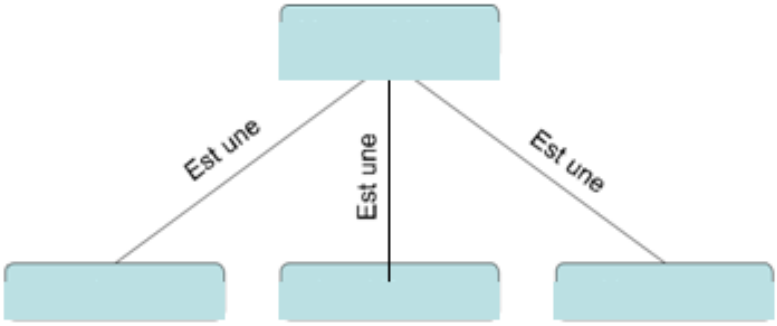

FIGURE 1.2 – *Exemple de réseau sémantique utilisant la relation " est-un "*

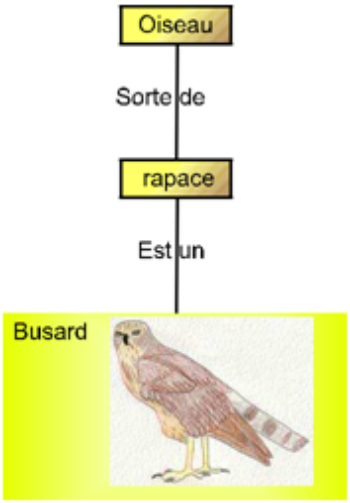

FIGURE 1.3 – *Exemple de réseau sémantique utilisant la relation " sorte-de "*

Les graphes conceptuels ont été mis au point par John F. Sowa pour modéliser une ontologie de haut niveau [Sowa,2000]. Un graphe conceptuel est un graphe étiqueté, biparti, connexe et fini. Les sommets représentent les entités, attributs, états ou évènements. Chaque sommet est typé. Ces types sont ordonnés dans une structure de treillis orienté du plus spécifique au plus général avec des relations "sorte-de ".

Afin de définir un graphe conceptuel, le langage CGIF (Conceptual Graph Interchange Form) a été développé. CGIF est une représentation concrète des graphes conceptuels dans laquelle chaque graphe est traduit dans une représentation logique équivalente.

L'intérêt de ces graphes réside dans leur non-ambiguïté et leur facilité d'utilisation. Ceci a incité les concepteurs de plusieurs applications à les utiliser, que ce

2. http ://plato.stanford.edu/entries/peirce/

soit dans l'acquisition des connaissances, la recherche d'informations et le raisonnement sur la connaissance conceptuelle. Un autre intérêt des graphes réside dans le fait qu'ils reposent sur la logique du premier ordre.

1.4.2 Les schémas

La notion de " schéma " (ou frame) est apparue en 1932 dans le domaine de la psychologie. Plus tard, les schémas ont été introduits en intelligence artificielle par Minsky afin de résoudre les problèmes de la vision par ordinateur. Ils ont été définis dans [Minsky,1975] par : " A frame is a data-structure for representing a stereotyped situation, like being in a certain kind of living room, or going to a child's birthday party. Attached to each frame are several kinds of information. Some of this information is about how to use the frame. Some is about what one can expect to happen next. Some is about what to do if these expectations are not confirmed. We can think of a frame as a network of nodes and relations. The "top levels " of a frame are fixed, and represent things that are always true about the supposed situation. The lower levels have many terminals - "slots " that must be filled by specific instances or data. Each terminal can specify conditions its assignments must meet (assignments themselves are usually smaller "sub-frames "). Simple conditions are specified by markers that might require a terminal assignment to be a person, an object of sufficient value, or a pointer to a subframe of a certain type. More complex conditions can specify relations among the things to several terminals ".

En conséquence, un schéma est une structure de données complexe. Il est considéré comme un prototype décrivant une situation ou un objet standard. Il sert de référence pour comparer des objets que l'on désire reconnaître, analyser ou classer. Les prototypes doivent prendre en compte toutes les formes possibles d'expression de la connaissance. Un schéma, comme le montre la figure 1.4, est caractérisé par des attributs, des facettes et des relations.

- Les attributs définissent la structure de données ;
- Les facettes définissent la sémantique des attributs et décrivent l'ensemble des valeurs possibles pour cet attribut. Elles peuvent être de deux formes : déclaratives et procédurales. Les premières associent des valeurs aux attributs, alors que les secondes décrivent les procédures appelées réflexes, activées lors des accès à ces valeurs. Les schémas sont organisés dans une structure hiérarchisée d'héritage d'attributs ;
- Les relations expriment la sémantique d'héritage. Elles peuvent être générales (spécialisation, composition) ou spécifiques à une application.

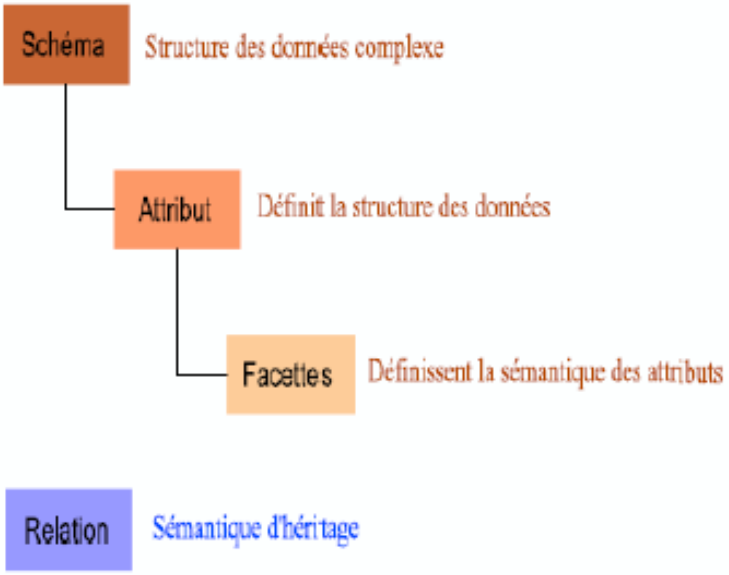

FIGURE 1.4 – *Éléments caractérisant un schéma*

1.4.3 Les scripts

La notion de " script " (ou scénario) a été introduite par Schank et Abel, sur le modèle des schémas pour le traitement du langage naturel. Ils ont défini un script par : " A script is a structure that describes appropriate sequences of events in a particular context. A script is made up of slots and requirements about what can fill those slots. The structure is an interconnected whole, and what is in one slot affects what can be in another. Scripts handle stylized everyday situations. They are not subject to much change, nor do they provide the apparatus for handling totally novel situations. Thus, a script is a predetermined, stereotyped sequence of actions that defines a well-known situation. Scripts allow for new references to objects within them just as if these objects had been previously mentionned ; objects within a script may take " the " without explicit introduction because the script has already implicitly introduced them " [Shank and Abelson,1988].

Un script est donc une structure de données qui regroupe des connaissances relatives à une situation et qui permet de combiner des représentations. Il peut être vu comme un ensemble d'actions élémentaires ou de références à d'autres scénarios, ordonnées selon leur déroulement dans le temps.

1.5 Construction des ontologies

Dans la littérature, il n'y a pas un consensus sur une méthodologie fixe pour la construction des ontologies. Plusieurs méthodologies ont été proposées dont la plus connue est celle proposée et fondée sur l'expérience de la construction de l'Entreprise Ontology par Uschold et Grüninger [Ushold and Grüninger,1996] . Comme la méthode est générique, ses étapes sont considérées comme la base

d'un processus standard de construction. Ce processus opère selon quatre étapes fondamentales (Voir la figure 1.5) :

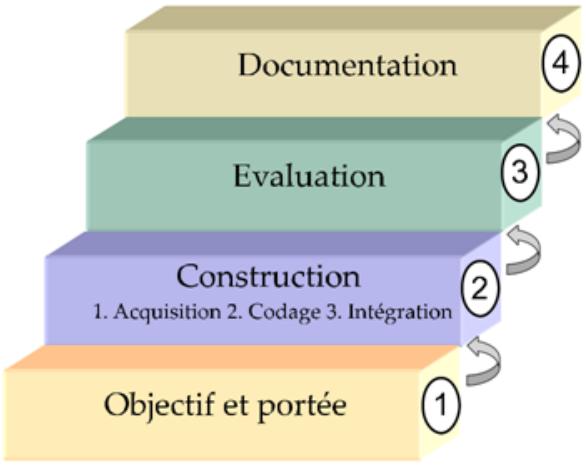

FIGURE 1.5 – *Étapes pour la construction des ontologies*

L'identification de l'objectif permet d'identifier, en termes généraux, l'objectif, la portée et les limitations de l'ontologie à construire.

La création de l'ontologie, étape la plus longue et la plus difficile, contient elle-même trois sous-étapes :

1. L'acquisition des connaissances sert à définir les concepts dans un domaine donné et les relations entre eux, de manière à ne pas être ambiguës.Différentes techniques permettent de faire l'acquisition des connaissances. Elles peuvent se matérialiser par des entretiens informels avec des experts ou des entretiens structurés en vue de collecter des connaissances spécifiques et détaillées sur les concepts, leurs instances et leurs relations. Elles peuvent également être obtenue sous forme d'analyse informelle de texte pour définir les concepts fondamentaux ou bien sous forme d'une analyse formelle afin de définir les structures des connaissances [Fernandez and al.,1997].
2. Le codage, une fois les concepts et leurs relations acquises, permet de représenter l'ontologie dans un langage formel. La formalisation de l'ontologie peut être de différents degrés [Brisson,2004] .
   — Très informel : l'ontologie s'exprime dans le langage naturel ;
   — Semi-informel : l'ontologie s'exprime dans une forme structurée du langage naturel ;
   — Semi-formel : l'ontologie est exprimée dans un langage artificiel défini formellement ;
   — Rigoureusement formel : l'ontologie est exprimée dans un langage formel utilisant une sémantique formelle avec des théorèmes et preuves.
3. L'intégration des ontologies existantes est l'étape qui permet de réutiliser les concepts déjà définis dans des ontologies existantes.

L'évaluation de l'ontologie a été définie en s'inspirant des travaux de Gomez-Perez [Gómez-Pérez and al., 1995] :"to make a technical judjment of the ontologies, their associated software environment, and documentation with respect to a frame of reference...The frame of reference may be requirements specifications, competency questions, and/or the real world ". D'un autre coté, Gruber a proposé quelques critères pour l'évaluation d'une ontologie :

- La clarté : les concepts de l'ontologie doivent présenter le sens voulu des termes ;
- La cohérence : les raisonnements construits à partir des axiomes d'une ontologie ne doivent pas aboutir à des contradictions ;
- L'extensibilité : l'ontologie doit être conçue de manière à ce qu'une nouvelle utilisation se fasse sans remettre en cause ce qui a été précédemment conçu ;
- Le biais d'encodage minimum : la spécification de l'ontologie doit être aussi indépendante que possible d'un méta-langage particulier de représentation ;
- L'engagement ontologique minimal : l'objectif est de permettre la spécialisation des spécifications d'une ontologie donnée selon des besoins réels ;

La documentation permet de renseigner les ontologies, leurs concepts importants ainsi que leurs objectifs. Le formalisme Ontolingua, représente l'un des éditeurs qui facilite cette étape en donnant des documentations formelles et informelles.

Une autre méthode, appelée METHONTOLOGY, a été définie par Gomez-Perez à l'université Polytechnique de Madrid [Fernandez and al.,1997]. Elle donne les étapes à suivre pour réaliser chaque activité du domaine étudié, les techniques utilisées, les produits à mettre au point ainsi que la façon de les évaluer. Elle couvre le cycle de vie d'une ontologie : planification, supervision, assurance qualité, spécification, acquisition de connaissances, conceptualisation, intégration, formalisation, implémentation, évaluation, maintenance, documentation et gestion de configuration. METHONTOLOGY est implémentée dans l'environnement WebODE [3] .

Il existe trois types de méthodes pour la construction d'ontologie : des méthodes manuelles, automatiques et mixtes. Dans la première, les experts créent une nouvelle ontologie d'un domaine ou étendent une ontologie existante comme par exemple l'ontologie Wordnet [4]. Dans la méthode automatique, l'ontologie est construite par des techniques d'extraction des connaissances : les concepts et leurs relations sont extraits de bases de connaissances et ensuite vérifiés par les inférences. Enfin, la méthode mixte, les techniques automatiques permettent d'étendre des ontologies qui ont été construites manuellement comme la base des connaissances Cyc [5].

3. WebODE est une plate-forme de conception d'ontologies fonctionnant en ligne. Elle fait suite à ODE, un éditeur qui assurait fidèlement le support de la méthodologie maison METHONTOLOGY
4. http ://wordnet.princeton.edu/
5. http ://www.opencyc.org/

1.6 Langages et plateformes pour les ontologies

Il existe de nombreux langages informatiques, plus ou moins récents, spécialisés dans la création et la manipulation des ontologies. Nous en décrivons quelques-uns dans la suite.

1.6.1 XML, RDF et OWL

Les ontologies peuvent être décrites en XML (Extensible Markup Language). C'est un langage qui permet de décrire des méta-données en facilitant leurs traitements et leurs échanges.

D'autre part, RDF (Resource Description Framework) est un modèle de graphe destiné à décrire, de façon formelle, les ressources Web et leurs méta-données et permettre le traitement automatique de telles descriptions.

RDFS (Resource Description Framework Schema) est un langage extensible qui permet la représentation des connaissances. Il appartient à la famille des langages du Web sémantique publiés par le W3C. Il fournit des éléments de base pour la définition d'ontologies ou de vocabulaires destinés à structurer des ressources RDF. Dublin Core est un schéma générique de méta-données qui permet de décrire des ressources numériques ou physiques et d'établir des relations avec d'autres ressources. Les déclarations de termes du Dublin Core sont représentées en RDFS.

Cependant, RDF et RDFS souffrent de limites, comme l'impossibilité de raisonner et de mener des raisonnements automatisés sur les modèles de connaissances établis à l'aide de ces langages. En conséquence, un nouveau langage, OWL (Web Ontology Language), est apparu.

Plus tard OWL (Web Ontology Language) est apparu. C'est un dialecte XML fondé sur une syntaxe RDF. Il fournit les moyens pour définir des ontologies Web structurées. Il se différencie du couple RDF / RDFS par le fait que c'est un langage d'ontologies, contrairement à RDF. Si RDF et RDFS apportent à l'utilisateur la capacité de décrire des classes et des propriétés, OWL intègre, en plus, des constructeurs de comparaison des propriétés et des classes : identité, équivalence, contraire, cardinalité, symétrie, transitivité, disjonction, etc. Ainsi, OWL offre aux machines une plus grande capacité d'interprétation du contenu web que RDF et RDFS, grâce à un vocabulaire plus large et à une vraie sémantique formelle.

De nombreux éditeurs d'ontologies sont apparus. Protégé est l'un des éditeurs d'ontologie les plus utilisés [6]. Il peut lire et sauvegarder des ontologies dans la plupart des formats d'ontologies : RDF, RDFS, OWL.

1.6.2 LOOM

LOOM est une plate-forme pour la représentation des connaissances. Son objectif principal est de construire des applications intelligentes. Les connaissances déclaratives dans LOOM sont composées de définitions, de règles, de faits, etc. Pour compiler les connaissances déclaratives, LOOM utilise un moteur déductif.

6. http ://protege.stanford.edu/

Ce dernier est un classifieur qui utilise le chaînage-avant, l'unification sémantique et des technologies orientées objet.

SUMO est l'une des ontologies utilisées dans LOOM par l'intermédiaire d'un outil SUMO2LOOM [Flatter,2003]. Motta a, lui aussi, montré qu'il est plus facile de compléter une ontologie existante que de partir de rien. Il utilise le langage OCML. Le projet, appelé WebOnto, consiste en une application Java couplée à un serveur Web qui permet de naviguer et d'éditer des modèles de connaissance [Enric and al.,2000].

1.6.3 ONTOLINGUA

Ontolingua est un mécanisme qui permet aux utilisateurs de créer et manipuler des ontologies[7]. Il supporte les ontologies portables pour qu'elles soient traduites dans différents sys-tèmes. Ontolingua est basé sur le langage d'interchange KIF(Knowledge Interchange Format).

Celui-ci est conçu pour l'échange de connaissances entre des systèmes informatiques répartis. Thomas Gruber a introduit la syntaxe et la sémantique utilisées dans KIF dans [Gruber,1993] . Ontolingua permet aussi de traduire des ontologies génériques en LOOM,KIF, etc.

1.6.4 OIL

OIL (Ontology Inference Layer) est un langage dédié à la spécification et à l'échange des ontologies sur le Web[8]. Il permet la représentation et l'inférence d'ontologies, en combinant des primitives de modélisation des langages de frame avec la sémantique formelle et les modes de raisonnement des logiques descriptives. Ainsi, il représente une ontologie par un conteneur (ontology container) et des définitions ontologiques (ontology definition). Pour cela, il se base sur des formalismes tels que RDF/RDFS et XML, ce qui garantit sa totale compatibilité avec ces formalismes standards ou des formalismes en cours de standardisation. Les liens existant entre la structure d'un document et la modélisation du domaine couvert par ce document sont étudiés dans [Klein and Loebbecke,2000] au travers d'une comparaison entre OIL et les schémas XML.

1.6.5 SHOE

SHOE (Simple HTML Ontology Extensions) est une extension du langage HTML qui permet aux auteurs de pages Web de générer une annotation de leurs documents, compréhensible par la machine[9]. Ce langage peut être utilisé par des agents pour la gestion des pages Web [Luke and al.,1997]. En effet, autant le langage HTML est utilisé pour rendre la connaissance facilement lisible par un humain, autant il n'est pas adapté pour permettre cette lisibilité à un système informatique. Un agent chargé d'extraire la sémantique d'un document a beaucoup de

7. http ://www.ksl.stanford.edu/software/ontolingua/
8. http ://www.ontoknowledge.org/oil/
9. http ://www.cs.umd.edu/projects/plus/SHOE/

difficulté à le faire, car les données et leur présentation sont entremêlées. SHOE évite ce problème grâce à sa propriété d'inclusion des données directement lisibles et exploitables dans les pages Web.

1.7 Typologie des ontologies

Les ontologies peuvent être classifiées selon plusieurs critères. Ces derniers sont explicités dans [PSYCHÉ and al.,2003] , afin de déterminer une typologie d'ontologies :

— l'objet de conceptualisation ;
— le niveau de détail ;
— le niveau de complétude ;
— le niveau de formalisme de représentation.

1.7.1 Typologie selon l'objet de conceptualisation

Cette typologie dispose des types d'ontologies suivants :

Les ontologies de représentation des connaissances regroupent les concepts impliqués dans la formalisation des connaissances. L'ontologie la plus citée dans ce contexte est l'ontologie de FRAME. Elle intègre les primitives de représentation des langages à base de frames : classes, instances, facettes, propriétés/slots,relations, restrictions, valeurs permises, etc.

Un autre exemple est l'ontologie de Sowa " Knowledge Representation (KR) ". C'est est une ontologie générique à visée universelle. Les concepts dans KR sont représentés par des prédicats unaires. De plus, un nouveau concept dans KR est la conjonction des prédicats unaires d'un niveau plus haut. Par exemple, le concept Proposition dans KR (voir la figure 1.6) est la conjonction des deux concepts de plus haut niveau Abstract et Relative.

Les ontologies supérieures ou de Haut niveau visent à étudier les catégories des choses qui existent dans le monde, comme les concepts de haut niveau d'abstraction tels que les entités, les événements, les états, les processus, les actions, le temps, l'espace, les relations et les propriétés. L'ontologie de haut niveau est fondée sur la théorie de l'identité, la méréologie (theory of whole part) et la théorie de la dépendance. Ces concepts sont indépendants d'un domaine ou d'un problème particulier.

Parmi ces ontologies on cite souvent " Upper Cyc ". Elle contient presque 3000 termes relatifs aux concepts les plus généraux sur les connaissances humaines " Every concept one can imagine can be correctly linked into the Upper Cyc ontology ". Elle a été construite en une douzaine d'années. Un exemple extrait de la figure 1.7, est la fonction de vérité qui est considérée comme concept de l'ontologie. Cette fonction peut être un prédicat, une connexion logique, etc.

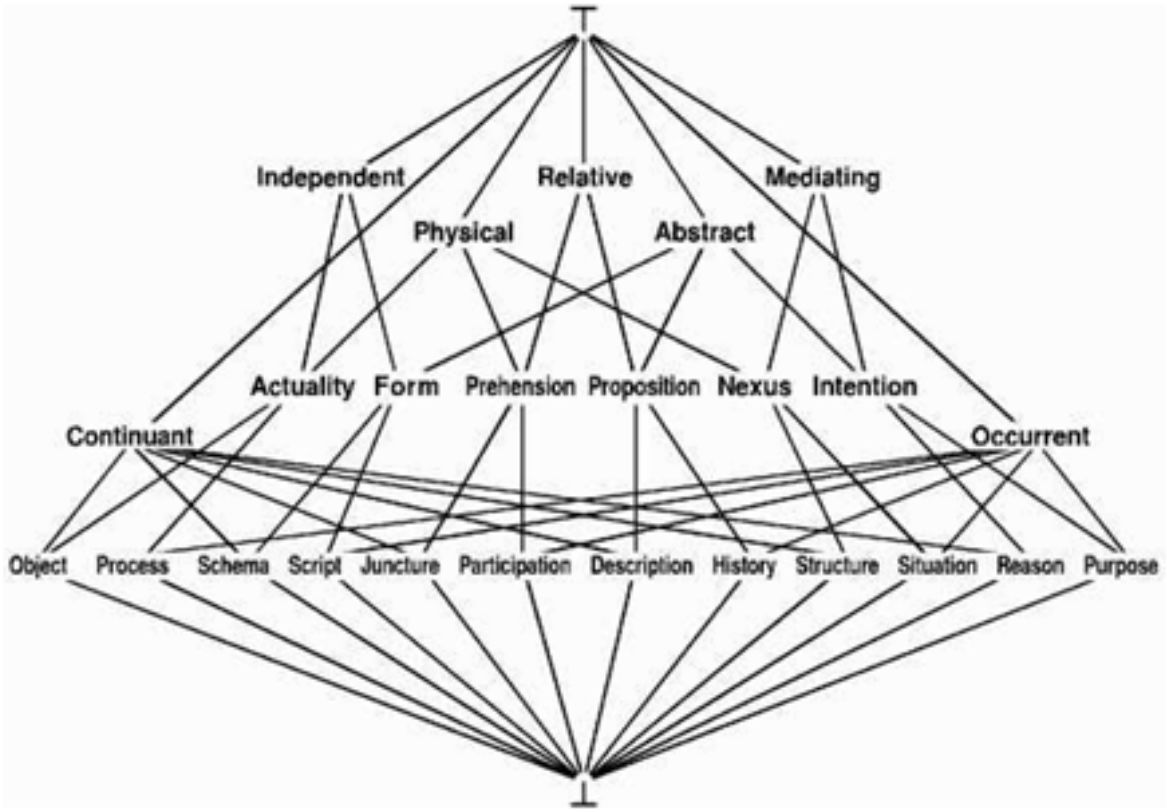

FIGURE 1.6 – *Diagramme KR de Sowa [Sowa,2000]*

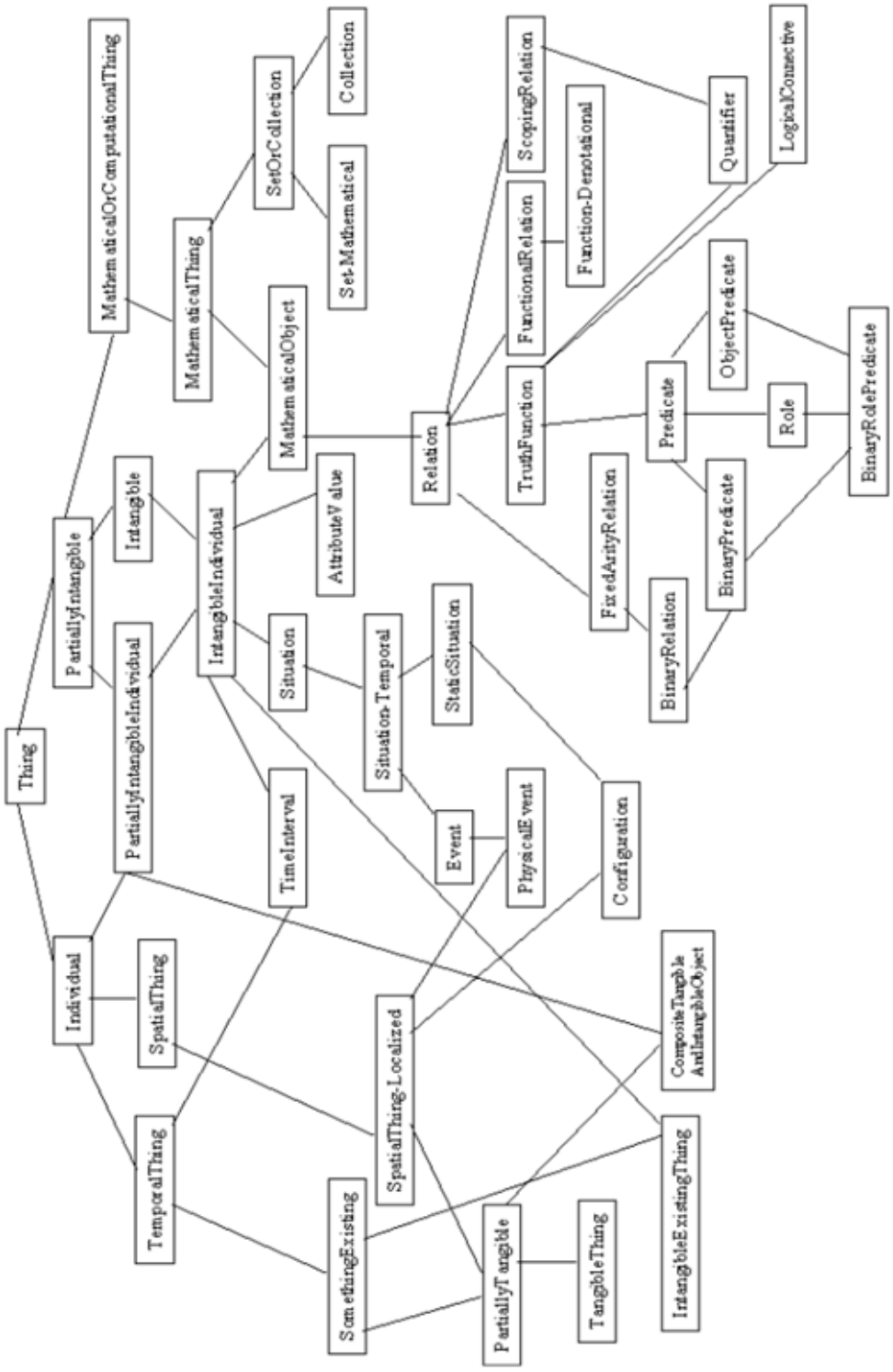

FIGURE 1.7 – *L'ontologie Upper Cyc* [10]

10. http ://www.openncyc.org

Les ontologies génériques sont appelées, également, des méta-ontologies ou-Core" ontologies". Elles décrivent des concepts génériques moins abstraits que ceux décrits par des ontologies supérieures. Dans cette classe, citons SUMO (Suggested Upper Merged Ontology) développée dans le cadre du projet IEEE SUO (Standar Upper Ontology). L'objectif assigné à SUMO est de constituer un standard pour permettre l'interopérabilité sémantique entre les systèmes d'information.
Actuellement, SUMO comporte plusieurs centaines de concepts et de relations généraux.

Dans la figure 1.8, nous donnons des exemples de ces concepts extraits de [Inaba and al., 2000].

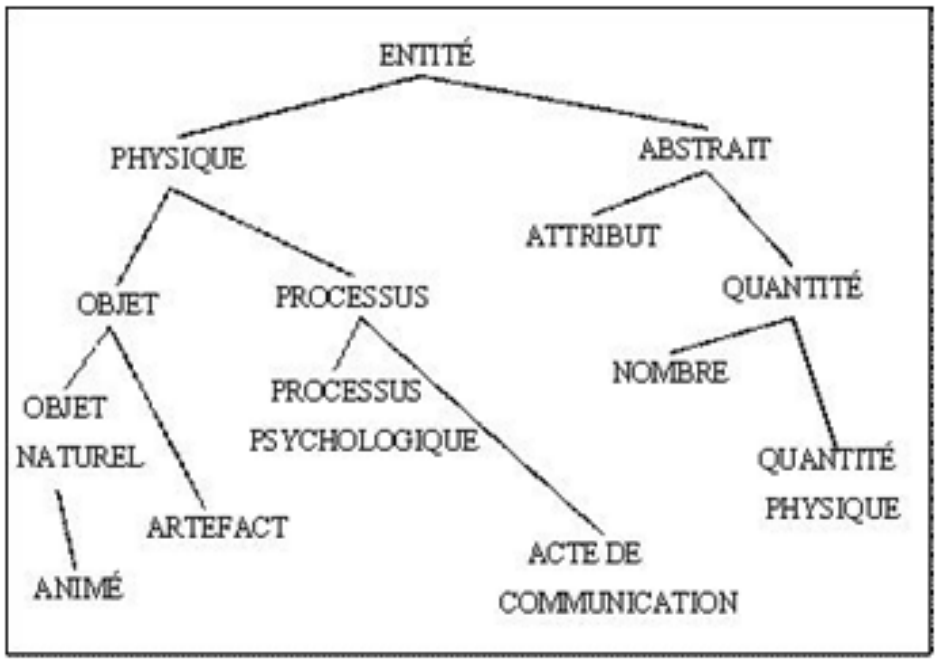

FIGURE 1.8 – *SUMO* [11] *[Inaba and al., 2000]*

Une autre ontologie générique a été développée WordNet. En fait, c'est un système de références lexicales croisées dont la conception a été inspirée par les théories actuelles de la mémoire linguistique humaine. Les noms de l'anglais, les verbes, adjectifs et adverbes sont organisés en ensembles de synonymes (synsets), représentant le concept lexical sous-jacent. Des relations relient les ensembles de synonymes entre eux [12].

Les ontologies de domaine décrivent le vocabulaire lié à des domaines particuliers comme la physique, la mécanique, la chimie, la médecine et la modélisation d'entreprise. Selon Guarino [Guarino and Giaretta,1995] , les ontologies de domaine sont définies de deux manières :

11. http ://www.ontology.teknowledge.com
12. http ://www.wordnet.princeton.edu/

- Des ontologies spécifiques à un domaine particulier, contenant le vocabulaire spécifique à un domaine précis ;
- Des ontologies de tâches qui contiennent l'ensemble des tâches réalisées dans un domaine particulier. Elles décrivent un vocabulaire en relation avec une tâche ou une activité générique. Elles fournissent un ensemble de termes au moyen desquels on peut décrire, au niveau générique, comment résoudre un type de problème. D'après Mizoguchi [Mizoguchi and Bourdeau,2000], l'ontologie de tâche caractérise l'architecture computationnelle d'un système à base de connaissances qui réalise une tâche. Les ontologies de tâches peuvent dépendre des ontologies de haut niveau.

Par exemple, dans le domaine de la modélisation et du manufacturing des entreprises, on trouve l'ontologie **TOVE** (**TO**ronto **V**irtual **E**ntreprise) dans laquelle des modèles d'entreprise peuvent être représentés pour que le système réponde à certaines questions [13]. En effet, la tendance actuelle des entreprises est d'identifier, de décrire les types de problèmes, sélectionner de nouveaux processus pouvant y apporter des solutions, les évaluer, etc. Toutefois, cette tâche requiert nombre d'acteurs, à tout instant et leur coopération, à tous les niveaux de la hiérarchie. C'est pourquoi il serait avantageux de recourir à une modélisation des processus d'activités avec une représentation des processus, ressources, produits, qualités, organisation, ensuite de disposer d'un outil d'aide à la décision.

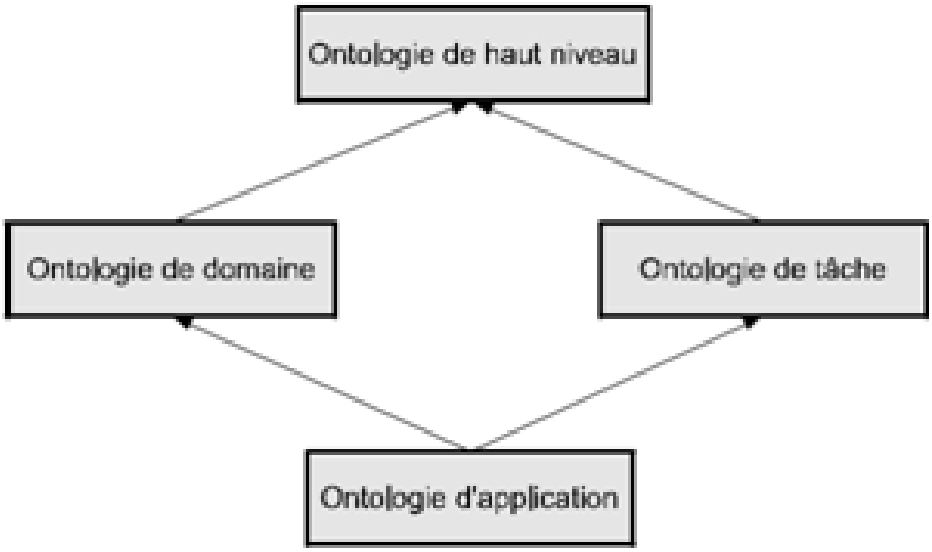

FIGURE 1.9 – *Types d'ontologie selon Guarino [Guarino and Giaretta,1995]*

Les ontologies d'applications dépendent, selon Guarino, à la fois d'un domaine particulier et d'une tâche spécifique (voir la figure 1.9).
Elles ont un domaine de validité restreint et correspondent à l'exécution d'un ensemble de tâches composant l'application. Les ontologies de cette classe précisent les connaissances nécessaires pour une application donnée, prenant en compte le

13. http ://www.eil.utoronto.ca/entreprise-modelling/tove/

vocabulaire plus spécialisé des experts d'application. Généralement, les ontologies d'application ne sont pas réutilisables et possèdent donc un intérêt plus limité.

On peut citer, par exemple, **PhysSys** qui a été construite pour assister des ingénieurs dans le développement d'applications concernant l'ingénierie de systèmes physiques dynamiques [Pim and al.,1997]. **PhysSys** exploite l'ontologie EngMath couvrant tous les aspects liés à la modélisation mathématique en ingénierie [Gruber and Olsen,1994].

D'autres exemples d'ontologies d'application sont CO et GO. **CO** (**C**himical **O**ntology) est une ontologie dans le domaine de la chimie qui permet d'identifier les groupes fonctionnels chimiques trouvés dans des inter-acteurs de petites-molécules [Feldman and al.,2005]. **GO** (**G**ene **O**ntology) est une ontologie qui vise à établir un vocabulaire structuré et contrôlé pour décrire certains domaines de la biologie moléculaire et cellulaire [14]. Dans le domaine juridique, l'ontologie LKIF Core Legal Ontology est employée pour organiser et représenter des concepts juridiques [15].

1.7.2 Typologie selon le niveau de détail

Dans cette typologie, deux types de granularité ont été distingués par Fürst [Fürst,2002]. Quand les ontologies sont très détaillées au niveau du vocabulaire utilisé, qui est plus riche, on parle de granularité fine. Ce vocabulaire doit assurer la pertinence des concepts d'une tâche spécifique, dans un domaine particulier. Souvent, les ontologies de domaine, les ontologies de tâches et les ontologies d'applications représentent des ontologies à granularité fine.

La granularité large concerne le cas où les ontologies sont moins détaillées. Un exemple est celui des ontologies de haut niveau, car elles disposent de concepts génériques qui peuvent être raffinés dans d'autres types d'ontologies (ontologie de domaine, de tâches et d'application).

1.7.3 Typologie selon le niveau de complétude

Cette typologie a été introduite par Psyché qui s'est inspiré des trois engagements considérés par Bachimont pour la définition d'ontologie. En effet, Bachimont [Bachimont,2000] a identifié trois engagements correspondant aux étapes de la modélisation des connaissances : un engagement sémantique qui fixe le sens linguistique des concepts, un engagement ontologique qui détermine le sens formel des concepts et enfin un engagement computationnel déterminant leur exploitation effective. Pour chaque engagement, un type d'ontologie est défini comme suit :
L'ontologie régionale est vue comme un arbre de concepts sémantiques. Dans [Bachimont,2000], un concept sémantique est défini par un libellé linguistique.

Celui-ci est emprunté à la langue du domaine. Son interprétation est contrainte par les principes différentiels : ceux qui lui sont directement associés et ceux de ses ancêtres dans l'arbre.

14. http ://www.geneontology.org/
15. http ://www.estrellaproject.org/lkif-core/

L'ontologie référentielle décrit un ensemble des concepts référentiels (ou formels) qui se caractérisent par un terme/libellé dont la sémantique est définie par une extension d'objets. Les concepts formels sont soit des concepts sémantiques dont on reprend le libellé et auquel on associe des référents conformément à l'engagement sémantique, soit de nouveaux concepts définis formellement par intersection de concepts formels déjà définis.

L'ontologie computationnelle traite des concepts computationnels qui sont caractérisés par les opérations qu'il est possible de leur appliquer pour générer des inférences.

1.7.4 Typologie selon le niveau du formalisme

D'autre part, Psyché a repris la classification de Uschold et Grüninger qui comprend quatre catégories [Ushold and Grüninger,1996] :

- Ontologies informelles exprimées dans un langage naturel ;
- Ontologies semi-Informelles décrites à l'aide d'un langage naturel structuré et limité ;
- Ontologies semi-formelles spécifiées dans un langage artificiel défini formellement ;
- Ontologies formelles basées sur un langage artificiel contenant une sémantique formelle, ainsi que des théorèmes et des preuves de propriétés telles la robustesse et l'exhaustivité [Gómez-Pérez and al., 1995].

Ces différentes typologies sont résumées dans la figure 1.10.

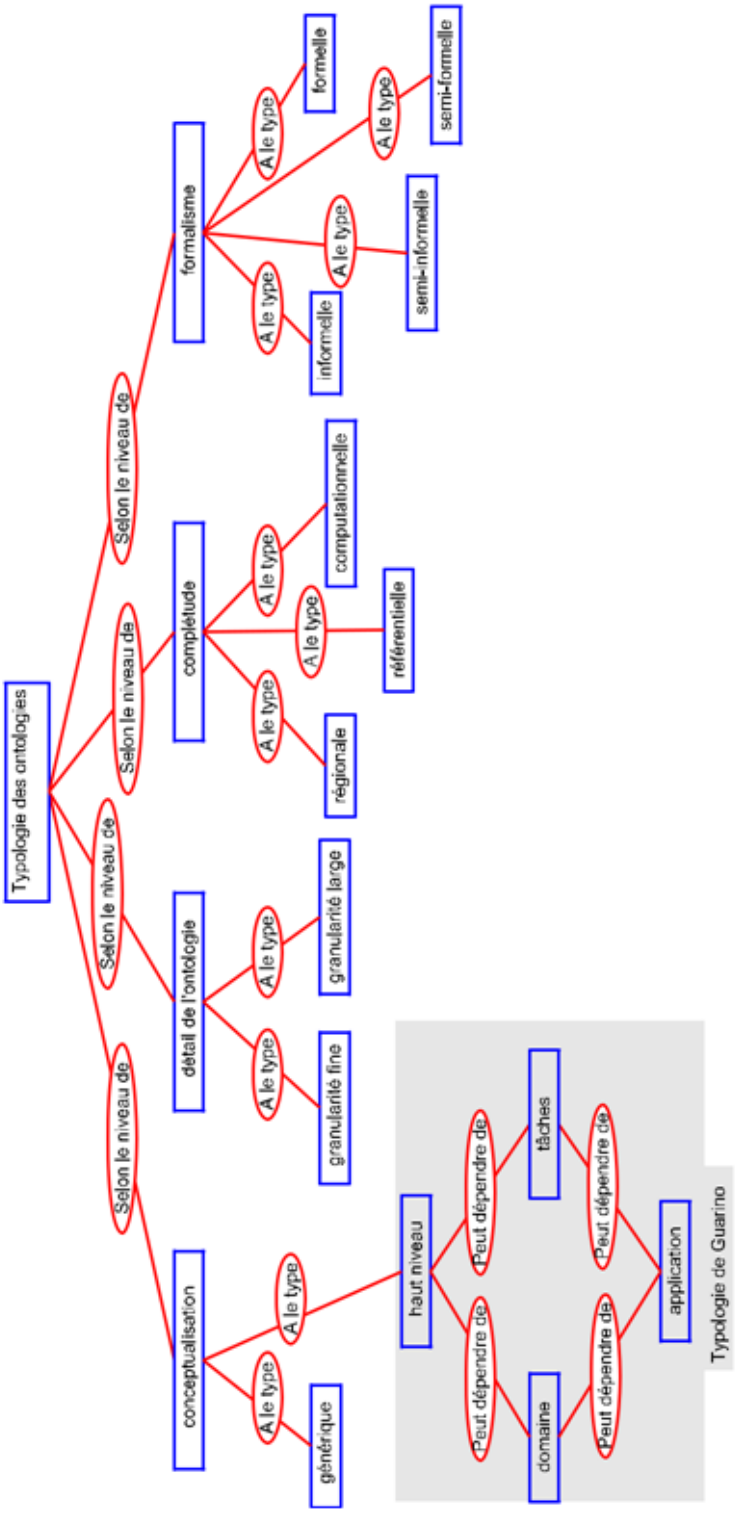

FIGURE 1.10 – *Typologies des ontologies [PSYCHÉ and al.,2003]*

1.8 Apports des ontologies

Les ontologies ont été employées dans divers domaines et pour différents objectifs. Leurs utilisations les plus répandues sont classées, selon Uschold, en trois catégories :la communication, l'interopérabilité et l'ingénierie des systèmes [Ushold and Grüninger,1996] . Elles ont également porté leurs fruits au sein des systèmes à bases de connaissances et du Web sémantique. Nous résumons les apports des ontologies comme suit :

La communication peut avoir lieu entre les hommes et/ou les systèmes. Les ontologies permettent alors le partage de la compréhension et la communication dans des contextes particuliers et selon les besoins. Ainsi, on peut utiliser l'ontologie pour créer un réseau de relations qui définit les connexions entre les composants du système. Cette caractéristique de communication est offerte grâce à la non-ambiguïté des termes utilisés et définis par l'ontologie dans les systèmes.

L'interopérabilité représente un grand défi. Elle se produit lorsque différentes organisations ont besoin de communiquer et d'échanger de l'information afin d'atteindre un objectif donné. Les ontologies contribuent à faciliter la compréhension et l'interprétation des informations échangées, en se présentant comme un format d'échange. Ce point sera plus détaillé dans le chapitre suivant.

Dans l'ingénierie des systèmes, les ontologies jouent un rôle important sur trois aspects : la spécification, la fiabilité et la réutilisation.

- Une ontologie peut aider à l'analyse des besoins et à définir les spécifications d'un système. Son rôle dépend du degré de la formalisation et l'automatisation de la méthode de spécification. Dans l'aspect informel, elles facilitent l'identification des besoins du système logiciel. Elles facilitent également la compréhension des liens et relations entre les composants de ce système. Pour l'aspect formel, elles définissent la spécification déclarative du système [16].
- Les ontologies informelles améliorent et assurent la fiabilité des systèmes logiciels en servant de base pour la vérification manuelle de la conception. Elles permettent la vérification semi-automatique du système en respectant, bien sûr, la spécification déclarative et l'intégration des différents composants du système.
- Les ontologies doivent être réutilisables. Néanmoins, un problème est rencontré quand les systèmes logiciels sont appliqués dans de nouveaux domaines : les ontologies satisfont les applications d'origine, mais non les nouvelles. Parmi les solutions proposées dans [Charlet and al.,2000], on trouve la construction d'une librairie d'ontologies réutilisables et adaptables dans différents environnements.

1.9 Conclusion

Dans ce chapitre nous avons discuté des ontologies et leur utilisation en ingénierie des connaissances. Les ontologies servent à décrire formellement des concepts

16. Une spécification déclarative fournit des caractéristique de l'ontologie indépendamment de la façon dans laquelle elle est implémentée.

et des relations entre concepts. Les concepts décrivent un système donné et participent à la signification des termes. Pour matérialiser ces concepts et relations, l'ontologie doit être construite et exprimée dans un langage. Elle doit, également, satisfaire un ensemble de propriétés consensus,(cohérence, réutilisation et partage).

Construire une ontologie est une tâche de modélisation mené à partir de l'expression linguistique des connaissances. Conformément aux étapes de construction, l'identification de l'objectif et la portée de l'ontologie viennent en premier. Elles sont suivies de l'étape d'acquisition des connaissances, du codage et de l'intégration des ontologies existantes, puis celle de la documentation et enfin, de l'étape d'évaluation.

Les ontologies peuvent être génériques, spécifiques à une tâche dans un domaine particulier ou bien directement destinées à des applications déterminées. Elles peuvent être décrites dans plusieurs langages de représentation (RDF,OWL,XML, ONTOLINGUA, SHOE, LOOM, etc).

Les apports de l'utilisation des ontologies sont divers. Les ontologies jouent un rôle important dans les systèmes à base de connaissance. Outre la réutilisation et le partage de connaissances, elles permettent de faciliter la communication entre les acteurs de différentes organisations. Elles permettent, en particulier, la réalisation de l'interopérabilité entre différents systèmes.

Jusqu'à présent, nous avons abordé la construction d'ontologie pour une application donnée.Nous allons nous intéresser à l'utilisation de l'ontologie une fois construite.

En effet, s'il est relativement facile de représenter les connaissances des systèmes utilisant les ontologies, il n'en va pas de même pour la modélisation des interactions entre ces systèmes, donc de leur interopérabilité. Quand on parle d'interaction entre systèmes, cela s'interprète par l'échange d'informations et de services. Cette interaction se retrouve actuellement dans bon nombre de systèmes distribués offrant des services aux utilisateurs. Dans ce cas, les services peuvent être considérés comme des connaissances sur ces systèmes, faisant ainsi intervenir les ontologies pour leur modélisation formelle. Ils sont définis, d'une manière générale, par les fonctionnalités, les activités et les tâches qui peuvent être effectuées par le système.

Dans le chapitre suivant, nous nous intéressons à discuter de la problématique de l'interopérabilité des systèmes d'information à base d'ontologies.

Interopérabilité des systèmes d'information à base d'ontologies 2

Sommaire

2.1 Introduction

Les systèmes ont généralement besoin de communiquer, d'échanger de l'information et des ressources sous forme de programmes, de données ou de services. Ces systèmes sont appelés systèmes distribués et l'exemple le plus connu n'est autre que le réseau Internet. Le concept de communication est à l'origine des transferts d'information dans les systèmes distribués. En termes simples, celui-ci est vu comme un ensemble de sous-systèmes logiciels et matériels, physiquement reliés par des canaux de communication. Sur cet ensemble de ressources réparties ou distribuées, les applications mises en œuvre sont appelées applications distribuées. De nombreuses applications distribuées connaissent un véritable succès, telles que le World Wide Web (WWW) sur lequel un nombre quelconque de navigateurs ou de clients peuvent se connecter.

Parmi les applications distribuées, un nombre non négligeable s'oriente vers la réalisation d'une tâche commune. Cette tâche est généralement associée à un objectif que l'on qualifie de but global du système distribué. Cet aspect est au cœur de notre problématique, puisque nous allons nous intéresser aux systèmes distribués dans lesquels chaque sous-système est dédié à la réalisation d'un fragment d'une tâche globale et possède ses propres objectifs. Dans ce domaine, un problème crucial concerne la capacité de deux applications (ou plus) à coopérer et à échanger de l'information afin d'atteindre un but global . On parle alors d'interopérabilité entre les systèmes pour la réalisation du but.

Plus précisément, la croissance exponentielle des informations et des ressources échangées entre les différents systèmes, qu'ils soient publics ou privés (Internet, bases de données, etc), augmente le taux d'hétérogénéité des informations et rend leur compréhension et leur analyse très difficiles. Un problème crucial découlant de cette hétérogénéité concerne la préservation du sens de l'information échangée. C'est ce que l'on appelle l'interopérabilité sémantique. Une définition est communément admise pour l'interopérabilité sémantique :

"Elle donne un sens aux informations échangées et s'assure que ce sens est commun dans tous les systèmes entre lesquels des échanges doivent être mis en œuvre" [Jouanot,2000] [Charlet and al.,2002] [Vernadat,2007]. La prise en compte de cette sémantique permet aux systèmes distribués de combiner les informations reçues avec des informations locales et de traiter l'ensemble de manière cohérente.

Pour assurer l'interopérabilité sémantique, l'information échangée entre systèmes doit d'abord être décrite dans une structure formelle permettant de préserver sa sémantique. Ce grand défi est omniprésent dans le domaine de l'ingénierie des connaissances, où des méthodologies et des techniques sont proposées pour percevoir, identifier, analyser, organiser et partager des connaissances entre différentes organisations. Parmi ces techniques, les ontologies

connaissent un essor très important depuis plus d'une vingtaine d'années et apparaissent comme le moyen efficace pour la représentation des connaissances. Rappelons qu'une ontologie représente la spécification d'une conceptualisation d'un domaine de connaissances [Gruber,1993].
Les ontologies ont été utilisées en tant que bases de connaissances dans l'intelligence artificielle des années 80. En ingénierie de connaissances, les ontologies n'ont pas pour objectif de comprendre le monde mais plutôt, d'en présenter les objets à des fins de manipulation informatique [Roche,2005].

L'interopérabilité sémantique requiert l'utilisation des techniques et de méthodologies qui établissent sémantiquement des liens de dépendances entre les services fournis par les entités communicantes du système distribué. Dans la littérature,la recherche de ces liens s'appelle l'alignement des ontologies. Il vise à trouver les correspondances entre les concepts appartenant à différentes ontologies au sein d'une même application. Nous montrerons au chapitre 4 que cet aspect est au cœur de notre travail.

Aussi l'objectif de ce chapitre consiste à clarifier ces différents concepts dans le cadre de l'intelligence artificielle en s'appuyant sur les diverses approches significatives (Figure 2.1).

Dans ce chapitre, nous commençons par évoquer le problème de l'interopérabilité sémantique au sein des systèmes distribués. Nous passons en revue les différentes techniques proposées pour réaliser l'interopérabilité sémantique et nous l'illustrons par des exemples d'application.

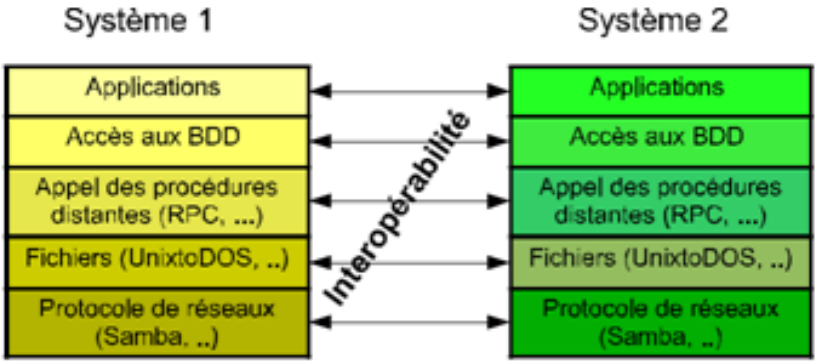

FIGURE 2.1 – *Les aspects de l'interopérabilité*

2.2 L'interopérabilité dans l'entreprise

Le besoin en interopérabilité est présenté sur quatre niveaux : données, services, processus et métier. Ce besoin est exprimé tant en intra-entreprise que pour le inter-entreprise.

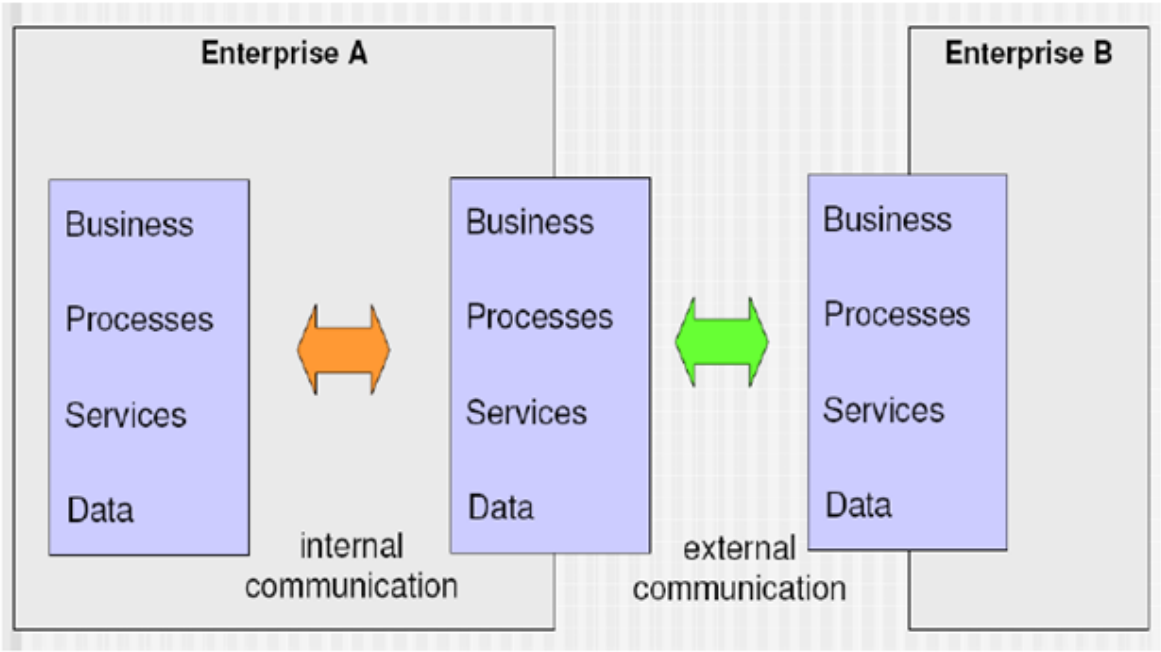

FIGURE 2.2 – *L'interopérabilité entre entreprises*

2.2.1 Niveaux d'interopérabilité dans l'entreprise

Interopérabilité au niveau des données

— Du point de vue des données, l'objectif est de faire communiquer des modèles de données différents (e.g. hiérarchiques, Relationnel, objet, etc.).
— En effet, les données sont organisées selon des schémas conceptuels différents (e.g. vocabulaires, structures de données, etc.) étroitement liés aux applications qui les supportent.
— L'interopérabilité des données revient donc à localiser et partager des informations provenant de sources hétérogènes appartenant à des bases de données différentes opérant sur des systèmes d'exploitation différents supportés par des machines différentes.

Interopérabilité au niveau des services

— Il s'agit d'identifier, composer et rassembler des fonctions de différentes applications conçues et implémentées séparément.
— Le terme « service » n'est pas limité à la notion de « service web» ou une application particulière, mais s'étend pour couvrir toutes les fonctions d'une compagnie ainsi que les entreprises en réseaux.

Interopérabilité au niveau des processus

— Un processus est définit par une séquence de services (fonctions) pour répondre à un besoin spécifique de l'entreprise.
— Généralement, ces processus évoluent en interaction (en série ou en parallèle).

Interopérabilité au niveau des métiers

— Il s'agit d'acquérir la capacité à connecter, tant en interne à l'entreprise qu'en externe avec ses partenaires, les différentes spécifications métiers.
— Cette connexion doit se faire indépendamment de la vision interne d'une entreprise, de ses modèles métiers, et de ses modes de décisions. Ceci facilite le développement et le partage des spécifications métiers entre les compagnies.

Les barrières de l'interopérabilité dans les entreprises

Les différents niveaux de l'interopérabilité sont confrontés à trois types de barrières :
— Des barrières d'ordre technologique provenant de l'utilisation de technologies différentes pour présenter, stocker, échanger, traiter et communiquer les données ;
— Des barrières d'ordre conceptuel provenant de la diversité des modes de présentation et de communication des concepts par les applications qui les utilisent. Ces applications ont été développées pour des objectifs différents ;
— Des barrières d'ordre organisationnel provenant des différents modes de travail, aspects légaux, structures organisationnelle, etc.

2.2.2 Les approches de l'interopérabilité

Il existe aujourd'hui trois approches pour réaliser l'interopérabilité entre les systèmes.
— **L'approche intégrée :** consiste à construire un format commun pour tous les modèles afin de développer un système unique. Suite à l'action d'intégration, les deux systèmes en interaction deviennent un seul avec un modèle unique.
— **L'approche unifiée :** consiste à conserver le propre modèle de chaque système en communication et définir un format commun à un méta-niveau pour faire la correspondance. Chaque système conserve alors sa propre structure avant et après la communication.
— **L'approche fédérée :** elle ne propose pas de format commun pour la communication et nécessite des efforts dynamique d'ajustement et d'accompagnement.

2.2.3 Synthèse

La figure 2.11 montre les différent niveaux de l'intéropérabilité dans l'entreprise : le besoin en interopérabilité, les barrières de l'interopérabilité, les approches de l'interopérabilité et les solutions de l'interopérabilité. Le besoin en interopérabilité est présenté sur quatre niveaux : données, services, processus et métier. Du point de vue données, l'objectif est de faire communiquer des modèles de données différents (hiérarchique, relationnel, objet, etc.) se présentant selon des schémas conceptuels différents (vocabulaires, structures et types de données, etc.) liés

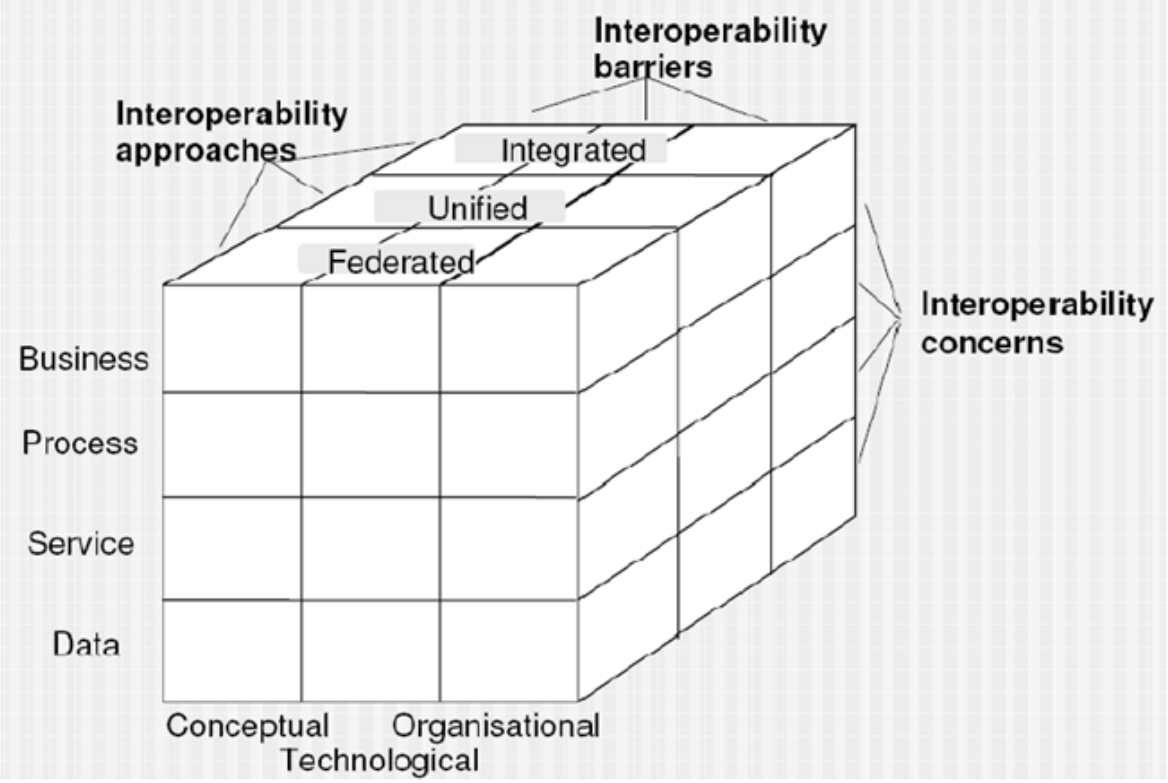

FIGURE 2.3 – *Les niveaux de l'interopérabilité dans l'entreprise*

aux applications qui les supportent. Il s'agit ici de localiser et de partager des informations provenant des sources hétérogènes appartenant à des bases de données différentes opérant sur des systèmes d'exploitation différents supportés par des machines différentes. Du point de vue des services, l'objectif est d'identifier, composer et rassembler des fonctions de différentes applications conçues et implémentées séparément. Ceci passe par la résolution des différences syntaxique et sémantique aussi bien que la connexion aux différentes sources d'information. Du point de vue des processus, dans le contexte interentreprises, l'objectif est d'étudier comment connecter des processus internes et de créer de nouveaux processus en commun.

2.3 Interopérabilité sémantique

2.3.1 Définitions

Suite au développement des systèmes informatiques une grande disparité de types de matériels et de logiciels est apparue. Ces systèmes ne peuvent dialoguer que s'ils possèdent une perception commune des informations. Cet aspect rend nécessaire la prise en compte d'un concept appelé interopérabilité. On dit que plusieurs systèmes (identiques ou différents) interopèrent s'ils peuvent communiquer sans aucune ambiguïté. Cette interopérabilité des systèmes informatiques est limitée aux aspects d'hétérogénéité et de normalisation. Elle est assurée au

niveau des couches basses des systèmes informatiques par la normalisation des interfaces physiques s'appuyant sur des standards concernant les couches hautes. A titre d'exemple, nous pouvons citer les protocoles du réseau Internet (TCP/IP, HTTP, etc) qui permettent à des ordinateurs utilisant des technologies et des systèmes d'exploitation différents d'échanger sans ambiguïté de l'information.

La notion d'interopérabilité est également présente dans les bases de données. Elle consiste à pouvoir acquérir et intégrer des informations, des données et des services issus de bases de données hétérogènes. Cette hétérogénéité survient par exemple lorsque les bases de données sont implantées par différents modèles, qu'ils soient relationnels, hiérarchiques, etc.

En ce qui concerne l'intelligence artificielle, et plus particulièrement la représentation des connaissances et le raisonnement, l'interopérabilité apparaît comme une étape cruciale vers une unification de la sémantique des connaissances distribuées. Les ontologies sont précisément un des moyens contribuant à faciliter la compréhension des informations échangées entre les systèmes interopérables en essayant de standardiser la représentation des concepts et de leurs relations.

2.3.2 Types d'interopérabilité

L'interopérabilité amène deux problèmes majeurs, les conflits syntaxiques et les conflits sémantiques. Le premier type de conflit résulte de l'utilisation de modèles de données distincts entre systèmes. Par exemple, des modèles de représentation différents sont utilisés pour structurer un même concept (relation dans le modèle relationnel,classe dans le modèle objet, XML, etc). Le second type de conflit est issu des différences de compréhension et d'interprétation entre les informations provenant de divers domaines d'application [Jouanot,2000] .

Ces deux types de conflits, nécessitent de caractériser l'interopérabilité syntaxique et l'interopérabilité sémantique.Dans le premier type, la syntaxe des informations échangées est définie et doit être respectée par les différents systèmes qui interopèrent. Dans le second type, l'interopérabilité vise à explorer les problèmes de cohérence dans le sens où les informations échangées doivent avoir la même signification au sein de ces systèmes. L'interopérabilité sémantique doit également préserver la sémantique des informations échangées.

Nous nous intéressons ici à l'interopérabilité sémantique, qui représente actuellement un défi dans plusieurs domaines de recherche, en particulier en intelligence artificielle à travers la notion d'ontologie.

F.Vernadat explique l'interopérabilité sémantique en une phrase : " To exchange services and data among systems that make sense (common "meaning")" [Vernadat,2007]. Par conséquent, l'interopérabilité sémantique consiste à donner

un sens aux informations échangées et à garantir que ce sens est distribué dans tous les systèmes entre lesquels des échanges doivent être mis en œuvre. La prise en compte de cette sémantique permet à ces systèmes de combiner les informations reçues avec des informations locales et de les traiter d'une manière appropriée dans le respect de cette sémantique.

2.3.3 Comment assurer l'interopérabilité sémantique ?

L'interopérabilité sémantique pose un problème de compréhension des informations échangées entre des systèmes coopérant à la réalisation d'une tâche globale. Pour remédier à ce problème, les chercheurs se sont orientés vers deux solutions complémentaires.

Les informations échangées doivent être structurées de manière à faciliter leur compréhension. Cette structuration amène à utiliser des ontologies qui décrivent, dans un cadre formel, les connaissances d'un domaine. La représentation par l'ontologie des buts du système appartient à la classe des représentations dites semi-formelles. Grâce à ce modèle formalisant les buts des systèmes, la réalisation de l'interopérabilité sémantique apporte une simplification notable à la modélisation.

Lorsque'un système reçoit des informations, la compréhension de celles-ci nécessite la mise en correspondances avec d'autres informations ou concepts connus au sein du système, afin de les exploiter. Cette mise en correspondance a donné lieu à la proposition et au développement de nouvelles techniques, telles que l'alignement qui consiste en la découverte de correspondances entre informations.
Nous présentons maintenant ces deux solutions plus en détails.

2.4 Techniques pour l'interopérabilité sémantique

Un certain nombre de techniques ont été proposées dans la littérature pour réaliser l'interopérabilité. Elles sont souvent utilisées pour permettre le partage des données entre des bases de connaissance hétérogènes et pour la réutilisation des formations de ces bases.

Dans l'ouvrage "Semantic Web Technologies" [Bruijn and al.,2006], l'auteur distingue trois catégories principales qui sont :

(a) Le mapping d'ontologies [1],qui a comme objectif la représentation des correspondances entre les ontologies. Ceci permet, par exemple, d'interroger

1. On emploie le mot anglais Mapping tout au long du manuscrit plutôt que le mot appariement dans la langue française.

des bases de connaissances hétérogènes en utilisant une interface commune ou en transformant des données entre différentes représentations.

(b) La fusion d'ontologies, qui permet de créer une nouvelle ontologie, appelée l'ontologie fusionnée capturant les connaissances des ontologies d'origine. 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.

(c) L'alignement d'ontologies, pour qui l'objectif consiste à découvrir des correspondances entre les ontologies.

Ces trois techniques seront détaillées dans les sections suivantes.

2.4.1 Le mapping d'ontologies

Le mapping d'ontologies constitue un problème récurrent dans plusieurs approches. La définition la plus pertinente est probablement celle de Noy pour qui le mapping d'ontologies est un processus qui spécifie une convergence sémantique entre différentes ontologies afin d'en extraire les correspondances entre certaines entités [Noy,2004].

Ces correspondances sont exprimées en introduisant des axiomes formulés dans un langage spécifique.

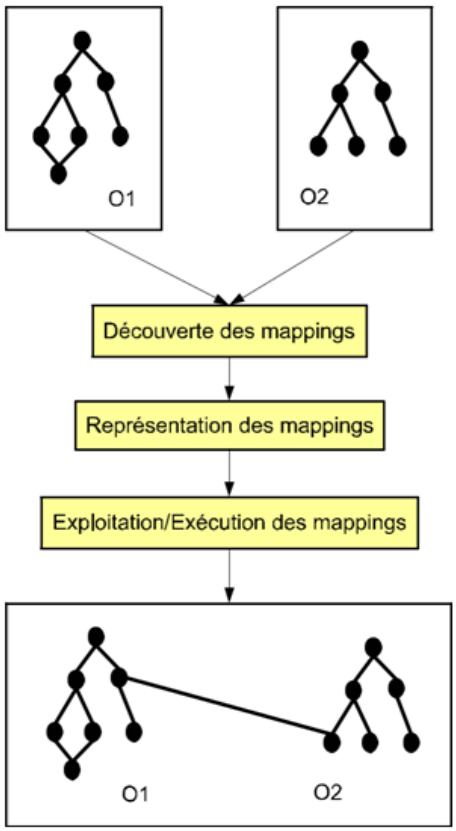

FIGURE 2.4 – *Le mapping des ontologies*

Trois phases principales peuvent être distinguées dans ce processus (voir la figure 2.2) :
— la découverte du mapping ;
— la représentation du mapping ;
— l'exploitation et l'exécution du mapping.
Les outils et méthodologies, parmi les plus significatifs dans cette catégorie, sont MAFRA [Maedche and al.,2002], IF-Map [Kalfoglu and Shorlemmer,2003], RDFT [Omelayenko,2003] , C-OWL [Bouquet and al.,2004] et OntoMap [Angele and Shnurr,2005] .

MAFRA

MAFRA (MApping FRAmework) est une plate-forme pour le mapping d'ontologies distribuées. Son approche repose sur la notion fondamentale de Passerelle qui permet de formaliser les mappings en établissant des correspondances entre les entités d'une ontologie source et celles d'une ontologie cible.
L'architecture conceptuelle de MAFRA distingue la dimension horizontale et la dimension verticale [Maedche and al.,2002](voir la figure 2.3).

La dimension horizontale est caractérisé par cinq modules. Le module de normalisation où les ontologies sont représentées de manière uniforme. Le module de

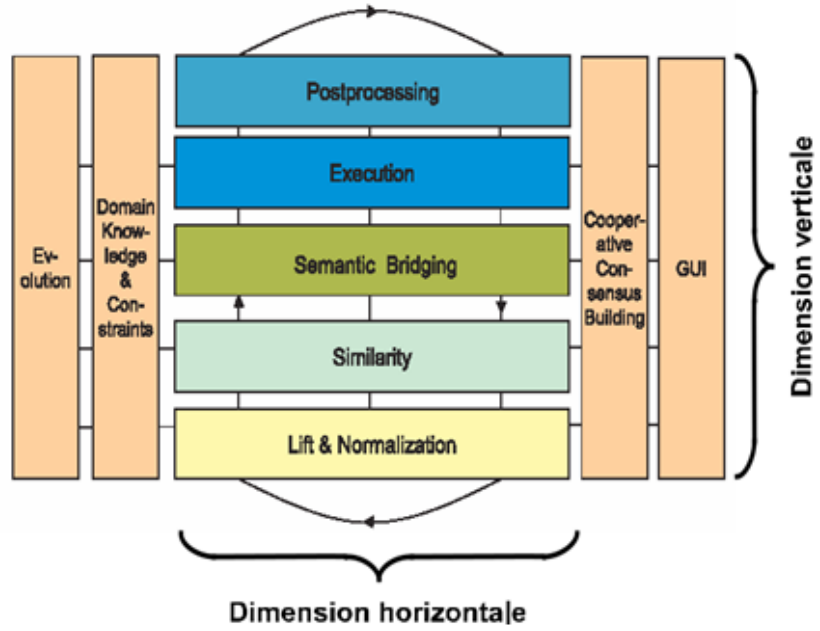

FIGURE 2.5 – *L'architecture conceptuelle de MAFRA selon [Maedche and al.,2002]*

similarités établit les ressemblances entre l'ontologie source et l'ontologie cible. Le module du " bridging " sémantique, basé sur les similarités, vise à établir les correspondance entre les entités appartenant aux ontologies sources et cibles en utilisant des heuristiques. Ce module spécifie des passerelles entre entités pour lesquelles chaque instance représentée dans l'ontologie source est transformée en l'instance la plus similaire dans l'ontologie cible. Ce module prend en compte les entités (concepts, relations et instances), la cardinalité $(1:1, 1:n$ et $n:1)$, la structure (spécialisation, alternative, composition, abstraction), les contraintes et les transformations.

Le module Exécution transforme les instances à partir de l'ontologie source vers l'ontologie cible en évaluant les passerelles sémantiques. Enfin, le module Post-Processing récupère le résultat d'exécution afin de vérifier et d'améliorer la qualité des résultats de transformation.

La dimension verticale est également divisée en sous modules. Le module d'Évolution vise à préserver les passerelles sémantiques générées dans le module bridging sémantique. Celui de la construction d'un consensus coopératif consiste à établir un consensus sur les passerelles sémantiques entre deux communautés participant au processus du mapping. Le module Domaine de connaissances et contraintes est utilisé pour améliorer les passerelles sémantiques. Enfin, le module Interface graphique pour l'utilisateur (GUI) est toujours recommandé pour masquer la difficulté des conceptions complexes.

En résumé, l'établissement des mappings dans la plateforme MAFRA repose

essentiellement sur la génération de passerelles, lesquelles établissent des correspondances entre les entités de différentes ontologies.

IF-Map

IF-Map (Information Flow Mapping) est une méthodologie proposée par Schorlemmer et Kalfoglou réalisant des mappings entre différentes ontologies de domaine [Kalfoglu and Shorlemmer,2003] . IF-Map repose sur la théorie de Barwise "The logic of Information Flow ", noté IF Model [Barwise and Seligman,1997] .

Les auteurs étudient la théorie des flux d'information qui fournit une description théorique consistante du processus partiel d'intégration sémantique effectué par deux agents. Pour cela ceux-ci réalisent un alignement progressif des ontologies à l'aide des instances de celles-ci. Leur approche est basée sur une mise en correspondance d'ontologies locales. Deux autres ontologies sont utilisées dans le processus de mapping : l'une que l'on appelle ontologie de référence et l'autre, ontologie globale " (voir la figure 2.4). Ils supposent que les ontologies locales sont utilisées par différentes communautés et sont peuplées par des instances de leur domaine respectif, alors que l'ontologie de référence est une interprétation commune dédiée au partage de la connaissance mais privée d'instances. L'ontologie globale est une ontologie virtuelle créée au vol pour les besoins de la fusion. Les relations entre les ontologies locales et l'ontologie virtuelle sont du type infomorphismes logiques.

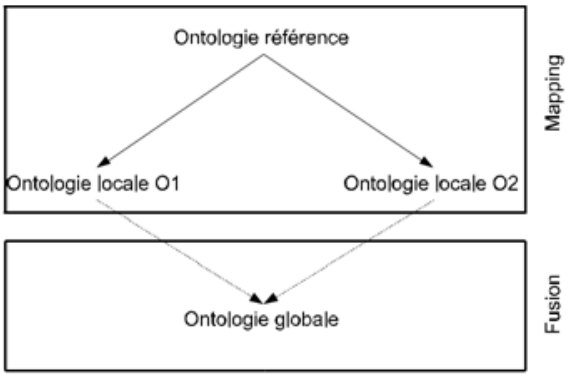

FIGURE 2.6 – *Le mapping entre deux ontologies locales [Kalfoglu and Shorlemmer,2003].*

Basées sur le modèle IF, les ontologies sont exprimées en termes de logiques locales [2]. Les mappings sont alors formalisés par les infomorphismes [3] (voir le chapitre 4 pour plus de détails). Le processus de mapping de l'approche IF-Map [Kalfoglu and Shorlemmer,2003] inclut quatre étapes principales représentées à la figure 2.5.

— L'acquisition des ontologies (Ontology Harvesting) qui utilise des ontologies existantes, en les téléchargeant à partir du Web, ou en les créant.
— La traduction d'un ensemble d'ontologies d'entrée qui peut se faire en clauses de Horn puisque IF-Map est spécifiée dans cette logique. Par exemple, une ontologie générée en RDF [4] peut être traduite en Prolog [5].
— La génération des infomorphismes(Infomorphisms Generation) qui construit des liens formels entre les éléments similaires de deux différentes ontologies.
— L'affichage des mappings (Display Mappings) qui traduit les infomorphismes en syntaxe RDF et les stocke dans une base de connaissances pour une utilisation ultérieure par d'autres applications.

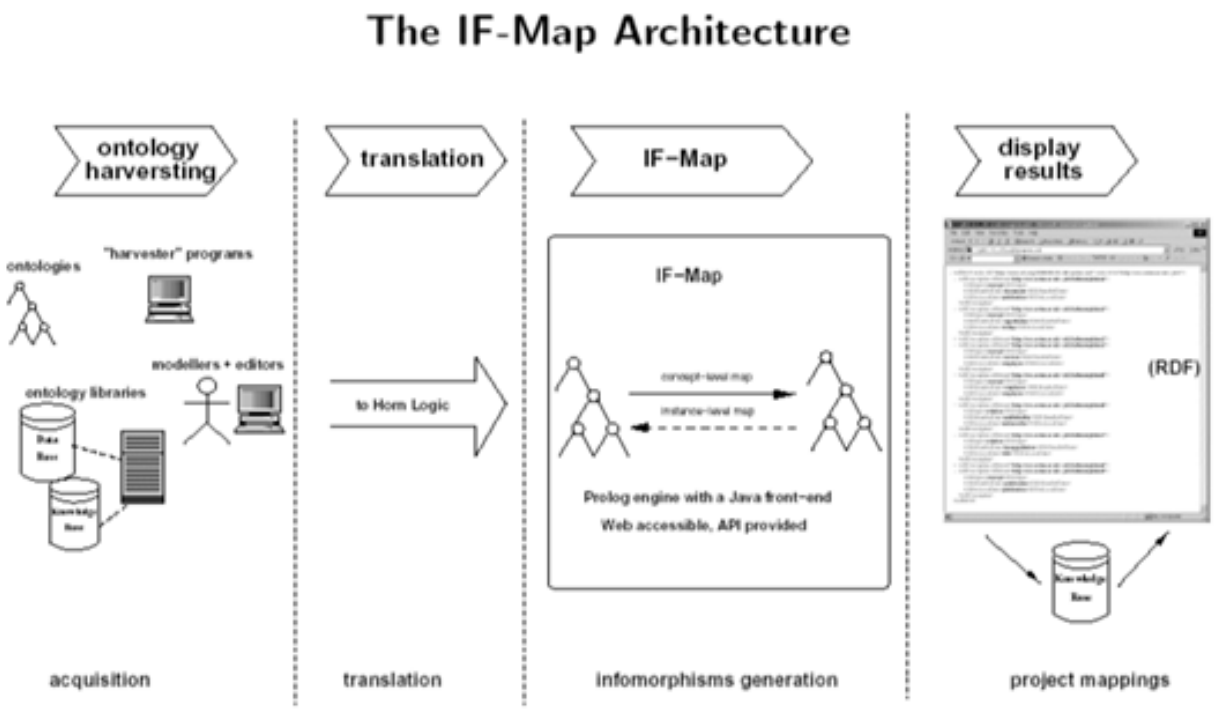

FIGURE 2.7 – *L'architecture de IF-Map [Kalfoglu and Shorlemmer,2003]*

2. Une logique locale dans le modèle IF vise à formaliser les régularités au sein des systèmes.
3. En termes simples, un infomorphisme est une fonction qui modélise les connexions entre systèmes.
4. RDF : Resource Description Framework.
5. PROgrammation LOGique est l'un des principaux langages de programmation logique.

RDFT

Omelayenko a proposé RDFT (Resource Description Framework Transformation) pour résoudre les mappings entre méta-ontologies [Omelayenko,2003] en spécifiant un langage entre des ontologies exprimées au moyen de DTDs XML et de schémas RDF.

RDFT dispose d'une classe de base appelée "BRIDGE " ou " Passerelle ". Cette classe vise à connecter des concepts et à spécifier leurs correspondances selon le type de passerelle. Comme le montre la figure 2.6, deux types de passerelles sont requis , les passerelles Un-à-plusieurs (one-to-many) qui détaillent les correspondances entre une seule entité et un ensemble d'entités et les passerelles Plusieurs-à-un (many-to-one) dédiées aux correspondances inverses.

RDFT se rapproche de MAFRA en ce qui concerne l'idée d'utiliser les passerelles pour le mapping d'ontologies.
La relation entre les composants source et cible de la passerelle peut être une relation d'équivalence ou encore une relation de version qui déclare que les éléments de l'ensemble cible forment une version postérieure des éléments de l'ensemble source dans l'hypothèse où les domaines sont identiques pour les deux ensembles.

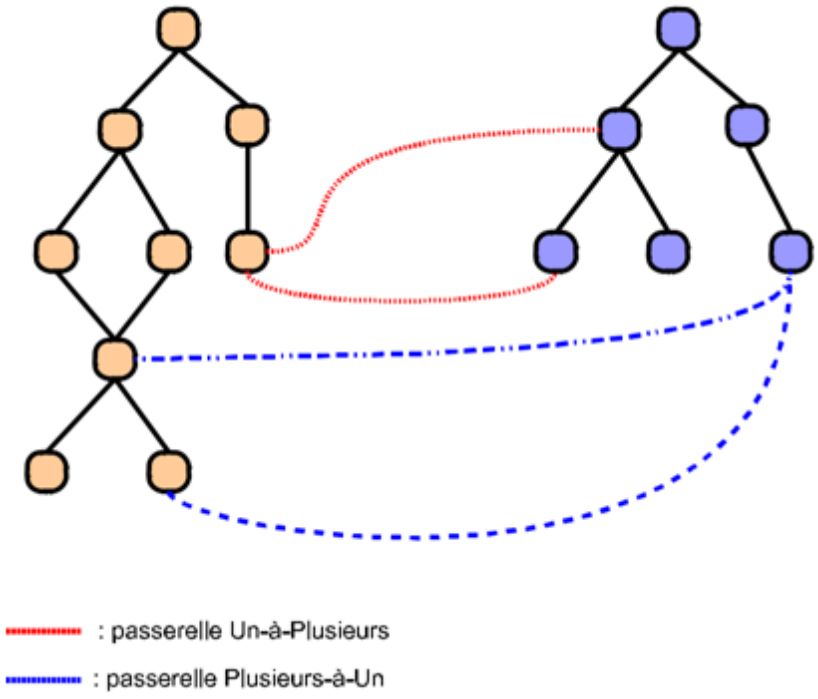

FIGURE 2.8 – *Les passerelles*

C-OWL

C-OWL " Contextual Ontology Web Language " est une extension de OWL [6] qui a été proposée pour la représentation d'ontologies contextualisées [Bouquet and al.,2004] . Ces dernières sont des représentations locales, appelées contextes. Les liens entre les contextes sont représentés en relation avec d'autres contextes au travers des mappings. Deux notions principales caractérisent l'approche C-OWL : l'espace de contextes et les passerelles.

Les connaissances sont contenues dans un ensemble de contextes, appelé espace de contextes où chaque contexte est exprimé par une ontologie décrite en OWL possédant son propre langage et sa propre interprétation.

Les mappings entre ontologies s'expriment sous la forme de passerelles. Une passerelle entre un contexte, représenté par une ontologie OWLOi, et un autre contexte, représenté par une autre ontologie OWL O_j , permet de déclarer une correspondance entre les éléments de connaissances de ces deux contextes. Sur la base de ces correspondances, une partie des connaissances contenues dans O_i peut être interprétée et réutilisée dans O_j

Selon les auteurs [Bouquet and al.,2004] , C-OWL permet l'exploitation de mécanismes de raisonnements globaux, s'appuyant sur les passerelles. Ainsi, le calcul de la subsomption globale repose sur ce qui est appelé le principe de propagation de la subsomption :la subsomption dans un contexte O_i peut être déduite de la subsomption dans un contexte O_j et de passerelles de O_i à O_j . L'instanciation globale repose sur le principe de propagation de l'instanciation, fortement inspiré du principe de propagation de la subsomption. Il est possible d'en déduire qu'un individu est une instance d'une classe dans un contexte en s'appuyant sur l'instanciation dans un autre contexte et sur les passerelles. Un mapping en C-OWL est défini par un ensemble de passerelles entre deux ontologies. L'ensemble des ontologies OWL et de leurs mappings constituent l'espace de contextes.

OntoMap

OntoMap représente un "plugin" dans la plate-forme OntoStudio [7]. Il permet la création et la gestion des mappings d'ontologies. Ceux-ci sont accessibles par une une représentation graphique et avec l'environnement Schema-View dédié à l'ontologie [Angele and Shnurr,2005] .

6. Ontology Web Language
7. http : //www.onto prise.de/content/e1171/e1249/indexing.html

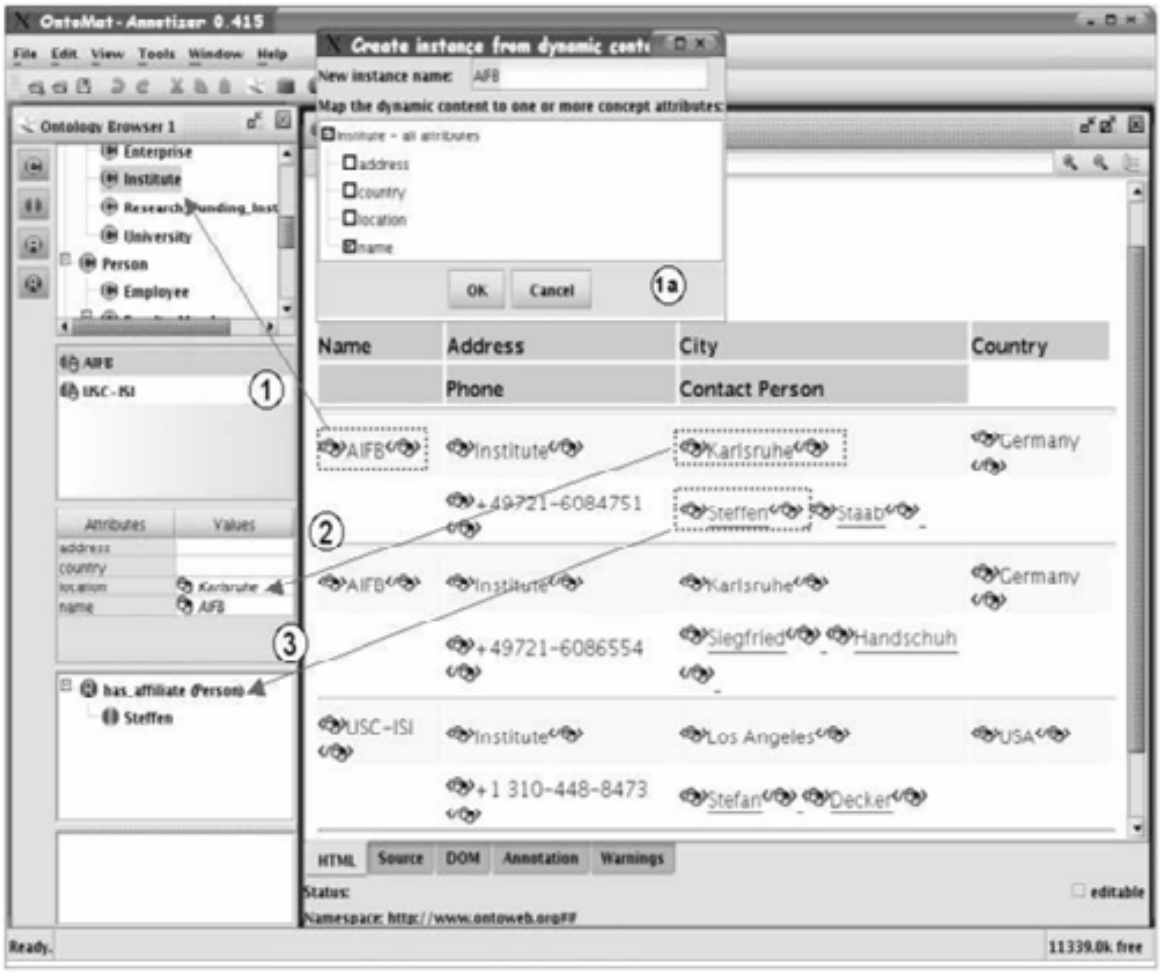

FIGURE 2.9 – *L'outil OntoMap*

OntoMap associe des phrases formelles à la déclaration des mappings limitant ainsi la tâche des utilisateurs à la compréhension de la sémantique de représentations graphiques (par exemple, une flèche qui connecte deux concepts). Les utilisateurs de OntoMap disposent également de la fonctionnalité "Drag and Drop" et peuvent aussi vérifier la consistance des propriétés.

OntoMap supporte un ensemble de patterns de mapping tels que concept-à-concept, attribut-à-attribut, relation-à-relation, et attribut-à-concept . Dans la figure 2.7, nous donnons un bref aperçu de cet outil [Angele and Shnurr,2005] .

2.4.2 La fusion d'ontologies

[Namyoun and al.,2006] donne la définition de la fusion d'ontologies suivante : "Ontology merging is the process of generating a single, coherent ontology from two or more existing and different ontologies related to the same subject".

La fusion d'ontologies représente la création d'une nouvelle ontologie à partir de deux ontologies ou plus. L'ontologie résultante unifie et remplace les on-

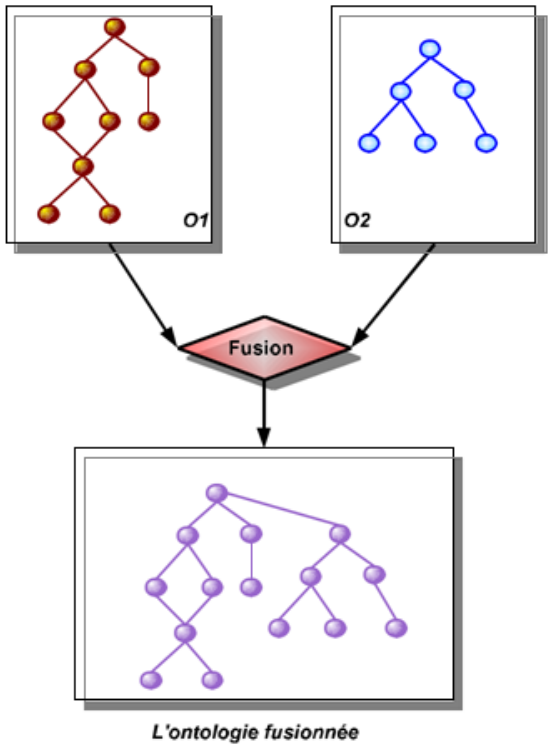

FIGURE 2.10 – *Le principe de la fusion d'ontologies*

tologies d'origine (Voir la figure 2.8). Cette définition ne précise pas comment l'ontologie résultante est reliée aux ontologies originales pour laisser ouvert le problème du choix de la méthode de fusion. 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 originelles.

Plusieurs approches mettant en œuvre la fusion d'ontologies ont été proposées telles que PROMPT [Noy and Musen,2000b], CHIMAERA [McGuinness and al.,2000], FCA-Merge [Stumme and Maedche,2001] et OntoMerge [Dou and al.,2002].

PROMPT

PROMPT est un outil dont le processus de fusion est de type interactif. L'ensemble de phases associé à ce processus comprend les étapes suivantes [Noy and Musen,2000b] :

— Les candidats à la fusion sont identifiés à partir des similarités des noms de

classes. Le résultat est présenté à l'utilisateur comme une liste d'opérations potentielles de fusion.
— L'utilisateur choisit une des opérations suggérées par la liste ou spécifie directement l'opération de fusion.
— Le système effectue l'action demandée et exécute automatiquement les changements additionnels dérivés de cette action.
— Le système crée une nouvelle liste d'actions suggérées par'utilisateur en se basant sur la nouvelle structure de l'ontologie. Il détermine les conflits présentés par la dernière action, les solutions possibles à ces conflits puis présente ces derniers à l'utilisateur.

PROMPT identifie un ensemble d'opérations pour la fusion d'ontologies (fusion des classes, fusion de slots, fusion des liens, etc.) et un ensemble de conflits possibles consécutifs à l'application de ces opérations (conflits de nom, redondance dans la hiérarchie des classes).

CHIMAERA

CHIMAERA est un environnement pour la fusion d'ontologies [McGuinness and al.,2000] qui aide les utilisateurs à créer et à maintenir des ontologies distribuées dans le Web. Il fournit des outils de diagnostic et supporte deux fonctions principales :
— Fusionner plusieurs ontologies.
— Analyser des ontologies individuelles ou multiples.

La fusion est exprimée par un opérateur entre les paires de termes, de noms et de définitions considérées comme candidats à la fusion. CHIMAERA dispose également de techniques permettant de lier les termes par des relations de subsomption, disjonction, etc. L'analyse effectuée par Chimaera comprend aussi bien une vérification de la rigueur logique d'une ontologie que le diagnostic des erreurs habituelles dans sa conception.

FCA-Merge

Stumme et Maedche [Stumme and Maedche,2001] ont proposé FCA-Merge (Formal Conceptual Analysis Merge) dans le but de fusionner des ontologies locales qui partagent le même ensemble d'instances. Pour cela, les auteurs exploitent l'analyse formelle des concepts. Le processus de fusion nécessite trois étapes :
— L'extraction des instances à partir de documents de type texte.
— La génération du treillis de concepts en appliquant l'analyse formelle des concepts aux instances. Chaque nœud du treillis est associé à un ensemble de concepts des ontologies locales lorsque les instances associées sont contenues dans les mêmes documents.
— La génération interactive de l'ontologie fusionnée est l'étape finale de l'analyse du treillis qui construit l'ontologie globale. Cette étape est à la charge du concepteur.

OntoMerge OntoMerge est un outil qui facilite la création d'une ontologie appelée Ontologie Passerelle. Celle-ci importe les ontologies originelles et relie les concepts grâce à un certain nombre d'axiomes [Dou and al.,2002].

OntoMerge est une approche en ligne dans laquelle les ontologies source sont maintenues après l'opération de fusion, alors que dans PROMPT l'ontologie fusionnée remplace les ontologies source. Le résultat de l'opération de fusion dans OntoMergen'est pas une ontologie complètement fusionnée, comme dans PROMPT, mais une ontologie passerelle qui importe des ontologies source. Des règles de traduction issues des axiomes relient la partie de convergence des ontologies source. Les ontologies source sont traitées avec les axiomes passerelle comme une seule théorie par un démonstrateur de théorèmes optimisé pour trois opérations principales :

— La traduction de l'ensemble de données d'une représentation à une autre.
— La génération d'extensions d'ontologie qui, étant données deux ontologies reliées O_1 et O_2, engendre une extension O_{2s} de O_2 ainsi qu'une extension O_{1s} de O_1.
— L'interrogation de différentes ontologies.

2.4.3 L'alignement des ontologies

L'alignement des ontologies permet d'établir des liens sémantiques entre les concepts et des relations inter-ontologies, comme l'illustre la définition de Namyoun [Namyoun and al.,2006] : "Ontology alignment is the task of creating links between two original ontologies. Ontology alignment is made if the sources become consistent with each other but are kept separate. Ontology alignment is made when they usually have complementary domains" .

L'alignement d'ontologies est un processus de découverte des correspondances entre deux ontologies source. Il est, généralement, décrit comme une application de l'opérateur MATCH [Rahm and Bernstein, 2001] , dont l'entrée est constituée d'un ensemble d'ontologies et la sortie, formée des correspondances entre ces ontologies (voir la figure 2.9).

On trouve dans la littérature plusieurs algorithmes qui implémentent cet opérateur. Ces algorithmes sont généralement regroupés en quatre classes. On trouve le matching basé-schéma,le matching basé-instance, le matching niveau-élément et le matching niveau-structure. Un matcher basé-schéma prend en compte différents aspects des concepts et des relations dans les ontologies et utilise des mesures de similarité pour déterminer la correspondance.

Les instances appartenant aux concepts des différentes ontologies sont comparées pour découvrir les similarités entre les concepts. Dans le matching niveau-élément et le matching niveau-structure, le " matcher niveau-élément " compare les propriétés d'un concept particulier et d'une relation particulière (par exemple pour trouver des similarités à partir du nom). Le " matcher niveau-structure " compare la structure, c'est-à-dire la hiérarchie du concept pour trouver les similarités [Noy and Musen,2000a] [Giunchiglia and al.,2005].

Selon Ehrig, ces " matchers " peuvent également être combinés [Ehrig and Staab,2004].

Parmi les approches d'alignement d'ontologies, on trouve Anchor-PROMPT

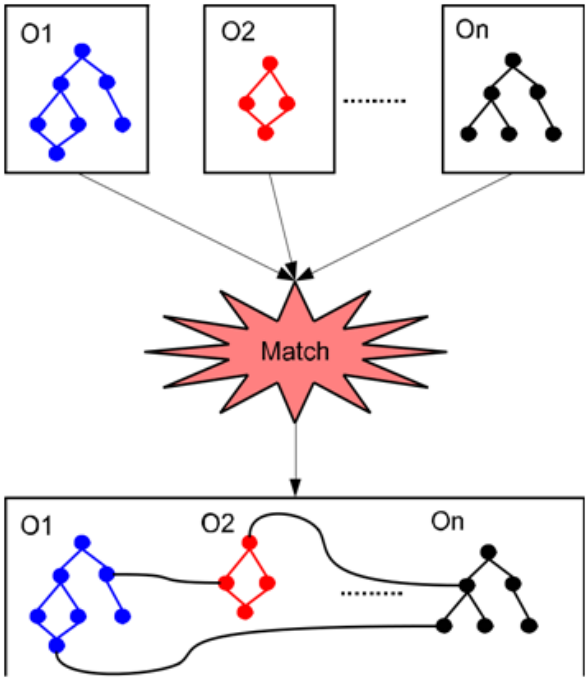

FIGURE 2.11 – *L'opérateur MATCH*

[Ehrig and Staab,2004], GLUE [Doan and al., 2003], [Giunchiglia and al.,2005], QOM [Ehrig and Staab,2004] et ASCO [Thanh LE and al.,2004] :

Anchor-PROMPT

Anchor-PROMPT est un algorithme qui a été développé par Noy et Musen [Noy and Musen,2000a] afin de découvrir automatiquement les termes sémantiquement similaires. Anchor-PROMPT traite une ontologie comme un graphe, les nœuds de ce graphe représentant les classes et les arcs représentant les propriétés.

L'algorithme utilise deux paires de termes relatifs comme entrée. Il analyse les chemins dans le sous-graphe délimité par des ancres et détermine quelles sont les classes qui apparaissent fréquemment dans des positions similaires sur des chemins similaires.

L'algorithme cherche alors des termes le long des chemins qui pourraient être similaires aux termes d'autres chemins.

Ces nouveaux termes relatifs sont identifiés par une similarité qui peut être modifiée pendant l'évaluation d'autres chemins dans lesquels ces termes apparaissent. Les termes qui sont fortement semblables sont présentés à l'utilisateur pour améliorer l'ensemble des suggestions possibles.

GLUE

GLUE est un système qui emploie une approche de type " machine learning " pour créer des mappings semi-automatiques entre ontologies hétérogènes.Il se base sur des données d'instances, une ontologie étant vue comme une taxinomie de concepts [Doan and al., 2003]. GLUE se focalise sur la détermination des mapping de type 1-à-1. La similarité de deux concepts A et B dans deux taxinomies $O1$ et $O2$ utilise l'ensemble des instances des deux concepts qui convergent.

Pour déterminer si une instance du concept B est également une instance du concept A, un classifieur est d'abord construit en utilisant les instances de A comme ensemble d'apprentissage. Ce classifieur est ensuite utilisé à son tour pour traiter les instances de B. Le classifieur décide alors pour chaque instance de B, s'il est également une instance de A ou non. Avec ces classifications, quatre probabilités sont calculées : $P(A,B), P(\overline{A},B), P(A,\overline{B})$ et $P(\overline{A},\overline{B})$. La probabilité $P(\overline{A},\overline{B})$ par exemple, s'interprète par l'appartenance de l'instance du domaine à A et la non appartenance à B. Ces quatre cas peuvent ensuite être employées comme paramètres pour calculer la distribution de probabilité commune pour les concepts A et B, laquelle est une fonction écrite par l'utilisateur.

S-Match

S-Match est une approche pour le matching des hiérarchies de classifications [Giunchiglia and al.,2005]. Les auteurs mettent en oeuvre un opérateur de correspondance qui admet en entrée deux structures sous forme de graphes (par exemple, schémas de base de données ou ontologies) et produit un mapping entre les éléments qui sont en correspondance sémantique.

Selon Giunchiglia et Shvaiko, presque toutes les anciennes approches utilisant les schémas et le matching d'ontologies sont des approches de matching syntaxiques par opposition au matching sémantique. Dans le matching syntaxique, les labels et parfois la structure syntaxique du graphe sont associés. Un certain coefficient de similarité exprimé dans l'intervalle [0.1] est obtenu. Il indique la similarité entre deux nœuds du graphe.
Le matching sémantique calcule une relation, de nature ensembliste, entre les nœuds en tenant compte de la signification de chaque noeud. La sémantique d'un nœud est déterminée par son label et celle de tous les nœuds qui sont plus haut dans la hiérarchie. Les relations possibles retournées par l'algorithme du matching sémantique sont l'égalité, l'intersection, la disparité, la généralité ou la spécificité.
Dans ce cas, le problème du matching est vu comme un problème de satisfaction d'un ensemble de formules du calcul propositionnel. Les graphes et les correspondances à tester sont traduits en formules de la logique propositionnelle en considérant non seulement leur nom, mais également la position des concepts dans le graphe.

QOM

QOM (Quick Ontology Mapping) est une approche qui a été conçue pour fournir un outil efficace de matching pour la création au vol des alignements entre les ontologies [Ehrig and Staab,2004].
Afin d'accélérer l'identification des similarités entre deux ontologies, QOM ne compare pas celles de la première ontologie avec toutes les entités de la seconde ontologie, mais emploie des heuristiques (par exemple, labels semblables) pour abaisser le nombre de mappings candidats. Le calcul réel de similarité est effectué en utilisant une large gamme de fonctions de similarité, telles que la similarité des strings.
Plusieurs mesures de similarité sont calculées et servent d'entrée à une fonction d'agrégation. QOM applique une fonction sigmoïde qui fait ressortir différentes similarités élevées et basses. Les correspondances réelles entre les entités des ontologies sont extraites en appliquant un seuil de mesure agrégée de similarité. La sortie d'une itération peut être utilisée en tant qu'élément d'entrée pour l'itération suivante afin de raffiner le résultat. Après un certain nombre d'itérations, une table de correspondances entre les ontologies est obtenue.

ASCO

ASCO est un algorithme qui permet de comparer deux ontologies [Thanh LE and al.,2004]. Il trouve des mappings en suivant un processus à deux phases.

- La phase linguistique dans laquelle la valeur de similarité entre deux entités, telles que des concepts ou des relations provenant de deux ontologies différentes, est calculée à partir de différentes informations disponibles telles que leurs noms, leurs étiquettes (des labels qui fournissent une version compréhensible par un humain du nom du concept ou de la relation) et leurs descriptions. Le calcul de la valeur de similarité linguistique est effectué de plusieurs manières. Pour améliorer la précision du calcul et pour exploiter les relations de synonymie ou hyperonymie entre termes, ASCO intègre WordNet.
- La phase structurelle exploite les informations taxonomiques dans les structures des ontologies. Elle utilise des heuristiques et les connaissances du domaine pour calculer les valeurs de similarité structurelle entre entités des deux ontologies. Les valeurs de similarité dans les deux phases sont combinées pour obtenir les valeurs de similarité finales entre les entités. Les alignements sont déduits de ces valeurs.

2.5 Conclusion

La majorité des approches citées dans les différentes catégories (mapping, fusion et alignement) exploitent des mécanismes qui ne reposent pas sur des fondements théoriques et/ou modèles mathématiques robustes. La plupart de ces mécanismes utilisent des heuristiques, la logique propositionnelle ou les probabilités. Ces approches reposent sur l'utilisation des calculs de similarités syntaxiques afin d'identifier les correspondances entre concepts, mais prennent rarement en considération leur sémantique.

Les techniques citées précédemment s'accordent sur certaines propriétés mais elles présentent certaines limites :

(a) La plupart de ces approches, telles que MAFRA, RDFT etc, sont limitées à l'utilisation d'algorithmes semi-automatiques pour le mapping, la fusion et l'alignement d'ontologies.

(b) La prolifération des concepts introduits dans les ontologies engendre souvent une explosion combinatoire.

(c) Pour lier les concepts de différentes ontologies, ces approches s'appuient sur des similarités syntaxiques entre ces concepts. Toutefois, deux concepts peuvent avoir une même syntaxe alors que leur sémantique est différente car placée dans des contextes différents.

(d) A l'inverse, deux concepts peuvent avoir la même sémantique alors qu'ils sont décrits par différentes syntaxes.

(e) Enfin, ces méthodologies sont difficilement compréhensibles par les utilisateurs à cause de leur complexité.

Nous proposons dans le chapitre suivant, d'analyser en profondeur les stratégies actuelles : le partitionnement et la modularisation utilisées par différents systèmes d'alignement, ainsi que les problèmes techniques rencontrés lors de leur exploitation .

Stratégies d'alignement à base d'ontologies

3

Sommaire

3.1 Introduction

Le matching ou alignement d'ontologies (fig.3.1) permet de trouver les correspondances entre des entités d'ontologies reliées sémantiquement. Ces correspondances peuvent être utilisées pour différentes applications telles que la fusion d'ontologies, la transformation des données ou pour le web sémantique. Plusieurs techniques d'alignement, basées sur différents critères, sont actuellement proposées dans la littérature. [Ardjani and al.,2015] fournit une synthèse des techniques d'alignement. Le choix d'une technique ou d'un procédé ou la composition de plusieurs d'entre eux n'est pas une tâche facile. De nombreuses méthodes d'alignement dédiées aux ontologies ont vu le jour au cours de la dernière décennie. Cependant, ces méthodes sont conçues pour aligner de petites ontologies. Plusieurs approches ont été proposées par [Hamdi and al., 2009a], [Hu and al.,2008], [Hu and al.,2006a], [Hu and al.,2006b], [Qu and al.,2006] et [Wang and al.,2006] pour étudier le problème de matching d'ontologies volumineuses.

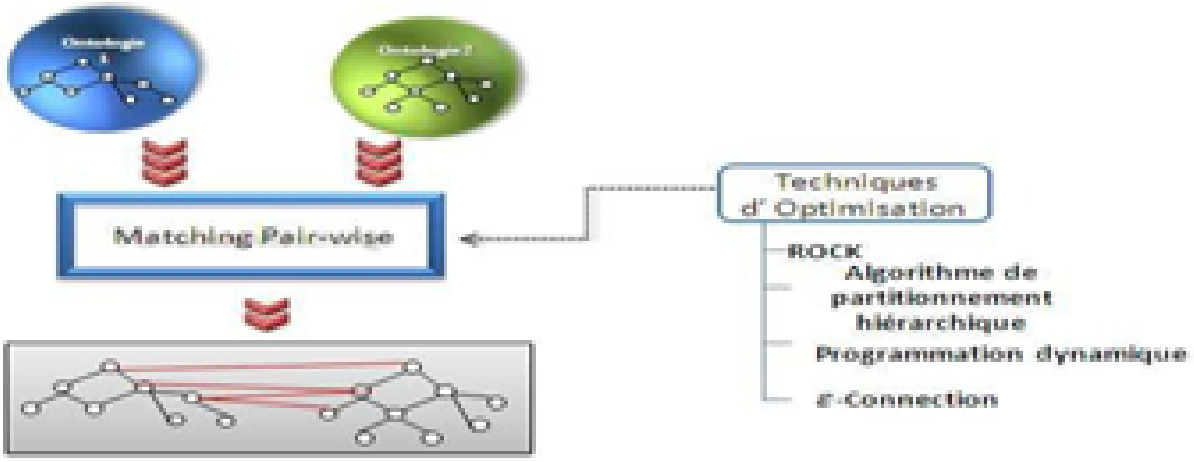

FIGURE 3.1 – *Matching d'ontologies*

Il existe deux principales stratégies d'alignement des ontologies. Ainsi, distingue-t-on la stratégie basée sur le partitionnement de l'ontologie [Pereira and al.,2017] et la stratégie basée sur l'extraction de module ontologique [Santos and al.,2015]. Ces méthodes ne peuvent fonctionner que si le nombre de concepts à l'entrée de l'outil d'alignement est limité.

C'est pour cette raison que ces stratégies sont conçues de manière à décomposer respectivement des ontologies volumineuses en des blocs ou en modules d'ontologies en vue de leur alignement.

Le partitionnement subdivise une ontologie en un ensemble de sous-structures autonomes ou dépendantes les unes des autres qu'on a appellera partitions, tandis que l'extraction de module consiste à extraire une sous-ontologie en fonction d'une signature précise [D'Aquin and al., 2009].

Pour plus de clareté dans la présentation des stratégies d'alignement, nous avons introduit un critère important pour distinguer la stratégie de partitionnement de la stratégie modularisation. Ce critère concerne la réutilisation des modules extraits. Le partionnement ne permet pas nécessairement la réutilisation des blocks obtenus, tandis que la modularisation permet la réutilisation des modules obtenus.

Jusqu'à présent les stratégies existantes restent encore à améliorer car elles ont initialement traité des ontologies légères. Ces stratégies sont confrontés à des problèmes techniques et de sémantique dès qu'elles s'attaquent à l'alignement des ontologies volumineuses. Parmi ces problèmes, on peut citer celui de manque de capacité mémoire pour le traitement du calcul de similarité des noms de classe (ou autres labels,...), ainsi que les longs temps de traitement. Si la modularisation présente théoriquement une meilleure visibilité des modules à obtenir des ontologies en entrée, car il faut rappeler que cette stratégie a été adoptée initialement lors de la conception de l'ontologie. Ce qui permet de dire que les concepteurs ont pris en charge l'aspect cohérence des modules et donc réduction de perte de sémantique. Quant à la stratégie de partitionnement, les blocs sont obtenus à partir des deux ontologies de manière séparée sans considérer les correspondances existantes entre elles. Ce qui conduit nécessairement à une perte de sémantique.

Les ontologies légères ou de petite taille ne posent pas de problème de partitionnement car il existe différents algorithmes qui permettent de le faire. Tandis que lorsque les ontologies sont de très grande taille, l'efficacité des stratégies d'alignement automatique diminue considérablement et elles sont confrontées à deux problèmes majeurs ; le manque d'espace mémoire et les longs temps d'exécution. Dans ce qui suit nous allons décrire en détail chacune des deux stratégies :

3.2 Etat de l'art sur la stratégie de partitionnement des ontologies

3.2.1 Introduction

Elle a été proposée par [Hu and al.,2006b] pour le partitionnement en blocs de deux classes hiérarchiques volumineuses. Ces classes représentent un type d'ontologies volumineuses. Cette stratégie se déroule comme suit : les deux classes sont partitionnées en se basant sur les affinités structurelles et linguistiques. Les affinités structurelles déterminent les relations dans les hiérarchies et les simi-

larités linguistiques sont calculées en déterminant les similarités entre les descriptions de classes. La combinaison de ces deux similarités détermine des liens pondérés entre les classes. Ainsi les classes hiérarchiques volumineuses peuvent être divisées en des petits blocs grâce à l'utilisation de l'algorithme de partitionnement ROCK (Robust Clustering Using Links). Ensuite deux relations entre les blocs sont déterminées : une relation via les ancres qui sont des appariements préalablement connus entre les termes de deux ontologies et définis par des techniques de comparaison de chaines de caractères ou par un expert, l'autre relation est déterminée via les documents virtuels. Les ancres sont des paires de classes prédéfinies et déjà mises en correspondances. Ils sont utilisés pour déterminer la relation entre les blocs. La relation entre les blocs est aussi calculée via les documents virtuels avec la technique de TF/IDF [1]. Les documents virtuels sont construits à partir de concepts (classes, propriétés ou instances) de deux ontologies. Le document virtuel d'un bloc est une collection de jetons pondérés provenant des descriptions de classes le contenant (e.g les noms). Ces deux relations (déterminées par les ancres et les documents virtuels) sont combinées pour mettre en correspondance les blocs. Cependant, cette approche n'est pas réellement applicable pour des ontologies volumineuses car les classes ne couvrent qu'une partie des ontologies. De plus, cette approche permet le partitionnement de deux classes hiérarchiques volumineuses séparément sans considérer les correspondances entre elles. Pour résoudre le problème de matching d'ontologies volumineuses, les auteurs [Hu and al.,2006a] et [Hu and al.,2008] ont proposé une nouvelle stratégie de partitionnement pour les ontologies. Cette approche considère les caractéristiques linguistiques et structurelles des entités en se basant sur les documents virtuels et les mesures de similarités. Le partitionnement des ontologies est réalisé par un algorithme de division hiérarchique en générant des blocs de mappings. D'autres travaux ont été proposés par [Hamdi and al., 2009c] utilisant également cette stratégie de partitionnement, définie en deux méthodes, afin de partitionner deux ontologies à aligner en plusieurs blocs.

Ces méthodes identifient avec une mesure de similarité les couples de concepts issus de deux ontologies dont le label est identique et utiliseront les ancres pour effectuer les partitions. Les deux ensembles de blocs ou partitions obtenus seront ainsi mis en correspondance comprenant un bloc de chacun des deux ensembles.

3.2.2 Le partitionnement des ontologies

Dans les domaines d'applications réelles, les ontologies devenant de plus en plus volumineuses, de nombreux travaux [Stuckenschmidt and Klein, 2004], [[Grau and al.,2005], [Grau and al., 2006]] , [Hu and al.,2006a] et [[Hamdi and al., 2009b], [Hamdi and al., 2009c]] se sont intéressés au problème de leur partitionnement. Ainsi les travaux de [Stuckenschmidt and Klein, 2004] visent la décomposition d'une ontologie en sous-blocs (ou îlots) indépendants les uns des autres, de

1. TF/IDF (term frequency-inverse document frequency) : mesure statistique qui permet d'évaluer l'importance d'un mot par rapport à un document extrait d'une collection ou d'un corpus

façon à faciliter différentes opérations sur les ontologies comme la maintenance, la visualisation, la validation ou le raisonnement.

Les travaux de [Grau and al.,2005] s'intéressent plus particulièrement aux problèmes de raisonnement et cherchent à construire des modules centrés autour d'une sous- thématique qui soient cohérents et auto suffisants pour raisonner.

Seul [Hu and al.,2006a] a pour objectif l'alignement d'ontologies mais nous verrons que sa méthode de décomposition ne prend pas complètement en compte toutes les contraintes imposées par cet objectif, en particulier le fait de travailler sur deux ontologies.

[[Hamdi and al., 2009b], [Hamdi and al., 2009c]] proposent deux méthodes différentes . Pour prendre en compte, au plus tôt, l'objectif d'alignement, ces méthodes s'appuient sur deux éléments : d'une part les couples de concepts issus des deux ontologies qui ont exactement le même label et peuvent être reliés par une relation d'équivalence, d'autre part l'asymétrie structurelle possible des deux ontologies à aligner.

3.2.3 Des modules indépendants facilitant la gestion d'ontologies volumineuses

L'approche de Stuckenschmidt et Klein [Stuckenschmidt and Klein, 2004] consiste à partitionner automatiquement des ontologies basée sur la structure de la hiérarchie des classes (concepts). Elle est basée sur l'hypothèse que les dépendances entre les concepts peuvent être dérivées de la structure même de l'ontologie. Cette dernière peut être représentée sous forme d'un graphe pondéré O = (C, D, w) où les nœuds (C) représentent des concepts et les arêtes (D) représentent des relations entre concepts, dont le poids (w) varie en fonction de la dépendance. Dans cette approche le partitionnement se fait en deux étapes :

— Extraire de l'arbre de dépendance qui est en fait un sous-graphe de l'ontologie initiale.
— Calculer le degré proportionnel de dépendance entre les concepts (Fig.3.2). Pour ce faire, le degré proportionnel de dépendance w(ci ,cj) est calculé entre deux concepts c_i et c_j, avec a_{ij} qui est le poids assigné à la relation qui les unit :

où a_{ij} est le poids assigné à la relation entre les concepts c_i et c_j.

Stuckenschmidt et Klein fixent la valeur de a_{ij} à 1 dans leurs expériences. Ainsi, le degré proportionnel de dépendance sera égal au rapport de a_{ij} par le nombre de concepts auquel le concept c_i est connecté [Stuckenschmidt and Klein, 2004].

La figure 3.2 fait état du cas où, le nœud d est utilisé dans le calcul des poids à l'aide des dépendances proportionnelles. Le nœud d a quatre voisins directs, ce qui entraine que le degré proportionnel de dépendance de la relation entre ce nœud et ses voisins est de 0.25 (la valeur a_{ij} qui est fixée à 1 par défaut divisée par quatre). On constate que différents niveaux de dépendances entre d et ses voisins proviennent de la dépendance de ces voisins là avec le nœud **d**. Toutefois, il est important de souligner que ce degré proportionnel de dépendances est non

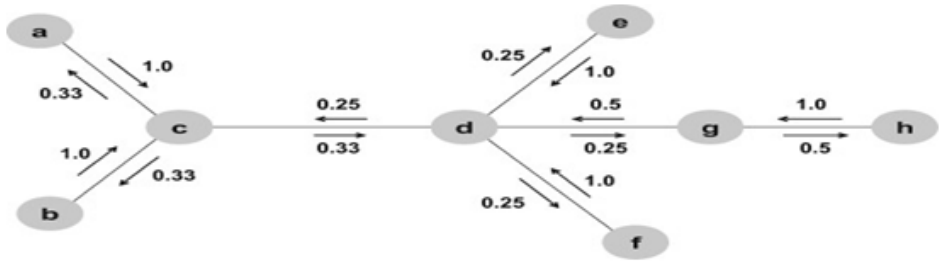

FIGURE 3.2 – *Un exemple de graphe avec des dépendances proportionnelles de force [Stuckenschmidt and Schlicht , 2009]*

symétrique, c'est-à-dire qu'il sera différent respectivement pour les relations et les nœuds e et **f** n'ont aucun voisins directs autre que **d** ou bien qui dépendent de celui-ci. De ce fait, le poids de leurs relations respectives avec le nœud **d** demeurent égal à 1. Le degré proportionnel de dépendances entre les nœuds **g** et **d** est de 0.5, car **g** n'a que deux voisins. Quant à la dépendance entre les nœuds **b** et **d** est de 0.33 étant donne que **b** n'a que trois voisins [Patrick, 2014].

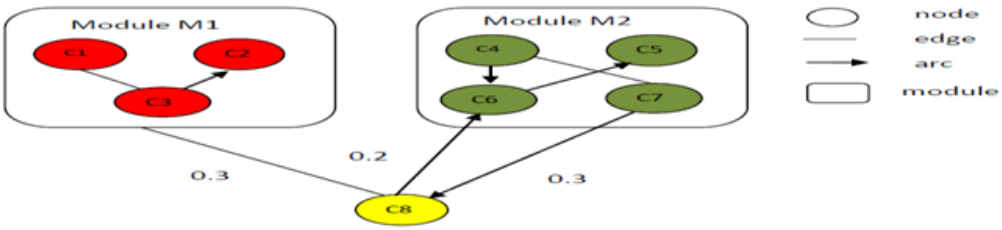

FIGURE 3.3 – *Exemple de graphe pour l'attribution des nœuds restant à des modules [Stuckenschmidt and Schlicht , 2009]*

Afin de déterminer les partitions, on procède à une analyse du graphe, qui représente un réseau de dépendances proportionnelles dans lequel les nœuds fortement liés seront regroupés sous forme de "cluster" (Fig.3.3). En effet, ces derniers représentent des sous- graphes à l'intérieur desquels les arêtes sont liées plus fortement entre elles qu'à toutes autres arêtes voisines, qui sont en fait des arêtes les liant ainsi à des concepts ou sous-graphes voisins. La taille du sous graphe est un paramètre que l'utilisateur se doit de fixer au début du processus de partitionnement. Stuckenschmidt et Klein estiment que les concepts dans un module se doivent d'être fortement interconnectés. Cette propriété du module permettrait d'identifier tout d'abord les concepts qui ne le sont pas afin de les exclure du cluster. Une fois les limites du potentiel module défini, chacun des concepts qui aurait été préalablement exclu sera rajouté au dit module uniquement dans la mesure où ce dernier contiendrait un ou plusieurs concepts auxquels ce concept potentiellement pertinent soit lié, et ce plus qu'à tout autre concept appartenant à un autre cluster ou potentiel module voisin.

Finalement, l'objectif de [Stuckenschmidt and Klein, 2004] est de décomposer une ontologie en blocs indépendants et cohérents. La méthode consiste donc à trouver des blocs (îlots) à partir d'un graphe de dépendance. Le processus de

partitionnement passe par cinq étapes :

— **Création du graphe de dépendance :**

Le graphe de dépendance est extrait à partir du fichier source de l'ontologie. L'idée est que les éléments de l'ontologie (concepts, relations, instances) sont représentés par des nœuds du graphe. Les liens entre les nœuds sont introduits si les éléments correspondants sont liés dans l'ontologie.

— **Déterminer la force de dépendance :**
La force des dépendances entre les concepts est déterminée en utilisant des algorithmes d'analyse de réseau. Elle est basée sur le nombre de liens auxquels participe un nœud. L'idée est que moins un nœud est relié à d'autres nœuds, plus les liens avec les nœuds avec lesquels il est lié sont forts. Des poids peuvent être introduits pour représenter l'importance des différents types de liens. Ainsi on peut décider que les liens traduisant les relations de sous-classe sont plus importants que les autres relations du domaine.

— **Déterminer les modules :**

Un module ou îlot ("line island") est défini comme étant un ensemble de nœuds, de taille comprise entre des valeurs minimales et maximales données, tels que la force de chaque connexion interne à l'ensemble est supérieure à la force de chacune des connexions reliant un des nœuds de l'ensemble avec l'extérieur.

— **Attribuer les concepts isolés :**

Cette approche fixe une borne minimale à la taille des modules. Un mauvais choix de cette borne fait apparaître, d'après les auteurs, de très nombreux concepts isolés dont l'affectation doit être forcée dans le module avec lequel ils ont la plus forte connexion.

— **Fusion :**

L'algorithme a aussi tendance à construire beaucoup de petits blocs avec une très forte corrélation interne dont il faut là aussi forcer la fusion. Celle-ci est décidée si certains modules voisins sont assez fortement liés. Dans de nombreux cas, il n'existe qu'un module adjacent pour fusionner. Dans les cas où plus d'un module adjacent existe, la fusion est faite avec le voisin le plus proche, déterminé par la force des dépendances entre les modules.
Le processus de génération des modules impose une contrainte sur la taille minimale des modules générés conduisant à des regroupements pas forcément très pertinents sémantiquement. Il construit, par ailleurs, beaucoup de petits blocs. Ces deux raisons ne sont pas favorables à l'utilisation de ce processus dans un but d'alignement d'ontologies.

3.2.4 Des modules autonomes pour le raisonnement

Grau et al [Grau and al.,2005] proposent une approche basée sur les ε-connexions pour résoudre le problème du partitionnement d'une ontologie, qui est considéré non seulement comme un langage de représentation de connaissances (combinaison de différents formalismes logiques) mais également comme un langage de définition et d'instanciation d'ontologies OWL.

Les ε-connexions ont été élaborées dans le but d'accroître l'expressivité de chaque composant logique tout en préservant la décidabilité des raisonneurs [Kutz and al., 2004]. Ce langage permet de séparer distinctement les domaines d'interprétation de n-systèmes combinés (chaque système peut être vu comme étant une base de connaissances en logique de description), où ces domaines sont liés à l'aide de n relations de type «link relations». Ces relations permettent de représenter les connexions entre différentes partitions de telle sorte que le raisonnement peut se faire sur chaque partition individuellement ou plutôt sur une combinaison de partitions liées entre elles.

Les partitions générées par les ϵ-connexion sont à la fois structurellement et sémantiquement compatibles [Grau and al.,2005]. Soient une ontologie O et une ε-connexion Σ ayant respectivement pour vocabulaires V et Vn qui est structurellement compatible avec O ($\Sigma \sim O$) si et seulement si :

(a) V_n est un vocabulaire partitionné de O. Ce qui sous-entend que Σ contient exactement les mêmes entités (concepts, propriétés et individus) et axiomes que O.

(b) $A \in \Sigma \iff A \in O$

Σ est sémantiquement compatible avec O ($\Sigma \approx O$) si et seulement si :

i. O est partitionnable pour V_n.

ii. Si $I(V_n)M$, alors $M \models \Sigma$ ssi $I(V_n \models O)$

Les ε-connexion assurent deux types de compatibilité : structurelle et sémantique.

La compatibilité structurelle garantie qu'aucune entité ou axiome ne sera ajouté, retiré ou modifié lors du partitionnement. En d'autres termes, on tient à s'assurer que chaque axiome existant dans la ε- connexion existe aussi dans l'ontologie. La compatibilité sémantique représente ici la relation souhaitée entre l'ontologie initiale et l'ontologie résultante du processus de partitionnement. En effet, le rôle de la compatibilité sémantique est de garantir que l'interprétation de l'ontologie partitionnée est équivalente à l'interprétation de la même ontologie non partitionnée, donnant lieu ainsi à la préservation du modèle d'interprétation. On s'assure principalement que chaque ontologie équivalente corresponde exactement au même ensemble de ε-connexions compatibles. L'approche de partitionnement de Cuenca [Grau and al.,2005] permet ainsi d'identifier les propriétés dans O qui lient O à Σ_i ou Σ_i à O , tout en garantissant à la fois les compatibilités structurelle et sémantique [Patrick, 2014]. Dans [Grau and al., 2006] la méthode de partitionnement vise à produire des modules autonomes dans le sens où toutes

les inférences à l'intérieur d'un module peuvent être faites uniquement sur la base d'un raisonnement local [Grau and al., 2006]. La méthode comporte trois étapes de base :

(a) **Safety Check :**
Cette étape vérifie que l'ontologie est partitionnable. Une ontologie est partitionnable si elle ne contient pas d'axiomes d'inclusion dits "dangereux" (des axiomes d'inclusion qui imposent des contraintes sémantiques sur l'ensemble de l'ontologie) pour la préservation de la compatibilité sémantique de ses partitions.

(b) **Partitionnement :**
L'algorithme appliqué dans cette étape débute avec une seule partition. Il crée ensuite une nouvelle partition en y introduisant un concept ou un axiome quelconque puis tous les concepts ou axiomes dépendants, i.e. qui y font référence et peuvent être déplacés dans la nouvelle partition sans violer la condition de complétude.

(c) **Génération de module :**
Le partitionnement créé dans la deuxième étape est utilisé pour déterminer les modules. Cela se fait par la fusion des partitions au sein des modules qui se chevauchent. A ce stade, il est possible d'introduire de la redondance dans l'ontologie.

Cette méthode garantit que tous les concepts reliés par des liens de subsumption seront regroupés dans un seul module. Ceci est un inconvénient majeur pour l'alignement d'ontologies qui comportent des milliers de relations de subsomption (c'est le cas, par exemple, de AGROVOC et NALT). Les auteurs conviennent que cette méthode peut conduire à la création de très mauvaises répartitions de tailles des modules dans le cas d'ontologies "monolithiques", qui ne seront pas du tout adaptées pour l'alignement.

Des modules à base de cohésion et couplage

Dans [Hu and al.,2006b], leur méthode consiste à partitionner chaque ontologie en blocs en utilisant la méthode de clustering Rock [Guha and al.,2000]. L'algorithme permettant de partitionner une ontologie en blocs s'appuie sur deux notions essentielles : la cohésion au sein d'un bloc et le couplage entre deux blocs distincts.

La cohésion mesure la somme des poids des liens reliant les concepts appartenant à un même bloc et le couplage, la somme des poids des liens reliant les concepts appartenant à deux blocs différents. Pour effectuer la partition, alors que [Guha and al.,2000] considèrent que les liens entre les concepts ont tous la même valeur, [Hu and al.,2006b], introduisent la notion de liens pondérés qui s'appuie sur deux mesures de similarité entre concepts, une mesure linguistique et une mesure structurelle.

Des modules ordonnés en fonction de leurs structures

Dans [[Hamdi and al., 2009b], [Hamdi and al., 2009c]], même avec de grandes ontologies, il est possible d'identifier, avec une mesure de similarité stricte et peu coûteuse à calculer, des concepts qui ont un label en commun dans les ontologies. Les partitions sont générées à partir des paires de concepts ancres.

L'asymétrie structurelle des deux ontologies peut servir à ordonner les ontologies à partitionner. Si une ontologie est plus structurée que l'autre, il sera plus facile de la décomposer en blocs avec une forte cohésion interne. Ainsi sa décomposition peut servir comme un guide pour la décomposition de l'autre ontologie. La solution proposée est de limiter la taille des ensembles de concepts en entrée de l'outil d'alignement, et pour cela de partitionner les deux ontologies à aligner en plusieurs blocs, afin de n'avoir à traiter que des blocs de taille raisonnable. Cela signifie que les différents blocs obtenus après les partitions devront être ensuite alignés par paires, composées de deux blocs issus chacun d'une des deux ontologies initiales, et l'objectif consiste à minimiser le nombre de paires à aligner.

Partitionner un ensemble E consiste alors à trouver des sous-ensembles $E_1, E_2, ..., E_n$, d'éléments sémantiquement proches c'est-à-dire liés par un ensemble de relations important. La réalisation de cet objectif consiste à maximiser les relations au sein d'un sous-ensemble et à minimiser les relations entre les différents sous-ensembles.

La qualité du résultat d'un partitionnement généré par l'algorithme établi, peut être apprécié selon différents critères :

La taille des blocs générés : les blocs doivent avoir une taille raisonnable, i.e. inférieure au nombre d'éléments que peut traiter l'outil d'alignement.

Le nombre de blocs générés : ce nombre doit être le plus faible possible pour limiter le nombre de paires de blocs à aligner.

Le degré de dépendance entre les blocs : un bloc sera dit faiblement dépendant des autres si les relations (lexicales et structurelles) sont fortes à l'intérieur du bloc et faibles à l'extérieur. Ce degré regroupe les éléments qui peuvent probablement s'aligner dans un nombre minimal de blocs et réduit ainsi le nombre de comparaisons à faire.

3.2.5 Méthodes d'alignement des ontologies larges

La méthode PBM

A notre connaissance, la seule méthode adaptée à l'alignement d'ontologies volumineuse est la méthode PBM [Hu and al.,2006b]. Cette méthode consiste à partitionner chaque ontologie en blocs en utilisant la méthode de clustering Rock [Guha and al.,2000], puis à mesurer la proximité de chacun des blocs d'une ontologie avec chaque bloc de l'autre ontologie de façon à n'effectuer l'alignement qu'entre les concepts des paires de blocs les plus proches.

Pour effectuer la partition, alors que Rock considère que les liens entre les concepts ont tous la même valeur, PBM introduit la notion de liens pondérés

qui s'appuie sur deux mesures de similarité entre concepts, une mesure linguistique et une mesure structurelle.

Mesures de similarité

Similarité lexicale : Soient d_i (resp. d_j) la chaîne de caractères correspondant à la description du concept c_i (resp. c_j) et comm(d_i, d_j) (resp. diff (d_i, d_j)) le nombre de caractères communs (resp. différents) dans les chaînes d_i et d_j.
La similarité lexicale entre deux concepts c_i et c_j, $Sim_L(c_i, c_j)$ se calcule comme suit :
$sim_L(c_i, c_j) = comm(d_i, d_j) - diff(d_i, d_j) + Winkler(d_i, d_j)$
où le terme Winkler(d_i, d_j) est ajouté pour améliorer le résultat en utilisant la méthode de Winkler [Winkler, 1999] (méthode de comparaison de chaîne de caractères).
Les concepts c_i et c_j sont jugés similaires si $Sim_L(c_i, c_j) > 0.65$.

Similarité structurelle

Soient c_i, c_j deux concepts d'une même ontologie O, c_{ij} leur plus petit ancêtre commun et depthOf (c) la distance en nombre d'arcs entre le concept c et la racine de O. PBM mesure la similarité structurelle $aff_s(c_i, c_j)$ en utilisant la mesure de Wu et Palmer [Wu and Palmer., 1994] qui se calcule comme suit :

$$aff_s(c_i, c_j) \models \frac{2 * depthOf(c_{ij})}{depthOf(c_i) + depthOf(c_j)}$$

Le calcul de similarité structurelle entre les concepts d'une ontologie de grande taille peut prendre beaucoup de temps. En considérant que seuls les concepts de profondeurs adjacentes auront des similarités élevées, PBM ne compare que les concepts qui satisfont la relation suivante :

$$|depthOf(c_i) - depthOf(c_j)| \leq 1$$

Les liens pondérés :

Le calcul du lien pondéré entre deux concepts, link(c_i, c_j), s'effectue comme suit :

$$link(c_i, c_j) = \left\{ \begin{array}{ll} aff(c_i, c_j) & si \ \ aff(c_i, c_j) > \epsilon_1 \\ 0 & sinon \end{array} \right\}$$

$$aff(c_i, c_j) = \alpha \ . \ aff_s(c_i, c_j) + (1 - \alpha) \ . \ Sim_L(c_i, cj)$$

où s_1 est un seuil donné tel que $\epsilon_1 \in [0, 1] \ \ et \ \ \alpha \in [0, 1]$ permet à l'utilisateur de faire varier le poids relatif des mesures de similarité.

Algorithme de Partitionnement

L'algorithme permettant à PBM de partitionner une ontologie en blocs s'appuie sur deux notions essentielles : la cohésion au sein d'un bloc et le couplage entre deux blocs distincts. La cohésion mesure la somme des poids des liens reliant les concepts appartenant à un même bloc et le couplage, la somme des poids des liens reliant les concepts appartenant à deux blocs différents. Ces deux notions sont représentées au sein d'une même mesure dite goodness dont le sens varie suivant qu'elle s'applique sur un bloc unique B_i ou sur deux blocs distincts, B_i et B_j tels que $Bi \neq Bj$:

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

$$Couplage(B_i, B_j) \;=\; goodness(B_i, B_j) \;\; avec \;\; B_i \neq B_j$$

$$goodness(B_i, B_j) \;=\; \frac{\sum_{c_i \in B_i, c_j \in B_j} link(c_i, c_j)}{sizeOf(B_i).sizeOf(B_j)}$$

L'algorithme prend en entrée l'ensemble B des n blocs à partitionner, où chaque bloc est réduit au départ à un unique concept, et la taille maximale t_max à partir de laquelle les blocs ne doivent plus être fusionnés. PBM initialise tout d'abord de façon uniforme, la valeur de cohésion de chaque bloc ainsi que les valeurs de couplage. A chaque itération, l'algorithme choisit le bloc qui a la cohésion maximale et le bloc qui a la valeur de couplage maximale avec ce premier bloc. Il remplace ces deux blocs par celui résultant de leur fusion et met à jour les valeurs de couplage de tous les blocs avec ce nouveau bloc. L'algorithme s'arrête quand tous les blocs construits ont atteint la taille limite fixée au départ. Le pseudo-code de cet algorithme est présenté ci-dessous dans Algorithme PBM :

Algorithm PBM(B, t_{max})

Require : $B = B_1, B_2, ...B_n$ l'ensemble des blocs à partitionner.
1 : for each bloc B_i dans B do
2 : initialiser la valeur de la cohésion de B_i
3 : calculer la valeur de couplage de B_i avec tous les autres blocs
4 : end for
5 : while l'ensemble B is not Empty do
6 : choisir B_i le bloc qui a la valeur de cohésion maximale
7 : choisir B_j le bloc qui a la valeur de couplage maximale avec B_i
8 : if B_j existe then
9 : fusionner les blocs B_i et B_j dans B_p
10 : supprimer B_i et B_j de l'ensemble B
11 : if taille de $B_p > t_{max}$ / 2 then
12 : ajouter B_p à l'ensemble P

13 : supprimer B_p de l'ensemble B
14 : else
15 : mettre à jour les valeurs de cohésion et de couplage de B_p 16 : for each bloc B_k dans B sauf B_p do
17 : mettre à jour la valeur de couplage de B_k
18 : end for
19 : end if
20 : else
21 : ajouter B_i à l'ensemble P
22 : supprimer B_i de l'ensemble B
23 : end if
24 : end while
25 : return P l'ensemble des blocs.

Identification des paires de blocs à aligner

Une fois le partitionnement des deux ontologies réalisé, l'évaluation de la proximité des blocs s'effectue en s'appuyant sur des ancres, i.e. des appariements préalablement connus entre les termes des deux ontologies, définis par des techniques de comparaison de chaînes de caractères ou par un expert. Plus deux blocs contiennent d'ancres communes, plus ils sont jugés proches.

Soient k le nombre de blocs générés par la partition d'une ontologie O et B_i un de ces blocs, k' le nombre de blocs générés par la partition de la deuxième ontologie O' et B_j un de ces blocs.

Soit la fonction anchors qui calcule le nombre d'ancres prédéfinies partagées par deux blocs B_u et B_v .

Le nombre d'ancres contenues par un bloc B_i est donc calculé par la somme :

$$\sum_{v=1}^{k'} anchors(B_i, B_v^{'})$$

La relation de Proximité entre deux blocs $B_i, B_j^{'}$

$$Proximity(B_i, B_j^{'}) = \frac{2.anchors(B_i, B_j^{'})}{\sum_{u=1}^{k} anchors(B_u, B_j^{'}) + \sum_{v=1}^{k'} anchors(B_i, B_v^{'})}$$

Les paires de blocs alignées sont toutes les paires ayant une proximité supérieure à un seuil s_2 $(0, 1)$. Un bloc pourra donc être aligné avec plusieurs blocs de l'autre ontologie ou avec aucun suivant la valeur choisie pour ce seuil.

Exemple :

L'algorithme PBM est appliqué sur deux petites ontologies jouets pour visualiser son comportement.

La Fig. 3.4 présente les deux ontologies O_S et O_T avant et après le processus de partitionnement en utilisant la méthode PBM. Pour effectuer ces partitionnements, il faut utiliser une version de l'algorithme à partir du web [2] qui ne se base que sur la similarité structurelle entre concepts. Il faut souligner que les labels des concepts n'interviennent pas dans le traitement.

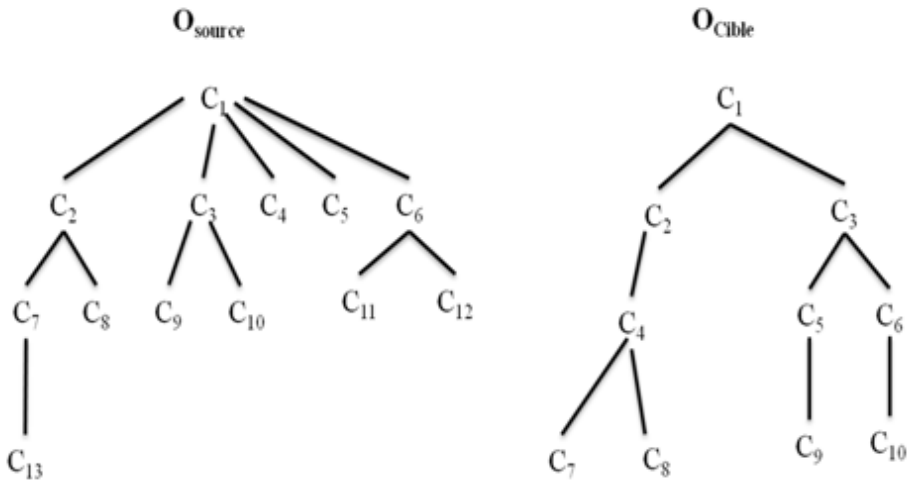

FIGURE 3.4 – *Les ontologies de l'expérimentation test*

Dans cet exemple, la taille limite est fixée à 4, ce qui signifie que l'algorithme a le droit de fusionner 2 blocs de taille 3 pour former au plus un bloc de taille 6. Il est ainsi certain d'obtenir au moins 2 blocs dans O_T (O_{cible}) qui contient 10 concepts et 3 blocs dans O_S qui contient 13.

Le partitionnement successif de O_T puis O_S fait effectivement apparaître 2 blocs pour O_T , B_{T1} et B_{T2} contenant 5 concepts chacun, et 3 blocs pour O_S , B_{S1}, B_{S2} et B_{S3} contenant respectivement 4, 3 et 6 concepts.

2. http ://iws.seu.edu.cn/projects/matching/

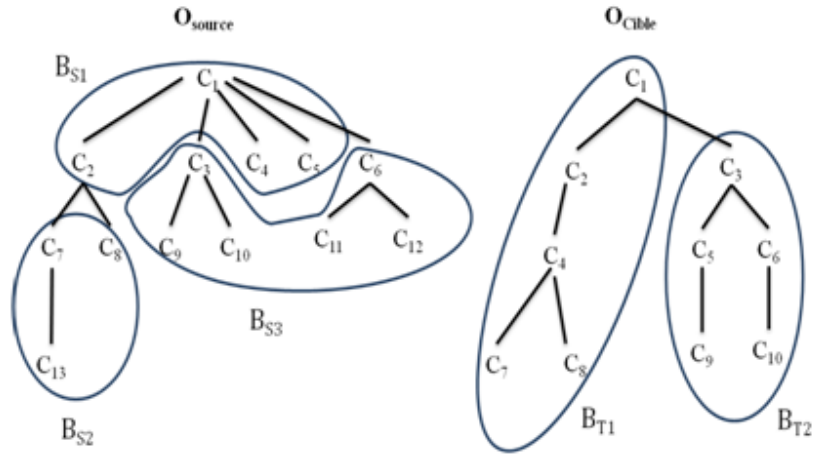

FIGURE 3.5 – *Les blocs construits avec la méthode PBM*

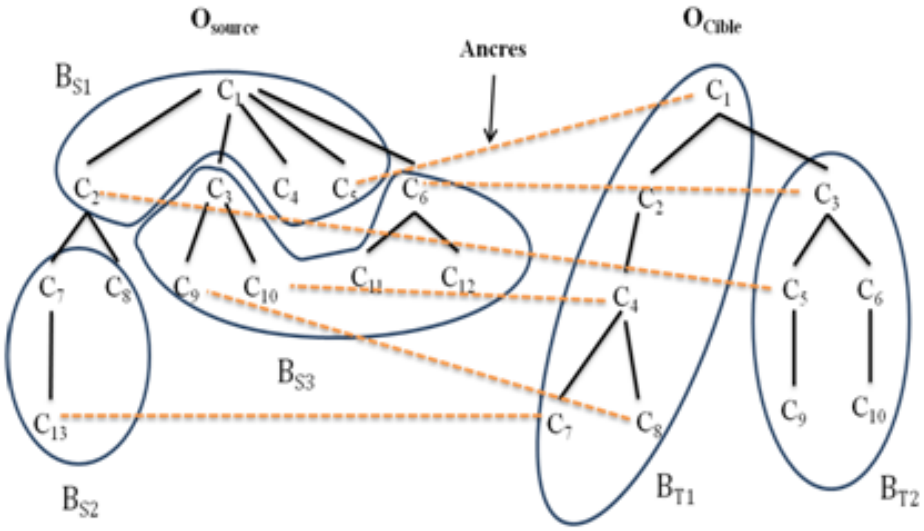

FIGURE 3.6 – *Les ancres partagées par les différents blocs*

Le bloc B_{S1} contient 2 ancres, l'une partagée avec B_{T1}, l'autre avec B_{T2}. Le bloc B_{S2} n'en contient qu'une partagée avec B_{T1}, et le bloc B_{S3} en contient 3, 2 partagées avec B_{T1}, la troisième avec B_{T2}.

Le calcul de proximité fondé sur les ancres partagées par les deux ontologies doit être effectué sur toutes les paires de blocs possibles (ici, 6 paires).

Le nombre de paires effectivement alignées dépend de la valeur fixée pour le seuil : plus celui- ci est bas, plus les alignements sont multipliés et les chances de retrouver des appariements, mais plus on augmente aussi le temps d'exécution. Si le seuil est élevé, on passe moins de temps à aligner des blocs éloignés mais on perd peut être des appariements potentiels.

Par exemple, la paire (B_{S1}, B_{T1}) ne partageant qu'une ancre alors que les blocs en contiennent respectivement 2 et 4, sa proximité est égale à 0.33. Si le seuil est fixé à 0.5, la paire n'est pas alignée et l'ancre partagée n'est pas retrouvée dans les appariements. Si le seuil est fixé plus bas, on retrouvera toutes les ancres dans

les appariements, mais il faudra aligner toutes les paires de blocs possibles à l'exception de la paire (B_{S2}, B_{T2}) qui ne partage pas d'ancre.

Cette méthode permet de décomposer des ontologies volumineuses, mais cette décomposition est faite a priori, sans prendre en compte l'objectif d'alignement, en s'appliquant sur chaque ontologie indépendamment l'une de l'autre. Pour effectuer l'alignement, PBM doit d'abord identifier quels sont les blocs les plus proches qui doivent être alignés entre eux et pour cela calculer la proximité des différents blocs en s'appuyant sur les couples d'ancres préalablement connues. Le partitionnement ayant été fait au départ à l'aveuglette, certaines ancres pourront ne pas se trouver dans des blocs finalement alignés et l'alignement résultant ne comprendra pas forcément tous les appariements possibles.

Enfin, le calcul des blocs pertinents à aligner est coûteux (en temps de traitement). Malgré ces critiques, l'algorithme de décomposition de PBM est le plus adapté à la tâche d'alignement puisqu'il permet de contrôler la taille maximale des blocs construits. Dans cette méthode une adaptation est effectuée afin de modifier la façon de générer les blocs en prenant en considération, au plus tôt lors du partitionnement, les relations existant entre les concepts des deux ontologies.

Méthode PAP (Partition, Anchor, Partition)

Elle consiste d'abord à décomposer l'ontologie la plus structurée dite cible O_T , puis à utiliser les ancres identifiées, pour forcer le partitionnement de la deuxième ontologie dite source O_S à être en accord avec le partitionnement de O_T . Cette première méthode casse partiellement la structure de la source de O_S . Ceci n'est pas un problème quand l'ontologie source est mal structurée.

Lorsque les deux ontologies sont structurées selon le même point de vue, la méthode PAP est inadaptée. Dans ce cas une autre méthode de partitionnement, appelée APP (Anchor, Partition, Partition), qui prend en compte la structure des deux ontologies. La méthode APP partitionne OT en favorisant la fusion de blocs partageant des ancres avec O_S , et partitionne O_S en favorisant la fusion des blocs partageant des ancres avec le même bloc de O_T .

Les sections suivantes présentent successivement la mesure de similarité utilisée pour calculer les ancres à moindre coût, puis la description détaillée des deux méthodes.

Mesure de similarité lexicale

Pour calculer les ancres, une mesure de similarité lexicale stricte est utilisée. Deux concepts sont équivalents si et seulement si leurs labels sont parfaitement identiques. Aucun prétraitement (filtrage, tokenisation, lemmatisation) n'est effectué sur ces labels. Cette mesure est définie comme suit :

$$sim_{light}(Label(c_i), Label(c_j)) = \begin{cases} 1 & si \;\; Label(c_i) = Label(c_j) \\ 0 & sinon \end{cases}$$

c_i, c_j : deux concepts tels que $c_i \in O_i$ et $c_j \in O_j$

La méthode PAP partitionne O_S en tenant compte du partitionnement de O_T . Elle comprend quatre étapes, en plus du calcul des ancres :

— Partitionner l'ontologie cible O_T en plusieurs blocs B_{Ti} en utilisant l'algorithme PBM.
— Identifier, dans chacun des blocs construits pour O_T , l'ensemble des ancres appartenant à ce bloc.
— Chacun de ces ensembles constituera le noyau ou centre CB_{Si} d'un futur bloc B_{Si} à générer dans l'ontologie source O_S .
— Partitionner l'ontologie source autour des centres CB_{Si} identifiés dans l'étape précédente.
— Aligner chaque bloc de O_S avec le bloc de O_T correspondant.
— Déterminer les blocs de O_T

Pour déterminer les blocs de l'ontologie cible O_T , l'algorithme PBM est utilisé tel qu'il est sur le web [3]. La différence entre cet algorithme et celui décrit dans la présentation théorique faite en section 3.1 à partir de l'article [Hu et al., 2006b], est que le lien pondéré calculé entre les concepts ne dépend que de la similarité structurelle ($\alpha = 1$ dans l'équation suivante). En effet, le calcul de la similarité linguistique prend beaucoup de temps, en plus, les concepts d'une même ontologie ont souvent des labels différents.

$$aff(c_i, c_j) = \alpha \ . \ aff_s(c_i, c_j) + (1 - \alpha) \ . \ Sim_L(c_i, c_j)$$

— Déterminer les centres de O_S

Les centres des futurs blocs de l'ontologie source O_S sont déterminés en se basant sur deux critères : les couples d'ancres identifiés entre O_S et O_T , et les blocs B_{Ti} construits à partir de l'ontologie cible O_T . Pour chaque bloc B_{Ti} construit à l'étape précédente, les concepts de O_S qui ont des équivalents dans ce bloc sont regroupés dans un paquet, CB_{Si}, en utilisant l'algorithme PAP_Centre :

Algorithm PAP_Centre(T, E, S)

Require : $C = B_{T1}, B_{T2}, ...B_{Tn}$ l'ensemble des blocs de l'ontologie cible O_T .
Require : $E = E_1, E_2, ...E_m$ l'ensemble des couples de labels identiques trouvés, tels que
$E_i = (c_{Si}, c_{Tj})$, où c_{Si} est un concept de l'ontologie source O_S et c_{Tj} est un concept de l'ontologie cible O_T .

1 : for each bloc B_{Ti} dans T do
2 : initialiser $CB_{Si} = \varnothing$
3 : for each concept c_{Tk} dans B_{Ti} do
4 : for each équivalence E_j dans E do

3. http ://iws.seu.edu.cn/projects/matching/

5 : if $c_{Tk} \in E_j$ then
6 : $CB_{Si} = CB_{Si} \cup c_{Sk}$
7 : end if
8 : end for
9 : end for
10 : end for
11 : return $S = CB_{S1}, CB_{S2}, ...CB_{Sn}$ les centres des futurs blocs de l'ontologie source O_S .

Partition de OS autour des centres CB_{Si}

Après l'identification des centres des futurs blocs de O_S , l'algorithme PBM est appliqué avec la différence suivante. Au lieu d'introduire en entrée l'ensemble des m concepts de l'ontologie comme m blocs réduits chacun à un unique concept, les n centres identifiés à l'étape précédente sont introduits , comme autant de blocs distincts mais regroupant plusieurs concepts. Les autres concepts de O_S qui n'ont pas d'équivalents dans O_T donnent chacun naissance à un bloc individuel. La cohésion des blocs représentant les centres de O_S est initialisée avec la valeur de cohésion maximale.

Identification des blocs à aligner

La phase d'identification des paires de blocs à aligner est ici immédiate et sans calcul. Chacun des blocs B_{Si} construits à partir d'un centre n'est aligné qu'avec le bloc B_{Ti} correspondant. L'algorithme peut mener à la constitution de blocs B_{Sj} indépendants des centres, i.e. ne contenant pas d'ancres et qui, dans l'état courant de l'implémentation, ne sont pas pris en compte dans le processus des blocs réduits chacun à un unique concept, les n centres identifiés à l'étape précédente sont introduits , comme autant de blocs distincts mais regroupant plusieurs concepts. Les autres concepts de O_S qui n'ont pas d'équivalents dans O_T donnent chacun naissance à un bloc individuel. La cohésion des blocs représentant les centres de O_S est initialisée avec la valeur de cohésion maximale.

Identification des blocs à aligner

La phase d'identification des paires de blocs à aligner est ici immédiate et sans calcul. Chacun des blocs B_{Si} construits à partir d'un centre n'est aligné qu'avec le bloc B_{Ti} correspondant. L'algorithme peut mener à la constitution de blocs B_{Sj} indépendants des centres, i.e. ne contenant pas d'ancres et qui, dans l'état courant de l'implémentation, ne sont pas pris en compte dans le processus d'appariement. En effet, ce bloc ne contenant pas d'ancres, il faut choisir une heuristique permettant de choisir les blocs B_{Ti} avec lesquels le rapprochement est possible.

Exemple : La méthode PAP est appliquée sur les ontologies de l'exemple présenté en Fig. 3.4. La génération des blocs de la cible s'effectue comme dans la méthode PBM (*Fig*.3.7).

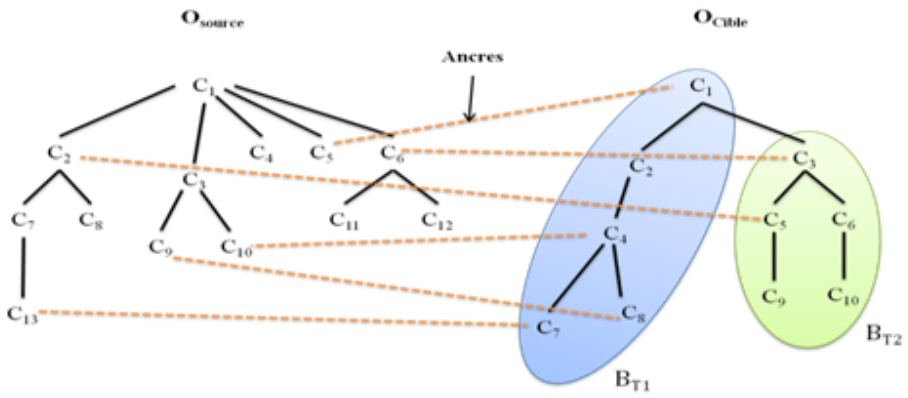

FIGURE 3.7 – *Génération des blocs de la cible identique à celle de PBM*

A partir des blocs générés pour la cible B_{T1} et B_{T2}, l'algorithme identifie les centres des futurs blocs de l'ontologie source ; $CB_{S1} : \{c_5, c_9, c_{10}, c_{13}\}$ et $CB_{S2} : \{c_2, c_6\}$ (*Fig*.3.8).

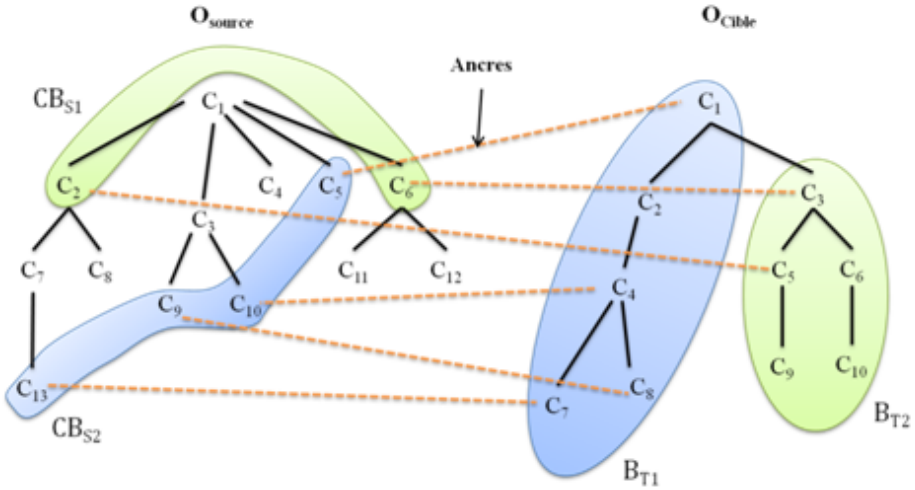

FIGURE 3.8 – *Identification des centres des blocs de l'ontologie source*

La figure 3.9 montre les blocs construits après le partitionnement de O_S à partir de ces centres. Le test sur la taille maximale des blocs construits ne s'effectuant pour les centres qu'après l'exécution d'une première fusion, le bloc B_{S1} contiendra 7 concepts au lieu de 6, mais ne peut plus être fusionné avec un autre bloc. De même, après une première fusion, le bloc B_{S2} ne peut plus être fusionné puisque la taille atteinte (4 concepts) est la taille limite. Le partitionnement fait donc apparaître un bloc sans ancre, qui ne sera pas aligné. Les paires devant être alignées (B_{S1}, B_{T1}) et (B_{S2}, B_{T2}) sont immédiatement identifiables par construction.

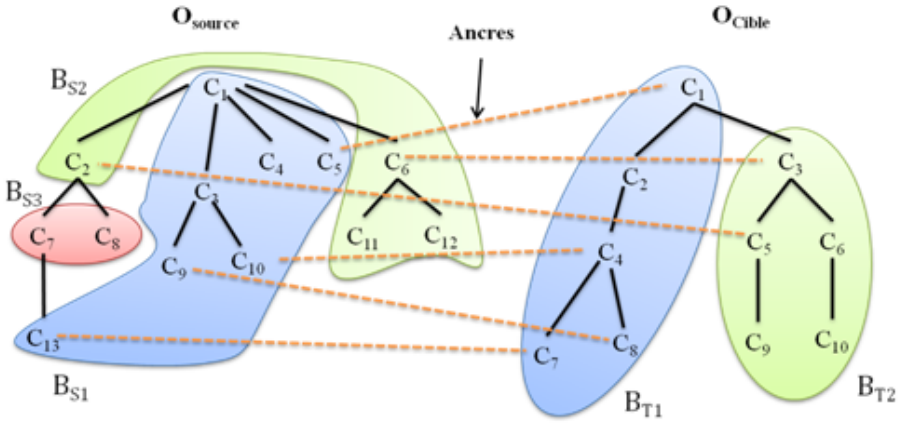

FIGURE 3.9 – *Génération des blocs de la source à partir des centres constitués précédemment*

Méthode APP

L'idée de cette méthode est de partitionner les deux ontologies en même temps, c.à.d. de faire du co-clustering. Le problème est que les ontologies ne peuvent pas être traitées réellement en parallèle du fait de leur grande taille. Pour simuler le parallélisme, le partitionnement de l'ontologie cible est nécessaire en se fondant sur toutes les équivalences identifiées avec l'ontologie source, ensuite le partitionnement de l'ontologie source est effectuée en se basant sur une partie des équivalences trouvées dans les blocs générés pour la cible. Dans cette méthode les deux ontologies cible et source sont supposées être structurées selon le même point de vue.

Prendre en compte les relations d'équivalence identifiées entre les ontologies dès le partitionnement de O_T , doit permettre par la suite de faciliter la recherche des paires de blocs les plus proches et d'améliorer les résultats de l'alignement. Ainsi ce partitionnement, contrairement à celui de PBM ou à celui implémenté dans la méthode PAP, constitue, pour les deux ontologies, des blocs qui regroupent un maximum d'ancres et qui sont structurellement proches.

Cette méthode comprend trois étapes :

i Partitionner la première ontologie O_T en utilisant l'algorithme PBM mais en prenant en compte l'ensemble des ancres lors de la génération des blocs.

ii Partitionner la deuxième ontologie O_S de la même manière mais en constituant des blocs contenant des ancres qui appartiennent à un même bloc de l'ontologie O_T .

iii Aligner les paires de blocs partageant le plus d'ancres.

Déterminer les blocs de O_T

Pour déterminer les blocs de l'ontologie cible O_T , l'algorithme PBM est utilisé en modifiant la définition de la mesure de goodness pour prendre en compte les liens entre les deux ontologies. Un coefficient est ajouté . Il représente la pro-

portion d'ancres présentes dans un bloc de O_T relativement au nombre d'ancres total identifiées avec l'ontologie O_S .
De ce fait, au cours de la génération des blocs, le choix du bloc qui a la valeur maximale de cohésion ou de couplage ne dépend pas seulement des relations des concepts à l'intérieur ou à l'extérieur des blocs d'une même ontologie, mais aussi des relations d'équivalences identifiées avec l'autre ontologie.
L'équation de goodness devient :

$$goodness(B_i, B_j) = \alpha\Big(\frac{\sum_{c_i \in B_i, c_j \in B_j} link(c_i, c_j)}{sizeOf(B_i).sizeOf(B_j)}\Big) + (1-\alpha)\Big(\frac{\sum_{c_j \in B_j, c_k \in O_s} sim_{light}(c_j, c_k)}{\sum_{c_n \in O_c, c_m \in O_s} sim_{light}(c_n, c_m)}\Big)$$

où $\alpha \in [0,1]$, $B_i$etB_j sont deux blocs de O_T, $\sum_{c_j \in B_j, c_k \in O_S} sim_{light}(c_j, c_k)$ représente le nombre d'ancres présentes dans B_j et $\sum_{c_n \in O_T, c_m \in O_S} sim_{light}(c_n, c_m)$, le nombre d'ancres total.

Déterminer les blocs de O_S

Les blocs de l'ontologie source O_S sont générés en se basant sur les poids des liens entre les concepts de O_S , les relations d'équivalences entre les deux ontologies et les blocs de l'ontologie O_T . En considérant un bloc B_i dans O_S avec une valeur de cohésion maximale, le calcul de goodness pour trouver le bloc ayant la valeur de couplage maximale avec B_i se base non seulement sur les relations structurelles dans O_S , mais aussi sur les relations d'équivalences avec le bloc de O_T qui a le maximum d'équivalents avec B_i.
L'équation de goodness devient :

$$goodness(B_i, B_j) = \alpha\Big(\frac{\sum_{c_i \in B_i, c_j \in B_j} link(c_i, c_j)}{sizeOf(B_i).sizeOf(B_j)}\Big) + (1-\alpha)\Big(\frac{\sum_{c_j \in B_j, c_k \in B_k} sim_{light}(c_j, c_k)}{\sum_{c_n \in O_c, c_m \in O_s} sim_{light}(c_n, c_m)}\Big)$$

où $\alpha \in [0,1]$, B_i et B_j sont deux blocs distincts de O_S et où B_k est le bloc de O_T qui partage le plus d'ancres avec B_i.

Lors des opérations de fusion, pour chaque bloc de O_S est conservé l'ensemble des ancres appartenant au bloc, ce qui facilitera lors de la recherche des blocs à aligner, l'identification des blocs partageant le plus d'ancres.

Identification des blocs à aligner

La conservation des ancres introduites dans chaque bloc facilite la phase de calcul des paires de blocs partageant le plus d'ancres, un bloc de O_S ne s'alignant qu'avec un seul bloc de O_T (cf. Algorithm APP_Align). Une fois identifiés les blocs de O_S devant être alignés avec le même bloc de O_T, il est possible de les fusionner si leur taille le permet, afin de diminuer le nombre de combinaisons à faire lors du processus d'alignement.

Algorithm APP_Align(S, T, P)

Require : $C = B_{T1}, B_{T2}, ...B_{Tn}$ l'ensemble des blocs de l'ontologie cible O_T .
Require : $S = B_{S1}, B_{S2}, ...B_{Sn}$ l'ensemble des blocs de l'ontologie source O_S .
1 : for each bloc B_{Ci} dans T do
2 : initialiser $P_i = \varnothing$
3 : for each bloc B_{Sj} dans S do
4 : if B_{Ti} est le bloc avec lequel B_{Sj} partage le plus d'ancres then
5 : $P_i = P_i \cup B_{Sj}$
6 : Supprimer B_{Sj} de S
7 : end if
8 : end for
9 : end for
10 : return $P = \{(P_1, B_{T1}), (P_2, B_{T2})...(P_m, B_{Tm})\}$ où les P_i représentent l'ensemble des blocs B_{Sj} devant être alignés avec un bloc B_{Ti}.

Exemple :

Fig. 3.10 et Fig. 3.11 présentent également des résultats obtenus sur l'exemple précédent présenté en Fig. 3.4.
Fig. 3.10 montre les blocs construits à partir de O_T selon la méthode APP, favorisant le regroupement en tenant compte des ancres.

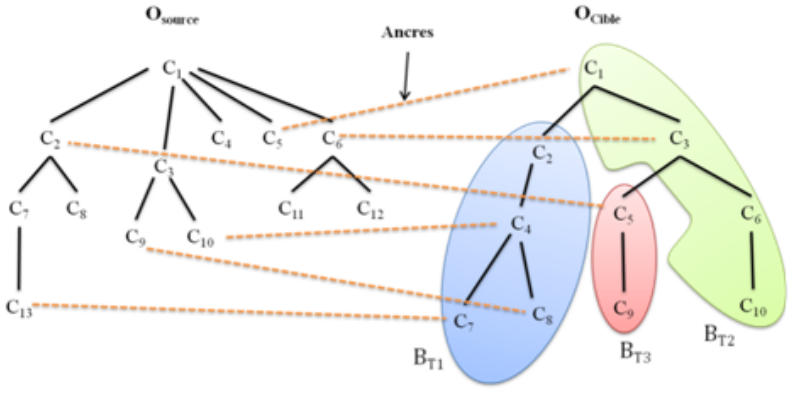

FIGURE 3.10 – *Les blocs construits à partir de* O_j

Fig. 3.11 montre les blocs construits à partir de O_S, favorisant la constitution de blocs partageant des ancres avec ceux de O_T tout en prenant en compte la structure de O_S .

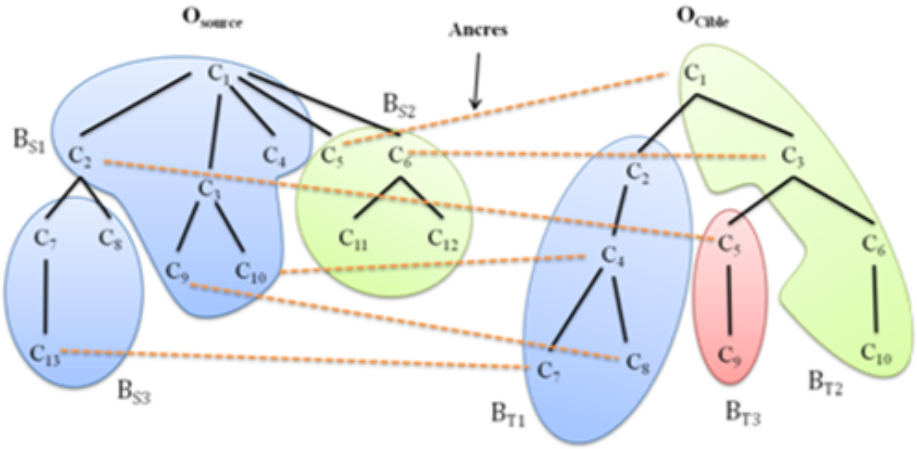

FIGURE 3.11 – *Génération des blocs de la source*

L'alignement s'effectue entre les blocs qui partagent le plus d'ancres : B_{S2} et B_{S3} sont alignés avec B_{T1} et B_{S1} avec B_{T2}. B_{T3} ne participe pas à l'alignement puisque qu'il ne contient qu'une ancre partagée avec B_{S2}, qui partage davantage d'ancres avec B_{T2} qu'avec B_{T3}. Ici est perdu l'appariement d'une ancre, (c_2, c_5), mais les temps d'alignement sont réduits et des blocs sont construits par co-clustering en prenant plus en considération les relations partagées entre les deux ontologies.

Méthodes de partitionnement à base de Clustering

Dans cette section nous passons en revue d'autres approches de partitionnement d'ontologie à base de clustering. Pour ce faire nous faisons tout d'abord un survol sur les concepts de base du clustering ensuite nous exposons quelques approches, car dans le domaine de partitionnement des ontologies OWL, peu de travaux existent.

La problématique générale du regroupement (ou clustering) consiste à organiser un ensemble d'objets en groupes de façon à ce que deux objets similaires se retrouvent dans un même groupe et deux objets dissimilaires dans des groupes distincts. De nombreuses stratégies ont été proposées pour répondre à cette problématique, comme par exemple les approches par partitionnement (k-moyennes), les algorithmes hiérarchiques (agglomératifs ou divisifs), les méthodes utilisant des mélanges de densités de probabilité, des découpages en grilles, etc.

Donc le Clustering consiste à créer une partition ou une décomposition de cet ensemble en sous parties (clusters) telle que :

- Les données appartenant au même groupe se ressemblent,
- Les données appartenant à deux groupes différents soient peu ressemblantes.

Mathématiquement, on a un ensemble X de N données décrites chacune par leurs P attributs.

Exemple : On utilise souvent le clustering en traitement d'images pour fixer les divers objets qu'elles contiennent (segmentation) : routes, villes, rues, des organes humains (pour les images médicales).

Exemples d'applications :
— Classification de documents
— Regroupement de documents par leur contenu
— k dimensions : nombre d'occurrences d'un mot dans un document (k mots en tout → beaucoup de dimensions)

Différentes Méthodes de Clustering

Il existe une trentaine de méthodes et leurs variations incluent dans différentes familles. Nous présentons les principales familles de méthodes de regroupement des données en clusters. Les méthodes peuvent être séparées en quatre groupes :

(a) **Les méthodes basées sur une distance :**

Ces méthodes se basent sur la notion de distance entre objets du jeu de données, en posant que si deux objets sont proches suivant cette distance, ils doivent être regroupés ensemble dans un même cluster. Les algorithmes K-means [Karol and Mangat, 2013] et Fuzzy-c-means [Rtk and al., 1995] sont les algorithmes les plus connus de cette famille d'algorithme. Ces méthodes permettent de trouver des formes de clusters convexes et sont très utilisées notamment à cause de leur coût d'exécution faible.

(b) **Les méthodes basées sur une grille :**

Leur processus consiste à regrouper les cellules denses les plus proches. Ces méthodes ont été proposées pour réduire l'explosion combinatoire des méthodes à base de densité qui fait suite à l'augmentation du nombre d'objets.

(c) **Les méthodes probabilistes :**

Ces méthodes supposent que les données suivent une certaine loi de probabilité. L'objectif est d'estimer les paramètres de cette loi et de définir un modèle de mélange de lois pour représenter les différents clusters.

(d) **Les méthodes hiérarchiques :**

Ces méthodes construisent une hiérarchie de clusters. Chaque nœud contient ses clusters enfants, et les nœuds frères partitionnent les objets contenus dans leurs parents. Ce type d'approche permet d'explorer les données à différents niveaux de granularité. Les méthodes de clustering hiérarchique sont décomposées en deux types d'approches, les approches ascendantes où l'algorithme part d'un grand nombre de clusters et ceux-ci sont

ensuite fusionnés jusqu'à n'obtenir plus qu'un unique groupe contenant tous les objets du jeu de données et les approches descendantes qui partent, de l'ensemble des données, et le divisent en clusters qui sont ensuite divisés récursivement.

3.2.6 Autres Travaux dans le domaine de partitionnement des ontologies à base de Clustering

Parmi les travaux dans le domaine de partitionnement à base de clustering on peut citer le travail de [Saruladha and al., 2012a] : dans ce travail un algorithme de partitionnement est appliqué pour la décomposition d'une vaste ontologie. C'est une révision de l'algorithme d'AHSCAN. (Agglomerative Hierarchical Structure Clustering Algorithm for Networks) qui est un algorithme de clustering avec une complexité de O(n). Dans cet algorithme un concept (V) est considéré comme (neighbor) d'un autre concept (W) s'il existe un lien entre eux. V est le subsumeur de W. Le concept V est le descendant ou le fils du concept W. Deux concepts reliés qui n'ont pas de nœud (edge) subsumeur entre eux ne sont pas considérés comme ascendant ou descendant. Dans cet algorithme un nouveau paramètre est introduit (paramètre de mesure de la qualité) afin de trouver la meilleure partition.

Dans les travaux de [Kolli, 2008] l'ontologie est représentée sous forme d'un graphe. L'algorithme de clustering est traversé en commençant par la racine (R) et assemblant MB nombres de nœuds avec un ensemble $(2 * MB)$ nombre de nœuds. L'objectif de cette approche comme indiqué par les auteurs est juste pour diviser l'ontologie afin de permettre son traitement dans la réalité.

Dans les travaux de Hu et al [Hu and al.,2006b] l'algorithme de partitionnement est réalisé sur un graphe construit en se basant sur les dépendances de la hiérarchie de classes. Dans cette approche, un poids est associé à ces dépendances en utilisant des informations linguistiques et structurelles des entités. Ensuite l'algorithme de Rock [Gómez-Pérez, 1999] est appliqué (c'est un algorithme agglomératif de clustering) pour partitionner le graphe. Dans l'étape finale des partitions sont générées et chacune est appelée bloc.

Dans l'étude réalisée par [Huang and Lai, 2006] pour partitionner une ontologie qui est représentée sous forme d'un graphe, les auteurs génèrent une matrice d'incidence. Ici la valeur de mesure de similarité entre deux entités est partiellement déterminée par le nombre des nœuds commun entre eux.

Dans la phase de partitionnement, cette approche utilise l'algorithme KNN (Knearest neighboard algorithm). Après cette phase les partitions générées avec similarités supérieurs sont alignées ensemble.

L'approche introduite par [Kunger and Puraji, 2009] est une revue sur les techniques de partitionnement des graphes en appliquant la notion de similarité sur une ontologie représentée sous forme d'un graphe. En exploitant les relations entre les concepts d'une ontologie dans le domaine (PHC) (preventive health care domain), des tables d'attributs et de concepts ainsi qu'une autre table de relations (liens) entre concepts sont générés pour implémenter l'algorithme CBRO

(clustering based relational ontology) un algorithme de clustering à base de relations ontologiques.

Les données sont partitionnées au niveau des concepts ensuite en fonction des relations ontologiques existantes, d'autres nouvelles relations seront dérivées entre deux concepts.L'algorithme est appliqué dans le domaine de PHC.

Trouver les correspondances sémantiques à travers des attributs multiples est plus important dans plusieurs applications telles que (l'intégration de schéma). Dans les travaux de [Ding and al., 2013] les auteurs utilisent l'algorithme de K-means pour réaliser l'alignement entre plusieurs attributs.

La méthode de K-Means ou algorithme de partitionnement par centre mobile permet d'effectuer un partitionnement d'un ensemble de données en K clusters. Un cluster regroupe plusieurs concepts similaires. Chaque cluster (partition) est décrit par son centre. Les centres des clusters sont mobiles au cours de l'exécution de l'algorithme.

L'algorithme peut se présenter comme suit :

Le nombre de clusters, le paramètre K, est fournit au départ.

Un ensemble de K centres choisi dans l'ensemble des données,

Les K clusters sont formés en regroupant dans chaque centre l'ensemble des données plus proches du centre courant que de tout autre centre,

Le centre de chaque cluster est calculé et devient le nouveau centre,

L'algorithme boucle alors sur l'étape précédente : Les données sont réaffectées en fonction de ces nouveaux centres et la condition d'arrêt est que les centres deviennent immobiles.

En premier lieu, les auteurs convertissent 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.

Le travail de [Jiang and al.,2006] propose une méthode de clustering d'une ontologie par la définition d'une nouvelle mesure pour calculer la similarité. L'arbre d'agrégation créé a une forte sémantique.

Cet algorithme combine la similitude de l'ontologie avec la valeur de l'objet et décide quel objet doit être remplacé. L'arbre d'agrégation est créé entièrement différemment selon les cas d'application.

3.3 Etat de l'art sur la stratégie de modularisation des ontologies

3.3.1 Introduction

Les ontologies doivent leur succès actuel à leur aptitude à être partageables et réutilisables. Quoi qu'avec le partage, on se retrouve avec des ontologies de plus en plus larges. La construction d'une ontologie à partir de zéro est une tâche complexe qui nécessite du temps. Ceci nous amène donc à envisager la modularisation comme étant une approche qui apporterait des solutions non seulement au problème de conception, mais aussi à celui du partage et de l'intégration des ontologies.
La notion de modularisation est basée sur le principe de «diviser pour mieux régner» généralement appliquée dans le génie logiciel où il est question de développer une application dont la structure repose essentiellement sur des composants autonomes facilement concevables et réutilisables.
Elle a été proposée par [Wang and al.,2006] pour traiter les ontologies volumineuses et complexes. Les auteurs ont proposé une approche basée sur la modularisation (MOM). Elle se base sur le principe de « diviser pour régner » qui décompose le problème du matching à large échelle en des sous petits problèmes en réalisant le matching au niveau des modules d'ontologies. Cette stratégie inclut des sous étapes telles que la détermination des modules similaires, le matching des modules et la combinaison des résultats. Cette méthode utilise les ϵ-connection [Grau and al.,2005] pour transformer une ontologie en entrée en une ϵ-connection avec le plus grand nombre possible de bases de connaissances connectées et un algorithme de partitionnement [Grau and al.,2005] qui permet de décomposer les ontologies e-connectées en des modules.
Cette section présente un état de l'art sur la modularisation des ontologies,
les concepts de la modularisation et les travaux que nous jugeons les plus importants réalisés dans ce domaine ainsi que les objectifs de la modularisation des ontologies. Nous partons d'une étude générale sur la modularisation, et de ses objectifs [Setti Ahmed and Benslimane, 2012].

3.3.2 Les objectifs de la modularisation des ontologies

La modularisation est un concept qui nous renvoie simultanément à deux aspects différents de l'ontologie à savoir : «l'ontologie comme un tout» et «l'ontologie comme un ensemble de modules». Les différentes approches de la modularisation et la façon dont elles sont mises en pratique dépendent en grande partie du but visé. Ainsi, le but de la modularisation est d'apporter des solutions aux problèmes suivants [Parent and Spaccapietra,2009] :

i L'extensibilité en fonction des requêtes et du raisonnement sur les ontologies.Les raisonneurs sont plus efficaces sur des ontologies de petite taille.
La performance de ceux-ci décroît au fur et à mesure que la taille de l'ontologie croît. Travailler avec des petites ontologies permet de pallier à la perte

de performance des moteurs d'inférences, et la modularisation apporterait une solution à ce problème, dans la mesure où elle fragmenterait une ontologie qui tendrait à gagner en taille en un ensemble d'ontologies plus petites. Ainsi, des requêtes peuvent être faites uniquement sur un module précis sans pour autant avoir à explorer l'ontologie en entier.

ii L'extensibilité en fonction de la maintenance et de l'évolution des ontologies. A ce niveau, le but de la modularisation serait de localiser et contenir l'impact que pourrait avoir la mise à jour d'une base de connaissances (knowledge repository) dans les limites des modules. L'implémentation d'une bonne distribution Une méthode qui combine les langages logiques et qui permet de partitionner des bases de connaissances en des sous-parties exprimables dans des formalismes décidables des connaissances nécessite, dans un premier temps, une bonne compréhension du principe de propagation de la mise à jour à l'intérieur même de l'ontologie et, en second lieu, une bonne connaissance de l'information contenue dans cette mise à jour.

iii La complexité dans la gestion des ontologies. L'on ne saurait garantir l'efficacité et la précision des bases de connaissances de grande taille élaborées par des humains, aussi bien en termes d'objets et de relations entre objets qu'en termes d'axiomes et de règles. Les ontologies qui contiennent des milliers de concepts ne peuvent être conçues et gérées par un seul expert. D'où la nécessité de ramener la conception des ontologies à la création de différents modules manipulables et de taille raisonnable, qui pourront par la suite être liés les uns aux autres.

iv La compréhensibilité du contenu des ontologies. Les utilisateurs doivent pouvoir être à même de comprendre le contenu d'une ontologie afin de pouvoir la manipuler. Que ce contenu soit représenté sous une forme graphique ou textuelle, il sera plus aisé pour des humains de le lire si l'ontologie est de petite taille. Toutefois, la compréhensibilité ne dépend pas uniquement de la taille de l'ontologie mais aussi de la façon dont les objets, relations et règles sont agencés les uns par rapport aux autres.

3.3.3 La réutilisation des ontologies

D'un point de vu ontologique, la modularisation est considérée comme un moyen pour structurer et pour organiser des ontologies. Le module lui même est défini comme un sous-ensemble d'un tout qui a du sens, [Doran and al., 2007] définit le module de l'ontologie comme un composant réutilisable d'une ontologie plus complexe et affirment que le module de l'ontologie est autonome mais doit toujours entretenir des relations avec d'autres modules (y compris l'original).

La réutilisation est vue donc comme l'une des motivations premières de la modularisation. Elle permet de mettre l'accent sur les mécanismes de construction de modules ontologiques de telle sorte que ceux-ci puissent être réutilisés par la suite. Il va de soi que le contenu de ces modules se doit d'être pertinent, néces-

saire et compréhensible afin que ces modules soient sélectionnés pour réutilisation.

Nous partageons le point de vue de Furst [Furst,2004] selon lequel les ontologies sont destinées à être réutilisées. La sémantique qu'elles représentent est liée au cadre applicatif à partir duquel le sens des termes et concepts est défini.

Cependant, la représentation ne dépend pas de l'opération faite avec l'ontologie. La sémantique de l'ontologie est liée au contexte mais la représentation n'implique pas que l'ontologie soit utilisée uniquement dans le contexte de sa création.

Définition et description d'un module

Un module est un sous-ensemble d'une ontologie qui a un « sens », du point de vue des applications ou des utilisateurs. On ne construira pas un module en y incluant des classes de façon aléatoire. On peut supposer qu'une classe sans aucun lien avec les autres classes du module n'est pas désirable. Mais plus concrètement on peut définir un module ayant un sens si [Pinto and al., 2000] : il est valide localement : si chaque information valide dans le fragment l'est encore dans la globalité de l'ontologie.

Il est localement complet : chaque information valide dans le domaine du fragment de l'ontologie globale l'est encore dans le fragment.

Un module devrait aussi être : petit de façon à raisonner plus vite sur le module.

Le plus indépendant que possible des autres modules. L'ajout ou le retrait d'un module n'affectera pas beaucoup les autres. On caractérise donc un module comme fermé s'il ne possède aucun lien avec l'extérieur et d'ouvert s'il contient des liens avec d'autres modules.Il est Compréhensible pour toute autre personne voulant le réutiliser.

Il existe deux approches de création d'un module [Pinto and al., 1999] :

(a) **Composition :** les modules sont construits indépendamment les uns des autres. Ils sont ensuite assemblés pour former une ontologie plus large.

(b) **Décomposition :** on crée un module en partitionnant une ontologie déjà existante. Le but de cette approche est de pouvoir y arriver de façon semi-automatique. La validation est faite par l'administrateur.

Critères de modularité

L'approche de composition est similaire à l'intégration d'une ontologie. Mais dans une ontologie modulaire, on peut solutionner le problème de la duplication d'information en liant les redondances d'un point de vue conceptuel par des inter-modules qui diminuent la taille des modules [Benoit ,2005].

Selon [Pinto and al., 1999], ils considèrent que dans l'approche de décomposition on peut se baser sur des experts du domaine pour définir les modules, mais ce n'est pas suffisant, ce qui nécessite de déterminer aussi des critères de décomposition semi-automatiques qui seraient validés par un expert. On pourrait

décomposer une ontologie en déterminant un noyau de classes devant faire partie du module, pour ensuite identifier et inclure les classes rattachées à ce noyau. Dans cette méthode il faut pouvoir analyser les requêtes auxquelles doivent répondre l'ontologie et sauvegarder le chemin utilisé pour y répondre. Une autre possibilité est de décomposer une ontologie basée sur des algorithmes de décomposition de graphe pour obtenir des sous -graphes.
Il faut tout de même faire attention dans l'approche de décomposition, de ne pas perdre d'informations. Une requête doit fournir le même résultat avant et après décomposition. Mais que se passe-t-il si la perte d'information est inévitable ?

Relations inter-modules

Si on considère les modules comme des sous-ontologies indépendantes, il est peu probable qu'elles puissent fournir une réponse complète à une requête. Il faut déterminer les modules nécessaires à la requête et ensuite fusionner et synchroniser les réponses partielles pour avoir une réponse globale. On considère deux cas [Pinto and al., 1999] :

— Si les modules sont fermés. Les méta-données peuvent être centralisées dans un « repository », contenant les informations contenues dans le module et la façon pour récupérer ces données.
— Si les modules sont ouverts, les liens inter-module peuvent être procéduraux, ce type de lien peut être comparé à un mécanisme de vue. On extrait la connaissance d'un module sur la base d'une requête. Les liens peuvent aussi être des relations par assertion, ils établissent un degré de ressemblance entre les composants de plusieurs modules et peuvent donc trouver plus d'éléments concernant une information dans un autre module [Benoit ,2005].

3.3.4 Les principales approches de modularisation des ontologies

Une ontologie O est définie par la formule suivante [Palmiz et al., 2009] : $O = (Ax(O), Sig(O))$ où $Ax(O)$ représente l'ensemble d'axiomes composé de concepts, de relations et de fonctions (sous-classe, équivalence, inst anciation, etc ...). $Sig(O)$ est la signature de O qui représente l'ensemble de noms des entités qui se retrouvent dans les axiomes. En d'autres termes, il s'agit du vocabulaire de O. La modularisation de l'ontologie O permet de définir un module tel que :

$M = (Ax(M), Sig(M)$ où M fait partir de O, avec $M \subseteq (Ax(O))^I)$ et
$Sig(M) \subseteq Sig(O)$

L'on ne saurait définir l'ontologie sans pour autant introduire la notion d'interprétation I qui est une base de la sémantique de la logique descriptive.
On la symbolise par $I = (\Delta^I, \bullet^I)$ où Δ^I est le domaine d'interprétation et $\bullet^I$ la fonction d'interprétation.

Les techniques d'extraction de module

L'extraction de module ontologique est un processus au cours duquel un module couvrant une signature spécifique est extrait d'une ontologie O, telle que $Sig(M) \subseteq Sig(O)$. M est la partie pertinente de O qui couvre l'ensemble d'éléments définis par $Sig(M)$. Le module est une ontologie en soi, et comme tel, d'autres modules peuvent être extraits à partir de lui. Formellement, l'extraction du module peut être définie comme suit :

L'extraction est basée sur le parcours de graphe

Les techniques d'extraction basées sur une approche structurelle nécessitent la représentation de l'ontologie comme un graphe afin de parcourir celle-ci et d'en extraire le module ontologique. Parmi ces approches nous distinguons :

L'approche de d'Aquin et al.

D'Aquin et al. [D'Aquin and al.,2006] considère que la sélection des connaissances»(Knowledge selection) est un processus plus complexe et l'extraction de module ontologique ce n'est qu'une étape de ce processus. Ce dernier a comme objectif de récupérer les composants pertinents des ontologies disponibles en ligne pour l'annotation automatique de la page Web retournée par le navigateur. Cette approche a été implémentée au sein d'un outil appelée Magpie, qui est un plugin pour navigateur.

Magpie permet d'identifier les instances de classes d'ontologies sur une page Web en surlignant chacune d'elles avec une couleur qui lui est associée.

Magpie permet aussi d'extraire les parties les plus utiles et pertinentes des ontologies qui décrivent les classes des instances sur une page Web.

Le processus de sélection de connaissances se fait en trois étapes (Figure 3.12) :

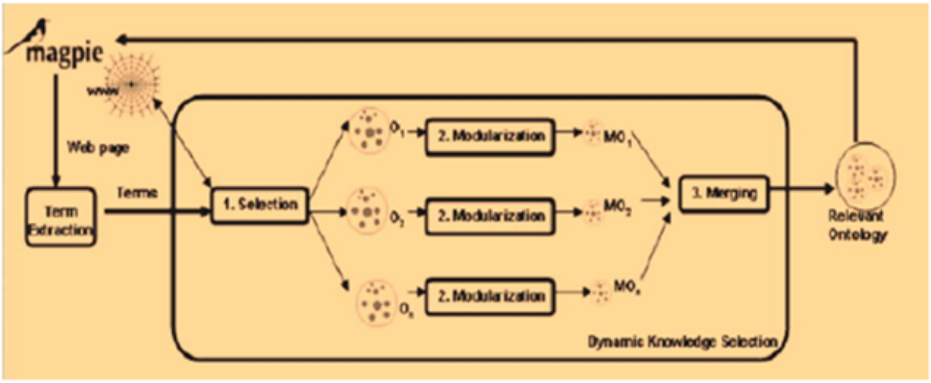

FIGURE 3.12 – *Le processus de sélection de connaissances et son utilisation avec Magpie [D'Aquin and al.,2006].*

i La sélection des ontologies pertinentes. Tout d'abord, on introduit un ensemble de termes pour lesquels on recherche une ontologie. Ensuite, on identifie les ontologies ou les ensembles d'ontologies qui couvrent les termes qui

ont été introduits. Par couverture, on entend que l'ensemble des ontologies contenant des concepts, propriétés, ou instances qui sont sémantiquement liés aux termes donnés doit être retourné par l'algorithme.

ii La modularisation des ontologies sélectionnées. La technique de partitionnement de Stuckenschmidt et Klein est appliquée sur les ontologies retournées lors de l'étape précédente afin d'identifier les différentes sous-ontologies qui contiennent une connaissance suffisamment pertinente pour la tâche à réaliser [Stuckenschmidt and Klein, 2004]. Le principe d'extraction de cette approche est de rajouter au fur et à mesure dans les modules tous les éléments participant directement ou indirectement à la définition des entités déjà incluses dans chacun de ceux-ci. En d'autres termes, si un concept impliqué dans une définition n'est pas encore inclus dans l'un des modules, alors celui-ci y est systématiquement rajouté [Patrick, 2014].

iii La fusion des modules ontologiques pertinents. Dans la mesure où plusieurs ontologies seraient retournées après la phase de sélection, les modules extraits individuellement de celles-ci sont fusionnés afin de produire une ontologie unique, résultat final du processus de sélection de connaissances. D'Aquin et al. ne fournit aucun détails sur le déroulement du processus de fusion des modules.

L'approche de Doran et al.

L'approche de Doran et al. a comme perspective de pouvoir réutiliser une partie d'une ontologie existante. Elle ne prend pas en considération le type de langage dans lequel l'ontologie est représentée, dès lors que celui-ci est transformable en un modèle de graphe abstrait [Doran and al., 2007].

Ce graphe abstrait de Doran et al. est un graphe orienté et étiqueté C avec une alphabet donnée E, tel que $C = (V, E)$ où :

- V est un ensemble fini de sommets.
- $E \subseteq V * \sum_E * V$ est une relation ternaire décrivant les arêtes.

La particularité de l'approche de Doran et al. est que dans ce graphe il ya un ensemble d'arêtes à parcourir et un autre à ne pas parcourir, du moins lors de la première itération de l'algorithme.

Par exemple, lorsqu'on extrait un module d'une ontologie OWL-DL, les arêtes owl :disjointWith ne sont pas parcourues à la première itération, elles le sont uniquement lors des itérations suivantes. Les arêtes sont étiquetées en fonction du langage de l'ontologie.

En utilisant le modèle abstrait de Doran et al., il serait donc possible de définir un module ontologique GM tel que $GM = (VM, EM)$, et $VM \subseteq V \wedge VM \neq \emptyset$ et $EM \subseteq E$. Ce qui implique que $GM \subseteq G$.

Cette approche permettrait de ramener le problème d'extraction de module à un parcours de graphe avec pour point de départ un sommet x tel que $x \in VG$. La

seule condition est que les arêtes étiquetées « disjoint» du sommet x ne soient pas parcourues à la première itération. Deux concepts sont disjoints s'ils n'ont aucun ancêtre commun. Toutefois, Doran et al. supposent que si l'utilisateur souhaite inclure les concepts disjoints dans le module lors de la première itération, alors il devra tout simplement choisir pour point de départ leur ancêtre commun.

Le module extrait est un graphe représentant respectivement les ensembles de sommets et d'arêtes. Le module ontologique généré par cette approche reste transitivement fermé et ce, même en tenant compte des différentes relations parcourues. Afin de garantir cette fermeture transitive, l'ontologie dont on veut extraire le module se doit elle aussi d'être transitivement fermée.

La notion de fermeture transitive ici suggère que toute relation existante entre concepts est considérée, et ce même si cette relation lie un concept du module à un concept intermédiaire (concept qui n'appartient pas au module mais qui est lié à un autre appartenant au module). Il convient tout de même de noter que les concepts hiérarchiquement supérieurs au concept à partir duquel le processus commence ne sont pas parcourus ; il revient à l'utilisateur de définir le concept de départ et selon son choix, s'il souhaite inclure des concepts disjoints dans le module lors de la première itération. En considérant les concepts plus «généraux» lors du processus d'extraction, on risquerait de se retrouver avec un module dont les proportions sont équivalentes à celles de l'ontologie à modulariser [Doran and al., 2007].

L'avantage de cette approche est que le modèle de graphe abstrait est applicable à toutes les ontologies, quelque soit leur langage.

L'approche de Noy et Musen

Noy et Musen dans leur approche considèrent un module ontologique comme étant une vue d'ontologie, en se basant sur le principe d'encapsulation. Comme la notion de vue issue du domaine des bases de données. [Noy and Musen, 2004].

Ce module représente une sous-structure qui résulte comme réponse à une requête de l'utilisateur sur l'ontologie initiale. Les composants de la requête sont la liste des termes qui représente en fait la signature du module à obtenir.

Le point de départ du processus d'extraction doit être fixé par l'utilisateur en désignant un concept dont les relations seront parcourues récursivement afin d'inclure l'ensemble des entités qui lui sont liées.

Aussi, les relations à parcourir sont sélectionnées par l'utilisateur qui attribue à chacune de celles-ci une profondeur de parcours : il s'agit de la directive de parcours (traversal directive) TD.

Lorsque l'algorithme arrête de parcourir la relation sélectionnée une fois la profondeur est atteinte. La directive de parcours D d'une ontologie O est définie par la paire (C_{st}, PT) où :

- C_{st} est le concept de départ du parcours.

- PT est un ensemble de directives de propriétés (property directives).

Chaque directive de propriété est une paire $< P, n >$ où P est une propriété de O et n est un entier non-négatif ou infini qui définit la profondeur de parcours de la propriété P. Si $n = \infty$, alors le parcours inclura aussi une couverture transitive pour P à partir de C_{st}.
Noy et Musen définissent une spécification de la vue de parcours (traversal view specification) T comme étant un ensemble de directives de parcours. Soit une ontologie O et une spécification de la vue de parcours T constitué d'un ensemble de directives de parcours TD. Le résultat d'une spécification de la vue de parcours $TV(O, T)$ est aussi une vue de parcours qui en fait, représente l'union de tous les résultats des directives de parcours D , tel que $D \in TD$. On rappellera qu'une vue de parcours contient toutes les classes et instances rencontrées tout le long du parcours.
La technique de Noy et Musen a été implémentée et incorporée dans l'outil PROMPT, qui est un plugin pour l'éditeur d'ontologie Protégé et qui donne à l'utilisateur la possibilité de gérer plusieurs ontologies et ce, en lui permettant de comparer les versions, de fusionner et d'extraire les modules ontologiques [Noy and Musen, 2003].
Cette approche est flexible et permet à un ingénieur de connaissances de construire un module de façon itérative, mais pour ce faire, il doit avoir des connaissances approfondies sur l'ontologie à manipuler afin d'être capable de définir lui même des directives de parcours appropriées [Noy and Musen, 2004].

L'approche de Seidenberg et Rector

L'approche de Seidenberg et Rector est une technique d'extraction de module ontologique à partir d'une ontologie médicale GALEN [Seidenberg and Rector, 2005]. Soit un concept A de l'ontologie et une signature de M tel que $Sig(M) = A$. Le processus d'extraction se fait en deux phases : (Figure 3.13)
L'ontologie est parcourue vers le haut afin d'inclure toutes les superclasses de A.
La hiérarchie de l'ontologie est parcourue vers le bas dans le but de rajouter les sous-classes de A.
Il est à noter que les classes ayant un ancêtre commun et se trouvant à la même profondeur dans la hiérarchie sont ignorées. Il faut les rajouter dans $Sig(M)$. pour les inclure dans le module. Les restrictions, intersections, unions et toutes les classes équivalentes de celles déjà incluses sont aussi rajoutées au module.
En dernier lieu, les propriétés des classes précédemment sélectionnées sont aussi parcourues vers le haut afin de rajouter d'autres classes au module.
Seidenberg et Rector proposent un algorithme amélioré en introduisant deux notions essentielles qui sont le filtrage de propriétés et l'utilisation de classes frontières, pour pouvoir contrôler les proportions du module dans le cas où on travaillerait avec une ontologie dense et large. Le processus de filtrage de propriétés consiste à supprimer les propriétés choisies par l'utilisateur [Seidenberg and Rector, 2006].

Pour illustrer cette approche [Patrick, 2014], considérons un cas où l'utilisateur ne serait pas intéressé par l'ensemble des maladies modélisé dans l'ontologie GALEN. Celui ci choisira alors d'exclure toutes les propriétés locatives, c'est-à-dire uniquement des propriétés qui lient des maladies à des parties précises du corps humain. Par exemple, l'utilisateur pourrait, afin d'éliminer l'ensemble de maladies juste supprimer toutes les propriétés hasLocation dans des relations similaires à celle-ci : "IschaemicCoronaryHeartDisease hasLocation Heart".

Le filtrage de propriétés passe aussi par la suppression de toutes les restrictions de classes dans lesquelles ces propriétés apparaissent. Toutefois, il arrive fréquemment qu'en supprimant une restriction, la définition de la classe concernée devienne, soit impossible à distinguer, soit équivalente à une autre définition de classe similaire. On assiste ainsi à l'apparition dans l'ontologie de longues séries de classes équivalentes qui, bien que correctes sont impossibles à

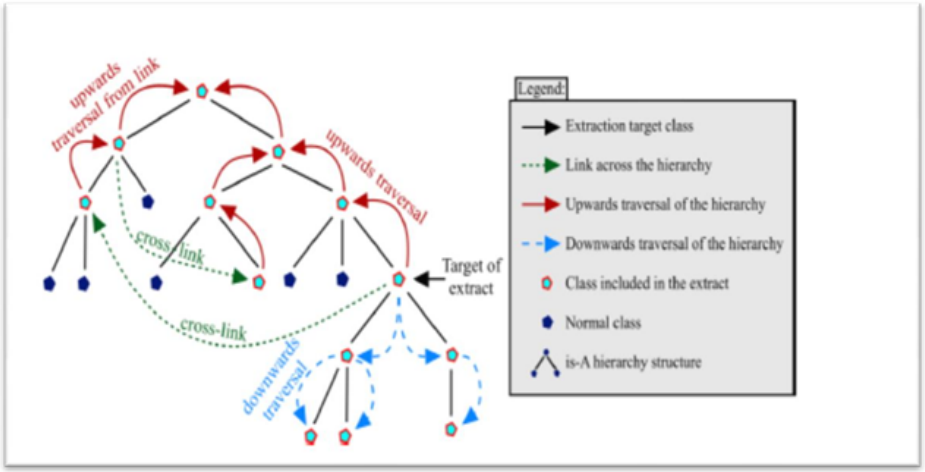

FIGURE 3.13 – *Parcours de la hiérarchie des classes à travers les liens [Seidenberg and Rector, 2005].*

visualiser dans un éditeur d'ontologie tel que Protégé. Pour résoudre ce problème, Seidenberg et Rector proposent une méthode permettant de transformer les classes équivalentes en classes primitives qui conservent leur position dans la hiérarchie, devenant ainsi facile à visualiser dans les éditeurs. L'exemple suivant illustre le filtrage d'une propriété avec le retrait d'une définition :

Lors du processus de filtrage, la restriction présente sur la classe $SkinOf$ Scalp est retirée. La résultante est une classe équivalente, dont la définition est convertie par la suite en classe primitive.

La seconde méthode de délimitation du module proposée par Seidenberg et Rector se fait à l'aide de deux éléments préalablement définis par l'utilisateur à savoir la profondeur de récursion et le concept cible à partir duquel le processus de segmentation débute.

Le parcours des liens à travers l'ontologie en partant du concept cible s'arrête lorsque la classe qui se trouve à la frontière est atteinte :

on parlera alors de classe frontière (boundary class). En effet, une classe frontière est la classe sur laquelle l'algorithme de segmentation s'arrête lorsqu'une certaine profondeur de récursion est atteinte, entraînant ainsi la suppression de

tous les liens de cette classe (Figure 3.14). Il s'agit d'une méthode efficace qui permet de limiter la taille du module dès lors que les classes frontières sont considérées comme étant la condition d'arrêt de l'algorithme.

L'algorithme basique d'extraction de module ontologique de Seidenberg et Rector est implémenté dans l'outil SegmentationApp. C'est une application contenue dans un fichier .jar .
Cet outil prend en entrée deux éléments à savoir l'ontologie à modulariser et le fichier texte contenant la signature du module à extraire. Le module ontologique ainsi généré est automatiquement exporté dans un fichier .owl . L'algorithme basique de Seidenberg et Rector n'est pas paramétrable.

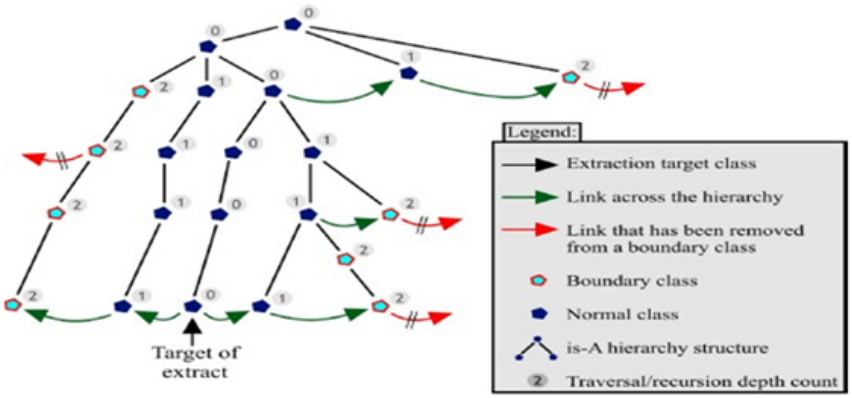

FIGURE 3.14 – *Extraction de segment avec profondeur de* 2 *[Seidenberg and Rector, 2006].*

L'extraction basée sur la logique

En plus des techniques d'extraction basée sur le parcours de graphe, il existe d'autres techniques d'extraction basée sur la logique qui reposent sur la notion d'extension conservatrice (conservative extension). Une ontologie est considérée comme une extension conservatrice si les implications logiques à propos du module sont comprises dans sa signature et un module est une sous-structure de cette ontologie dont il est extrait.
Formellement, et selon Lutz et al. une extension conservatrice est définit comme suit [Lutz and al., 2007] :

Soient τ_1 et τ_2 des T-Box formulées en logique descriptive $\mathcal{L}$ et soit $\Gamma \subseteq Sig(\tau_1)$ une signature. Alors $\tau_1 \cup \tau_2$ est une Γ-extension conservatrice de τ_1, si pour tout $C_1, C_2 \subseteq \mathcal{L}(\Gamma)$, on a $\tau_1 \cup \tau_2 \models C_1 \sqsubseteq C_2$.
Cette définition sous-entend que toutes les implications logiques de la signature d'un module ontologique sont les mêmes que si on fait l'union de ce module et de l'ontologie dont il a été extrait.
Déterminer si une ontologie O est une extension conservatrice est un problème indécidable en langage OWL-DL c'est l'inconvénient avec la définition de Lutz et al.

Pour pallier à ces limites, Grau et al. proposent d'utiliser des contraintes moins strictes : on parlera de modules basés sur la localité (locality-based modules) [Grau and al., 2008]. Grau et al. introduisent également deux nouvelles notions : la couverture (coverage) et la sécurité (safety) qui sont des propriétés garanties par des modules basés sur la localité. Ces propriétés sont définies en des termes d'un module importé par une ontologie locale (L) comme suit [Grau and al., 2007a] :

La couverture garantit que l'ensemble des termes qui se rapportent aux termes spécifiés sera extrait de l'ontologie. Un module O' couvre une ontologie O pour les termes d'une certaine signature Sig dans la mesure où pour toutes les classes

A et B tels que $A, B \in Sig(O')$, si $L \cup O \models A \sqsubseteq B$ alors $L \cup O' \models A \sqsubseteq B$.

La sécurité garantit que la signification des termes extraits ne sera pas modifiée. L utiliserait les termes de la signature Sig de façon sécuritaire dans la mesure où pour toutes les classes

A et B tels que $A, B \in Sig(O')$, si $L \cup O' \models A \sqsubseteq B$ alors $O' \models A \sqsubseteq B$.

Grau et al. décrivent aussi deux types de localité : la localité syntaxique qui peut être calculée en temps polynomial, et la localité sémantique dont le calcul est un problème PSPACE-complet [Grau and al.,2009]. À la différence de la localité syntaxique qui est calculée à partir de la structure syntaxique des axiomes, la localité sémantique repose uniquement sur l'interprétation (I) de l'axiome.

Jimènez et al. quant à eux proposent deux conditions différentes de localité afin d'extraire les modules ontologiques [Jiménez and al., 2008]
La $\bot$-localité qui extrait un module approprié pour un raffinement et celui-ci contient tous les super-concepts de la signature.
La $\top$-localité qui extrait un module approprié pour une généralisation et qui contient tous les sous-concepts de la signature.
L'approche d'extraction basée sur la logique de Grau et al. est implémentée dans l'outil OWL Module Extractor. C'est une application Web développée par Rafael Gonçalves dans le cadre d'un projet en ingénierie de connaissances à l'université de Manchester (Royaume- Uni).
OWL Module Extractor prend en entrée deux paramètres : l'URI ou le contenu de l'ontologie à modulariser et la signature qui représente la liste de termes à couvrir par le module à extraire.

Comparaison

Dans cette section nous présentons une étude comparative entre les approches d'extraction de modules basées sur le parcours de graphe Tableau 3.1 et les

approches d'extraction de module basées sur la logique selon un certain nombre de critères [Setti Ahmed and al., 2011].

Approaches	Coverage	Minimality	DL EXP	Tractable
Whole Ontology	+	-	Any	+
LocalityBased	+	-	Owl	+
MEX	+	+	EL++	+
Conservative	+	+	Any	-

TABLE 3.1 – *comparaison des approches d'extraction de modules basée sur le parcours de graphe.*

Approaches	Interactive	Traversal direction	Property filtring	Least common subsumer	Assume inferred Model
Whole Ontology	-	Up & Down	-	-	-
D'Aquin et al	-	Up & Down	-	+	+
Doran et al	-	Down	-	-	-
Noy and Musen	+	Up & Down	-	-	-
Seidenberg and Rector	-	Up	+	-	-

TABLE 3.2 – *comparaison des approches d'extraction de modules basée sur la logique.*

L'extraction basée sur SPARQL

Les approches proposées par Borgida et Giunchiglia ainsi que d'Aquin et al. nécessitent que l'utilisateur doit avoir des connaissances sur les formalismes non standards [Borgida and Giunchiglia, 2007], [D'Aquin and al., 2007].

En revanche le travail de Doran et al. utilise RDF et SPARQL comme base pour un framework commun d'extraction de module, car tous deux sont des standards W3C [Doran and al., 2008].

Toutes les ontologies OWL peuvent être représentées en un graphe RDF et SPARQL est le langage permet tout particulièrement, l'interrogation de descriptions RDF. A l'image de RDQL (dont il est une extension), SquishQL ou TriQL, SPARQL exploite, en premier lieu, RDF au travers de la notion de triplets ou d'ensembles de triplets. La réponse à la requête posée va ainsi correspondre à la restitution du sous-graphe RDF satisfaisant le filtre exprimé au travers d'opérations de mise en correspondance de patterns de graphe (possiblement optionnels) et d'opérations basées sur les connecteurs logiques (conjonction, disjonction).

Doran et al. présentent le framework SOMET (Figure 3.15) et montrent ainsi qu'il est possible de classer les approches d'extraction de module basées sur le parcours comme étant une série de requêtes SPARQL sur un graphe RDF.

Le framework SOMET contient donc des représentations SPARQL des différentes techniques d'extraction de module par parcours de graphe. Cet outil permet à l'utilisateur d'ajouter, de modifier ou de retirer des requêtes SPARQL de l'ensemble de requêtes qui doivent passer au moteur d'extraction (Traversal Extraction Engine). Ce qui lui juge flexible. Étant donné que les différentes approches d'extraction sont assimilées à un ensemble de requêtes SPARQL, ces ensembles ne sont pas disjoints et leurs intersections permettent ainsi de mettre en évidence les points communs entre les différentes techniques.

SOMET utilise un algorithme de parcours de graphe dont les paramètres sont : l'ontologie dont on veut extraire le module, la signature qui décrit le module et un ensemble de requêtes SPARQL. Il s'agit d'un algorithme itératif qui, dans un premier temps, applique des requêtes sur les éléments de la signature afin de construire le module désiré, et dans un deuxième temps de nouvelles requêtes sont appliquées sur chacun des éléments retournés par les requêtes précédentes. Ainsi de nouveaux éléments sont rajoutés au module à chaque itération [Doran et al., 2008].Dans le cas de l'approche de Doran et al. , dont le processus d'extraction se fait à partir d'un concept unique, on distingue les requêtes SPARQL suivantes :

DESCRIBE ? *c* : cette requête permet de décrire le concept sur lequel elle est appliquée et ce, en lisant toutes les déclarations (statements) dans lesquelles le concept ? *c* apparaît comme étant le sujet.
CONSTRUCT { ? y rdf s : domain ? c} WHERE { ? y rdf s : domain ? c}
: retourne toutes propriétés ?y dont le concept ? c est l'espace de départ (domain) de la propriété.
DESCRIBE ? y WHERE { ? y rdf s subClassOf ? c} : retourne toutes les sous-classes de ? c
CONSTRUCT { ?y owl : equi valentClass ? c}
WHERE { ?y owl : equivalentClass ? c} : retourne tous les concepts ?y
où ?y est une classe équivalente à ?c.

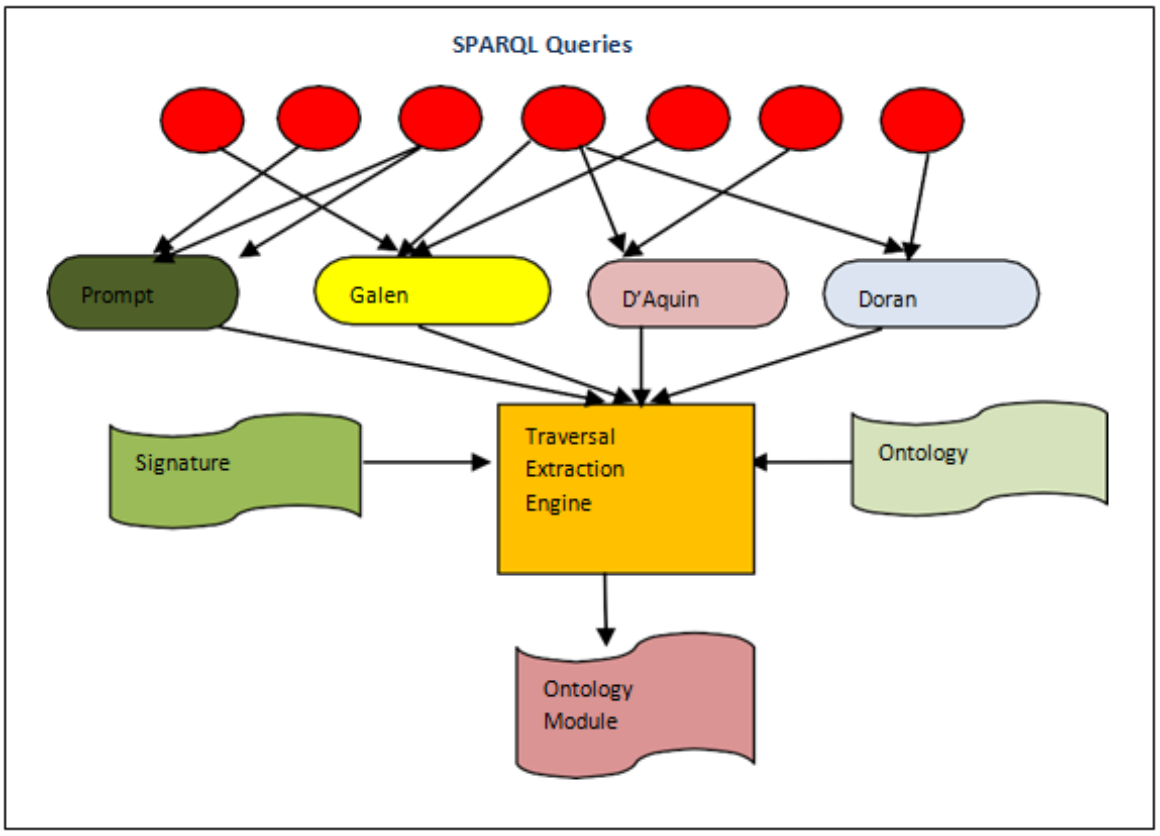

FIGURE 3.15 – *framework SOMET [Doran and al., 2008].*

3.4 Travaux connexes

Les systèmes basés sur les cadres existants fonctionnent bien lorsqu'il s'agit de petites ontologies, cependant, plusieurs limitations apparaissent lorsqu'il s'agit de grandes ontologies. Les outils d'alignement ontologique perdent de leur efficacité sur les ontologies de grande taille.

Certains travaux récents ont défini les défis qui doivent être relevés lorsqu'il s'agit de grandes ontologies.

Auteurs	Principe	Opérations	Outils	Observations
[Stuckenshmidt and Klein, 2004]	partitionnement automatique des automatique des grandes ontologies en modules plus petits basés sur la structure de la hiérarchie de classe.		SUMO	

Auteurs	Principe	Opérations	Outils	Observations
[Wang and al., 2006]	MOM	Décomposition		
[Hu and al., 2008]	Partitionnement basé sur la structure Clusters.	similitudes structurelles similitudes linguistiques.		
[Wang and al., 2011]	Neuronale Alignement d'ontologies basées sur la réduction ancres.	Réduction d'ancres		
[Jiménez-Ruiz and Grau, 2011]	- Intégration neuronale - Partionnement simple	Index Lexical à base de clustering Module de localité	LogMap	détection d'insatisfai-sabilité
[Saruladha and al., 2012a]	Partitionnement basé sur la structure. Proximité structurelle basée sur le voisinage.	l'appariateur linguistique V-DOC et l'appariateur structurel GMO.	LOMPT	
[Hartung and al.,2012]	Composition d'ontologies intermédiaires.			
[Jiménez-Ruiz and al., 2012]	Algorithmes de raisonnement et de diagnostic évolutifs, qui minimisent les incohérences logiques introduites par le processus de comparaison.			Interaction utilisateur.
[Lambrix and Kaliyaperumal, 2013]	Calculs de mappings partiels.		Frame-work	Semi-automatique
[Kachroudi and al., 2013]	Partitionnement blocs cohérents(sans perte	Partitionnement guidé par des ressources	Wordnet	

Auteurs	Principe	Opérations	Outils	Observations
	d'informations)	externes Ajout de concepts voisins à base de relations structurelles et sémantique avec le concept cible.		
[Diallo, 2014]	la combinaison du calcul lexical et du calcul de similarité contextuelle de l'apprentissage machine	Recherche d'Information	ServoMap	
[Pesquita and al., 2014]	- une large sélection de stratégies et d'algorithmes d'appariement - sélection automatique des sources de connaissances de base.	- Utilisation de sous graphes	AML GUI	
[Santos and al., 2015]	Techniques de réparation d'aligement.		AML	
[Ivanova and al., 2015]	Combinaison de méthodes .			- Présence de l'utilisateur.
[Setti Ahmed and al., 2015]	Partitionnement Clustering Algorithme K-means.	Mesure de similarité sémantique		Amélioration
[Wang and al., 2018]	Architecture Neuronale pour l'ajout des informations externes	Utilisation de ressources externes pour les définitions et contextes (Wikipedia)	OntoEmma Word2Vec	
[Jurish and Igler, 2018]	intégration de RDF et des techniques	RDF2Vec		

Auteurs	Principe	Opérations	Outils	Observations
	de classification communes. Graphes			
[Oliveira and Pesquita, 2018]	filtrage de l'espace de recherche basé sur des mappings partiels			
[Annane and al., 2018]	l'utilisation de ressources externes de connaissances de base	Background Knowledge (BK) est défini comme tout ensemble de ressources de connaiss-ances externes qui fournit des informations lexicales ou sémantiques	(1) une sélection basée sur un ensemble de règles et (2) une sélection basée sur un apprenti-ssage machine supervisé.	
[Moawed and al., 2019]	-les stratégies d'appariement inter, intra et hybride -Sur une approche d'appariement parallélisée fondée sur la mise en grappes -Méthodes de mise en grappes hiérarchiques	-Divise d'abord la tâche d'appariement d'origine en petites tâches indépendantes. -Stratégie d'appariement intra-parallèle	MetMat	

TABLE 3.3 – *Travaux connexes.*

3.5 Synthèse

Avantages et limites du partitionnement

L'approche de Stuckenschmidt et Klein ne considère pas la sémantique de l'ontologie, elle est applicable donc à travers différents langages d'ontologie ; tandis que l'approche de Cuenca Grau et al. s'appuie uniquement sur la logique de description, qui apporte à leur approche la garantie qu'il n'y aura aucune altération de l'information contenue dans les partitions par rapport au contenu de l'ontologie d'origine. Mais il reste à signaler que le calcul de degré de dépendance dans l'approche de Stuckenschmidt et Klein pourrait dans un certain sens être vu comme un facteur limitant l'efficacité de leur algorithme, dans la mesure où en plus de faire abstraction de la sémantique de l'ontologie, il ne considère aucunement le cadre dans lequel la partition créée entend être utilisée.

À l'instar de l'approche de Stuckenschmidt et Klein, l'algorithme de partitionnement de Cuenca Grau et al. n'est pas paramétrable, mais son réel défaut vient du fait qu'il ne s'applique pas à toutes les ontologies, car certaines ne sont pas partitionnables à l'aide des ϵ-connexions, et ce dans les cas où les compatibilités structurelle et sémantique ne pourraient être garanties lors du processus de partitionnement.

Avantages et limites de la modularisation

La façon la plus évidente pour l'Homme de visualiser une ontologie est de la représenter comme un graphe dont les sommets sont des concepts et individus et les arêtes sont des relations (hiérarchiques et sémantiques). Le problème d'extraction de module ontologique serait donc ainsi ramené à l'extraction d'un sous-graphe contenant les concepts les plus pertinents par rapport à la liste de termes entrée par l'utilisateur [Dupont and al., 2006]. Les avantages de cette approche structurelle sont multiples. Tout d'abord, il existe déjà des algorithmes efficaces d'extraction de sous-graphes. En supposant que l'ontologie dont on veut extraire le module est de qualité satisfaisante, cette approche ne déstructure pas l'ontologie dans la mesure où le module extrait conserve la même structure interne que l'ontologie d'origine. Et pour peu que le graphe ne soit pas dense, la complexité de l'algorithme s'en trouve réduite. [Patrick, 2014].

Les avantages de l'approche sémantique sont la pertinence et la précision, car ce qui fait le principal atout d'une ontologie demeure sa sémantique. Il suffit qu'un concept soit lié à un concept qui ne cadre aucunement avec le contexte pour lequel l'ontologie a été créée pour que la qualité de celle-ci s'en trouve altérée, car on se retrouverait ainsi avec une ontologie qui ne cadre plus avec la réalité. L'approche sémantique nous permet donc de mettre l'ontologie à plat et de tester la similarité sémantique de tous les couples de concepts de l'ontologie. La structure et la sémantique de l'ontologie sont des aspects dont on ne saurait faire abstraction lors de la modularisation [Patrick, 2014].

3.6 Conclusion

Nous avons vu dans le présent chapitre que dans les environnements distribués comme le web sémantique, les ontologies sont construites au travers de différents points de vues indépendamment les unes des autres. Elles sont non seulement hétérogènes mais aussi elles prennent des volumes importants .Ce qui peut produire lors de l'exploitation des systèmes existants des défaillances techniques telles le manque de mémoire ou des temps longs d'éxécution.

En effet, les outils actuels d'alignement d'ontologies perdent leur efficacité sur de grandes ontologies, l'objectif de ce travail était d'étudier les stratégies de partitionnement et de modularisation des ontologies larges pour voir si d'autres pistes pouvaient exister afin de remédier à ce genre de contraintes.

Ce chapitre décrit en premier deux méthodes utilisant l'algorithme de partitionnement d'ontologies PBM, développé pour le système d'alignement FALCON. Ces méthodes sont présentées par comparaison à la méthode PBM. Cette description est complétée, par la présentation de l'utilisation de la méthode de partitionnement PAP dans un contexte applicatif différent. Au lieu d'appliquer l'algorithme, comme PBM, successivement et indépendamment sur chaque ontologie, le contexte de la tâche d'alignement est pris en compte dès que possible dans le processus de partitionnement des méthodes PAP et APP . Celles-ci sont appliquées sur les deux ontologies simultanément, et utilisent les données d'alignement. Ces données d'alignement sont faciles à extraire, même à partir de grandes ontologies. Elles comprennent des paires de concepts, un concept de chaque ontologie, qui ont le même label, et des informations structurelles sur les ontologies à aligner.

La méthode PAP est bien adaptée pour les ontologies qui ont une structure dissymétrique ou des structures correspondant à des décompositions selon des points de vue différents. Cette méthode commence par décomposer l'ontologie la mieux structurée ou celle dont le point de vue de décomposition est retenu pour la construction des partitions et ensuite force la décomposition de l'autre ontologie à suivre le même modèle.

La méthode APP peut être appliquée lorsque les deux ontologies sont bien structurées et que leur structuration est faite selon un même point de vue. Cette méthode favorise la génération de blocs de concepts comparables sémantiquement car contenant des éléments majoritairement liés par des liens d'équivalence.

Le fait que les algorithmes de partitionnement étudiés utilisent seulement des données faciles à obtenir via un traitement peu coûteux, permet à des ontologies de très grande taille d'être partitionnées. C'est donc une démarche qui passe à l'échelle.

L'inconvénient principal c'est le fait d'avoir des partitions non cohérentes car il y a des concepts qui sont isolés et donc risque de perte de sémantique.

Par ailleurs, plus une ontologie est grande plus il est difficile de maintenir l'exactitude du modèle.

En plus et dans le cadre de la réutilisation il est intéressant de réutiliser les mêmes ontologies dans diverses applications. Dans le cas de la modularisation

, le concepteur crée un modèle compréhensible et appréhendé pour ensuite l'intégrer dans l'ontologie finale.C'est pourquoi la modularisation est une solution pour le passage à l'échelle parce que dès le départ les modules sont conçus pour être réutilisés .Par conséquent, la perte de sémantique est moins importante que dans le cas du partitionnement. Mais le risque est toujours présent.

D'un autre côté, nos recherches se sont donc essentiellement concentrées sur les méthodes de partitionnement des ontologies larges tout en montrant les mesures de similarités utilisées.

Le domaine de l'identification de la similarité a été considéré comme un sujet de recherche fortement recommandé dans les domaines du Web sémantique, de l'intelligence artificielle et de la littérature linguistique.

L'identification de la similarité dans les ontologies est un concept fondamental qui est adopté par plusieurs techniques telles que le regroupement, la fouille de données (data mining), le Web sémantique et en particulier, le domaine de la recherche de l'information. Ce dernier repose largement sur des mesures pour l'identification de la similarité entre les documents [Baeza and Ribeiro, 1999], [Salton and McGill, 1983].L'idée essentielle des approches PBM, APP et PAP est la prise en compte des rapports ontologiques entre concepts. Les rapports ontologiques entre concepts peuvent être détectés par un processus de calcul de similarité entre des paires d'objets contenus dans l'ontologie.

Finalement, nous avons remarqué que toutes les recherches se sont concentrées sur le partitionnement et la modularisation comme étant les deux stratégies communément utilisées pour rendre les systèmes d'information à base d'ontologies hétérogènes intéropérables. Les recherches se sont intéressées donc, sur les techniques permettant de réaliser les alignements des ontologies larges. Au vue de l'état de l'art, ces techniques ont atteints leurs limites. Nous pensons qu'au vu de ces limites, il serait judicieux d'inverser la problématique, et de pouvoir la reformuler. Dans le chapitre suivant nous apportons les éléments de réponse en présentant la nouvelle stratégie ONTEM. En effet, la nouvelle stratégie élimine la limite principale qui est le risque de perte de bon candidats au matching.

4 ONTEM : Une nouvelle stratégie d'alignement des ontologies larges à base d'extraction.

Sommaire

4.1 Introduction

Après des recherches approfondies et une bonne lecture de l'état de l'art, nous nous sommes posés la question : Existerait-t-il une autre stratégie que le partionnement et la modularisation pour l'alignement des ontologies larges tout en évitant les inconvénients déjà cités. A première vue, nous voulons rapprocher deux ontologies à aligner en examinant les éléments communs. A partir de ce moment, en se rapprochant de la théorie des ensembles, rechercher les éléments communs voudrait dire trouver l'ensemble des éléments qui appartiennent à la fois aux deux ontologies. A partir de ce moment, s'est posée une autre question : Parmi le reste des éléments des deux ontologies, peut-t-on encore les rapprocher ?. [Euzenat, 2004] nous apporte un élément de réponse car parmi les hypothèses de recherche des éléments communs aux deux ontologies, il avance que deux concepts sont similaires s'ils partagent assez d'éléments communs. Nous avons étudié de près les caractéristiques de chaque concept. Nous avons trouvé que l'étiquette « label »faisant partie de la description d'une classe pouvait nous apporter une richesse sémantique supplémentaire pour pouvoir rapprocher deux concepts appartenant aux deux ontologies. Nous nous sommes intéressés par la suite à étudier les concepts mis en correspondances par différents systèmes à base de calcul de similarité.

Nous avons trouvé que les concepts mis en relation, sont soit ils sont atomiques (le concept est constitué d'un seul mot), alors il y avait entre les deux concepts des chaines de caractères communes, soit les concepts étaient composés de plusieurs mots et là nous avons remarqué qu'ils partageaient des mots communs mais dans un ordre de classement qui pouvait être différent. L'idée nous est venu de classer les deux concepts dans le même ordre. Les deux concepts avaient finalement des parties communes mais malheureusement, il y avait des mots qui se répétaient à chaque fois tels que « set », « structure », et qui nous ont laissé devant un problème, celui de dire que les deux concepts sont similaires. A partir de cet instant, on s'est posé la question : « Si on pouvait supprimer ces mots, alors les deux concepts trouvés étaient identiques ». Nous avons dressé un tableau avec ce genre de mots, et nous avons pris à titre d'essai les ontologies larges du benchmark OAEI largeBio Track 2018. Nous avons trouvé que ces mots avaient un nombre d'occurrences assez important au niveau des ontologies étudiées. Etant donné que nous connaissons les principes de conception des bases de données, il nous a été assez évident de déduire qu'il fallait réduire au maximum ce nombre d'occurrences. Nous les avons supprimé des concepts à rapprocher : le résultat était évident , les deux concepts étaient identiques. Mais en cours du traitement de ces concepts une partie avait disparu. Il ne faut pas oublier que la syntaxe originelle des deux concepts est importante pour pouvoir dire que ces concepts sont similaires. Pour cela nous avons utilisé des indexes pour chaque concept, en mémorisant la syntaxe initiale et la syntaxe finale. Finalement, pour pouvoir retrouver la forme originelle de chaque concept il suffit de la rechercher au niveau de l'index correspondant. Le même problème que celui de « set »et « structure »s'est posé pour les prépositions telles que « of », « for »,etc.., ainsi que pour les caractère spéciaux « * », « / », etc.., et les chiffres de $0 - 9$. Nous

avons procédé de la même manière que « structure »et « set », nous les avons tout simplement supprimés.

Finalement, nous avons découvert une nouvelle stratégie d'alignement qui est complètement différente des stratégies existantes : le partionnement et la modularisation. Un prototype était nécessaire pour s'assurer du bien fondé de notre stratégie. Les tout premiers résultats étaient satisfaisants. Nous avons poussé nos recherches et nous sommes arrivés à dresser une liste de mots qui se répétaient dans chaque ontologie. Ces mots sont retirés automatiquement des concepts traités. Nous signalons que cette liste peut être établit par un expert en linguistique du domaine. Nous avons nommé cette liste comme étant la liste des mots de composition.

D'un autre côté nous avons essayé de réfléchir comment pouvoir rapprocher deux concepts dont la syntaxe était différente et qui partageaient des labels ayant des mots communs. Là aussi, nous avons bénéficié de l'hypothèse de [Euzenat, 2004] qui stipule que deux concepts sont similaires s'ils partagent assez d'éléments communs. A partir de ce constat, nous avons effectué les mêmes opérations que pour les noms de classe au niveau des labels. L'idée est de mettre en correspondance des concepts ayant des labels communs. Ce qui fût fait.

Les résultats obtenus après cette opération nous ont confirmé que notre stratégie a réussi le pari d'alignement des ontologies larges. Les résultats établis sur la base du Benchmark OAEI largeBio track 2018 le montrent clairement. Puisque les principes utilisés tout au long de notre stratégie reposent sur les techniques d'extraction, et que notre stratégie n'avait rien de similaire avec les stratégies de partitionnement et de modularisation, nous avons nommé notre stratégie « Extraction ».

4.2 La stratégie ONTEM

Dans cette section, nous décrivons ONTEM, une méthode à base d'extraction pour l'alignement ontologique [Zerhouni and Benslimane,2019a]. L'architecture de notre méthode est présentée à la figure 4.1. Les entrées sont les ontologies qui doivent-être alignées. La sortie est un alignement entre les ontologies qui consiste en un ensemble de mappings qui sont acceptés après validation.

Au meilleur de nos connaissances, dans le contexte de l'alignement ontologique, notre travail est le premier qui favorise l'extraction des concepts et des étiquettes des ontologies pour les aligner.

Le processus d'alignement comprend quatre étapes principales : le prétraitement, l'identification des entités communes, la génération des mappings et la génération de l'alignement. Le principe serait d'extraire les concepts communs aux deux ontologies et de pouvoir effectuer un ensemble d'opérations permettant la création automatique de correspondances. En examinant de près la structure des ontologies et plus particulièrement les étiquettes « labels », nous avons déduit qu'il est possible de trouver de nouvelles correspondances en rapprochant les étiquettes, et il nous sera donc facile de rapprocher les concepts correspondants.

De plus, nous utilisons le lexique WordNet pour les noms de concepts et les étiquettes pour lesquels aucune correspondance n'a été trouvée en recherchant les synonymes correspondants.

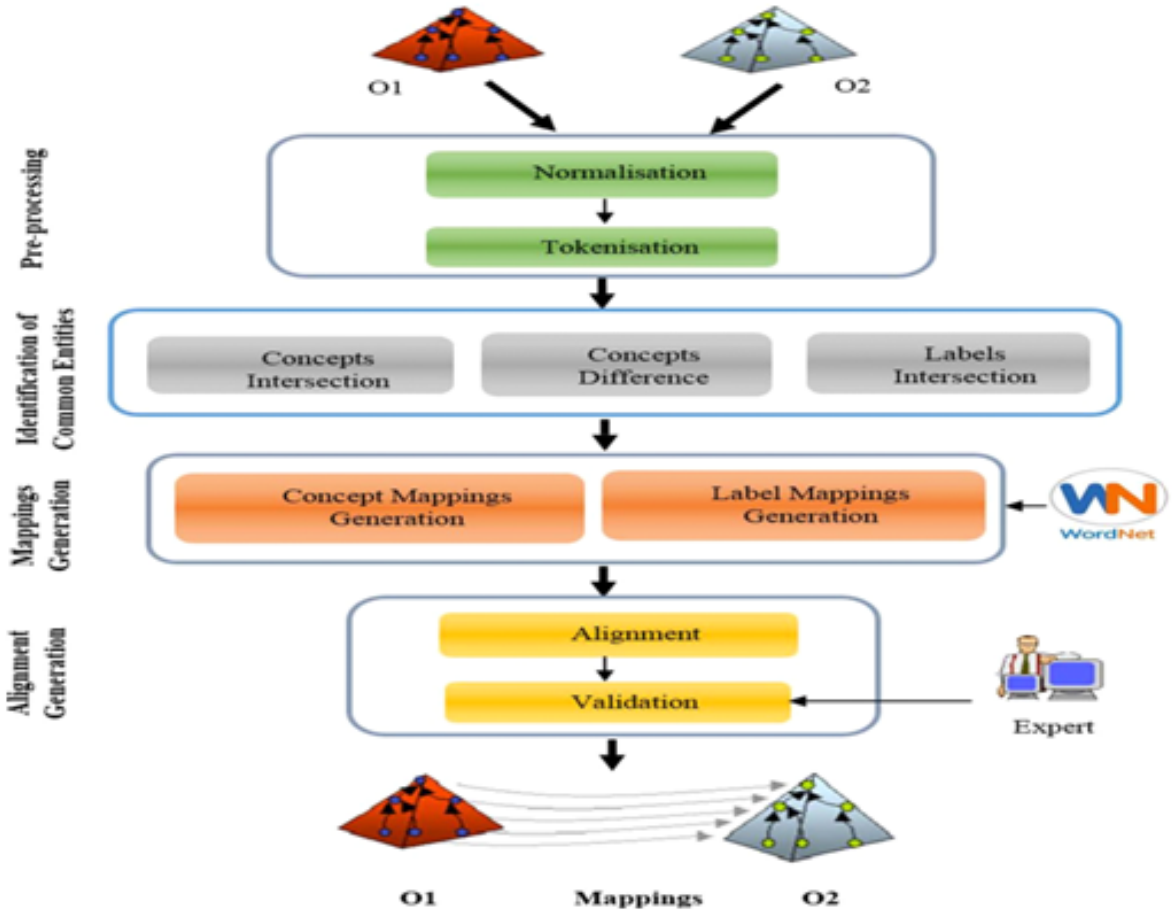

FIGURE 4.1 – *Architecture de ONTEM*

4.2.1 Prétraitement

Pour faciliter le processus de comparaison des termes des ontologies caractérisant les classes et leurs propriétés (c'est-à-dire le calcul des distances entre leurs chaînes de caractères), il est très important d'effectuer plusieurs opérations de prétraitement. Ils améliorent considérablement les résultats de l'alignement [Zerhouni and Benslimane,2019b].

De plus, lors de l'alignement basé sur l'extraction des synonymes, les opérations de prétraitement facilitent leur reconnaissance par les bases de données lexicales et/ou les dictionnaires de synonymes.

Les méthodes terminologiques [Anam and al., 2014] comparent les chaînes de caractères. Plusieurs idées ont été développées dans la littérature en utilisant des comparaisons linguistiques. Elles peuvent être appliqués aux noms, aux labels ou aux commentaires sur les entités pour trouver celles qui sont similaires.

Cette technique repose sur l'hypothèse suivante : deux termes sont similaires, c'est-à-dire qu'ils désignent des concepts similaires [Euzenat and Valtchev, 2004]. Il existe plusieurs façons d'évaluer la similarité entre deux entités. La façon la plus courante est de définir une chaine de mesures de similarité [Cheatham and Hitzler, 2013]. L'autre moyen serait de déterminer des concepts strictement identiques sur le plan syntaxique.

Nous nous sommes inspirés de la deuxième hypothèse : deux termes sont similaires s'ils partagent des éléments suffisamment importants [Euzenat and Valtchev, 2004]. Les étiquettes « label »font partie des éléments communs importants. Ils représentent une source sémantique importante qui nous permettra de rapprocher des concepts syntaxiquement différents qui partagent des termes communs.

Notions de concepts et de labels

Un concept (une classe) est tout ce dont on peut parler. Il peut s'agir de la description d'une tâche, d'une fonction, d'une action, d'une stratégie, d'un raisonnement, etc.[Gómez-Pérez, 1999].

Les classes d'une ontologie sont extraites après la conceptualisation du domaine ciblé en fonction des objectifs à atteindre et de l'application qui va utiliser l'ontologie.

Nous appelons ancre un concept d'une ontologie appariée à un concept unique d'une autre ontologie, du même nom et du même sens qu'elle. Nous parlons d'une paire d'ancres lorsque nous nous référons aux deux concepts liés l'un à l'autre. Cette définition révèle trois caractéristiques pour la paire de concepts considérés : ils ont le même nom, des significations considérées similaires et leur correspondance est unique.

Avant d'identifier une paire d'ancres, ces trois caractéristiques doivent être vérifiées successivement. Il s'agit de deux types de paires d'ancres : celles obtenues à partir de l'intersection de concepts et celles obtenues à partir d'étiquettes « labels »communes. L'approche exploite la richesse des étiquettes conceptuelles « labels ». Les étiquettes sont composées de plusieurs mots [Ruder and al., 2018]. Ils supposent que des concepts similaires partagent une partie de leur étiquette « label ». Il est donc plus approprié d'aligner des taxonomies dont les noms de concepts sont des expressions composées de plusieurs mots car, dans ce cas, ces noms peuvent partager des mots, ce qui peut révéler des points communs entre les concepts concernés. Lorsque ces étiquettes sont composées de plusieurs mots, et que l'intersection de ces étiquettes révèle l'existence d'éléments communs, alors cela représentera une richesse supplémentaire permettant d'obtenir des concepts communs en plus des concepts trouvés directement par l'opération d'intersection. Un concept ou une étiquette se compose d'un ensemble de mots concaténés à l'aide du caractère "_".

Les techniques linguistiques sont basées sur les techniques de traitement du langage naturel $(NLP3)$ en exploitant les propriétés morphologiques des termes utilisés. Les techniques linguistiques utilisées sont les suivantes :

Normalisation

Il s'agit de transformer tous les caractères des termes ontologiques en minuscules. Suppression des caractères spéciaux, des espaces et des nombres.
Les caractères spéciaux ainsi que les chiffres contenus dans les noms de classe ou des labels sont supprimés, comme « , », « . », « - », « * », « : », « & », « / », « , », « = », « # », « ; », « ^ », « 1 », « 2 », « 3 », « 4 », « 5 », « 6 », « 7 », « 8 », « 9 ».

Élimination des conjonctions de coordination, des articles et des prépositions

Les conjonctions de coordination "and", "or", "but", "so", "yet", "not", "for" sont supprimées des termes ontologiques.
Les prépositions "in", "on", "to", "of", "of", "with", "at", "for", "over", "by" sont supprimées des termes ontologiques.
Les articles "the", "a", "an" sont supprimés des termes ontologiques.

Élimination des mots désignant des ensembles ou des éléments (mots de composition)

Les mots "set", "bone", "structure", "component", "subdivision", "collection", "body", "part", "portion", "branch" sont supprimés des termes ontologiques.
Le tableau 4.1 donne une idée sur le rôle important joué par ces mots au niveau des entités composées des ontologies du benchmark OAEI LargeBio track 2018 [1].

Words	FMA Whole	NCI Whole	SNOMED LARGE FRAGMENTS	FMA small Overlapping NCI	FMA Small Overlapping SNOMED
Set	7891	122	417	625	2891
Structure	268	3795	113026	105	115
Component	779	13673	1936	11485	378
Subdivision	6445	0	345	1108	2046
Collection	0	19	101065	4	0
Body	9349	1863	10244	221	669
Bone	22488	1808	15762	461	1277
Part	31397	3916	2543	585	1126
Branch	41157	144	2422	300	1635
Portion	356	107	680	93	151

TABLE 4.1 – *Nombre d'occurrences de mots de composition pour les ontologies OAEI LargeBio Track 2018.*

1. http ://www.cs.ox.ac.uk/isg/projects/SEALS/oaei/2018/

Tokenisation

La tokenisation, qui est une analyse lexicale consiste à transformer un flux de caract ères en un flux de tokens par un analyseur (tokenizer) qui reconnaît la ponctuation, les caractères blancs, etc.
Cette phase est suivie du tri des tokens (mots) et de leur concaténation, correspond à la phase de prétraitement qui est illustrée par l'algorithme 1.

Algorithme 1**. Prétraitement**

```
Inputs : Ontology O
Outputs : List_of_Concepts, List_of_Labels, Index_of_Concepts,
Index_of_Labels
Begin
For Each Concept of O do
        CN=Normalize (Concept)
        CT=Tokenize (CN)
        Add (Concept, CT)
        Add ((Concept, CT), INDEX_CONCEPTS)
    For Each Label of Concept
        LN=Normalize (Label)
        LT=Tokenize (LN)
        Add (LaBel)
        Add ((Label, LT), INDEX_LABELS)
    End For
End For
Return (List_of_Concepts, List_of_Labels, Index_of_Concepts,Index_of_Labels)
End
```

Extraits

```
<owl :Classrdf :about="Aortic_aneurysm" >

<rdfs :subClassOfrdf :resource="aneurysmaortic" />

</owl :Class>

<owl :Classrdf :about="Aortic_aneurysm_MRI" >

<rdfs :subClassOfrdf :resource="aneurysmaorticmri" />

</owl :Class>
```

Extrait 1 **: Extrait de INDEX CONCEPTS** *O*1 **(ICO1.OWL)**

```
<owl :Classrdf :about="Aortic_Arch" >

<rdfs :subClassOfrdf :resource="aorticarch" />

</owl :Class>

<owl :Classrdf :about="Aortic_Arch_Branch" >

<rdfs :subClassOfrdf :resource="aorticarch" />

</owl :Class>
```

Extrait 2 : Extrait de INDEX CONCEPTS *O*2 (*ICO*2.*OWL*)

Nous avons utilisé les fichiers index *ICO*1 et *ICO*2 avec l'extension OWL pour bénéficier des facilités accordées par l'API Jena lors de la recherche de tous les concepts ayant le même nom de référence. Dans cet exemple, on pourra rechercher tous les concepts ayant le même nom de référence que "aorticarch" qui sont : "Aortic_Arch" et "Aortic_Arch_Branch".

Les fichiers *ICO*1.*OWL* et *ICO*2.*OWL* servent à retrouver la syntaxe originale des noms de concepts et de labels.

Exemple :

Considérons les classes " Abdominal_part_of_ureter " et " Structure_of _abdominal_portion_of_ureter " appartenant respectivement aux ontologies Fma.owl et Snomed.owl.
En supprimant les termes "part", "structure" et "portion", et en effectuant l'opération qui permet de rendre en minuscules ces chaînes de caractères, on obtient le résultat "abdominal_ureter".
Si la liste des mots d'un concept ou d'une étiquette n'est pas classée par ordre alphabétique, une opération de tri est nécessaire. Ceci évitera de faire de nombreuses comparaisons entre les mots appartenant aux deux ontologies. Par conséquent, nous réduirons efficacement le temps de l'opération d'alignement.
Enfin, en supprimant le symbole "_", on obtient comme résultat "abdominalureter". Les deux classes sont donc équivalentes. Il est évident que la valeur de la mesure de similarité entre les deux classes est égale à ”1”.

4.2.2 Identification des entités communes

Puisque les ontologies traitées concernent le même domaine, il est évident qu'elles partagent des éléments communs. Cette observation nous amène directement à considérer les éléments communs aux deux ontologies. Les éléments communs aux deux ontologies peuvent étre des concepts, des étiquettes, des propriétés, ainsi que des relations. Compte tenu que les deux ontologies sont volumineuses, l'objectif est de faire correspondre les concepts de la première ontologie avec les concepts de la seconde ontologie. Pour ce faire, nous utiliserons l'opération "Intersection" pour trouver les concepts communs aux deux ontologies.

L'intersection des entités communes

Compte tenu d'un domaine D et de deux ontologies $O1 \in D$ et $O2 \in D$.
Soient :

- $LCO1$ et $LCO2$ la liste des concepts de l'ontologie1 et de l'ontologie2 respectivement.
- $LLO1$ et $LLO2$ la liste des labels de Ontology1 et Ontology2 respectivement.
- LIC la liste des concepts communs aux deux ontologies. $LIC = LCO1 \cap LCO2$.
- LIL la liste des labels communes aux deux ontologies.$LIL = LLO1 \cap LLO2$.

L'intersection de concepts communs et d'étiquettes communes est réalisée respectivement par l'algorithme 2 et l'algorithme 3.

Différence de concepts

Dans cette phase, nous ne retenons que les concepts non composés. Nous ne sélectionnerons que des concepts qui ne sont pas composés et qui n'appartiennent pas à LIC. Un concept composé est un nom qui contient au moins un caractère"_".
Soient :

- $LNCCO1$, liste des concepts non composés de l'ontologie 1.
- $LNCCO2$, liste des concepts non composés de l'ontologie 2.
- $LCSSO1$, liste des concepts pour la recherche des synonymes de l'ontologie 1.
- LCC, liste des correspondances de concepts obtenues directement.

Algorithm 2. Intersection of Concepts

Inputs : LCO1, LCO2
Outputs : *LIC*
Begin
LIC = Intersection (LCO1, LCO2)
Return (LIC)
End

Algorithm 3. Intersection of Labels

Inputs : LLO1, LLO2
Outputs : LIL
Begin
LIL = Intersection (LLO1, LLO2)
Return (LIL)
End

La différence de concepts est obtenue par l'algorithme 4.

Algorithme 4. Difference of concepts

Inputs : LNCCO1, LIC
Outputs : LCSSO1
Begin
LCSSO1 = Difference (LNCCO1, LIC)
Return (LCSSO1)
End

4.2.3 Génération de correspondances

Le processus de découverte de correspondances, appelé alignement ontologique, est une fonction f qui s'applique à deux ontologies $O1$ et $O2$, avec un ensemble de paramètres p (poids, seuils, etc.) et un ensemble de ressources externes r, et produit un ensemble de mises en correspondances A.

A=f($O1, O2$,p,r)

L'alignement comporte plusieurs étapes [Ehrig , 2007]. Extraction des données à rapprocher, sélection des paires d'éléments à comparer, calcul d'une similarité pour chaque paire sélectionnée et déduction de l'alignement à partir des mesures de similarité calculées précédemment. Chaque méthode de calcul d'une mesure de similarité correspond à l'exécution d'une technique d'alignement particulière. Plusieurs classifications de ces

techniques ont été proposées dans la littérature [Angele and Shnurr,2005], [Diallo, 2014], [Zhao and Zhang, 2016] et [Stoilos and al.,2018].

Correspondances directes de concepts

Nous constatons que chacun des concepts de la première ontologie renvoie directement aux concepts correspondants de la deuxième ontologie (voir figure 4.2).
L'algorithme pour générer des mappings de concepts directs est représenté par l'algorithme 5.

Algorithme 5. Génération directe de mappings conceptuels

```
Inputs : LIC, ICO1, ICO2
Outputs : LC, Index_Mappings
Begin
For Each Concept LIC do
        Entity_Name_1=Read(Concept, ICO1)
        Entity_Name_2=Read(Concept,ICO2)
        Add(Entity_Name_1,LCC)
        Add(Entity_Name_2,LCC)
        Add (Entity_Name_1, Entity_Name_2, Index_Mappings)
EndFor
Return(LCC)
End
```

Exemple de correspondances de concepts de type $(1-1)$

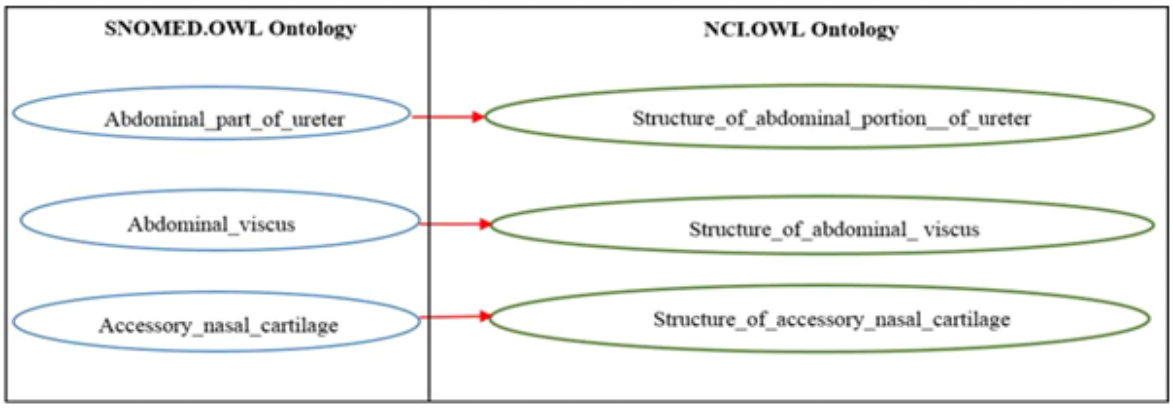

FIGURE 4.2 – *Correspondances de concepts de type* $(1-1)$

Correspondances de concepts à base de labels communs

A titre d'exemple, nous remarquons que l'intersection des deux ontologies : SNOMED.OWL et NCI.OWL sur les « labels » donne le mot "vertebral". Le label "vertébral"

désigne un concept "Vertebral_part" au niveau de la première ontologie et trois concepts "Vertebral", "Vertebral_body" et "Vertebral_bone" au niveau de la deuxième ontologie (voir figure 4.3).

Lors de la création des correspondances, si l'étiquette "vertebrale" est répétée plusieurs fois dans la liste des étiquettes, on vérifie alors si la première ancre {Vertebral_part,Vertebral} est déjà enregistrée dans l'Index_Mappings obtenus à partir des concepts communs. Si c'est le cas, nous passons à la deuxième ancre {Vertebral_part, Vertebral_body}, et ainsi de suite.

L'algorithme de génération des correspondances de concepts à partir de la liste des étiquettes communes est représenté par l'algorithme 6.

Exemple de correspondances de concepts de type (1-N)

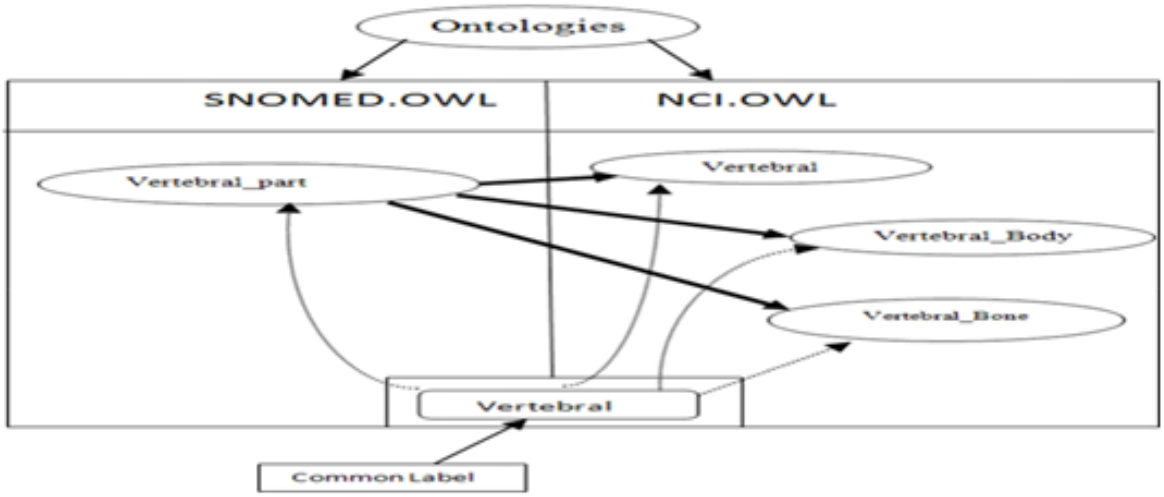

FIGURE 4.3 – *(1-N) correspondances de concepts*

Algorithme 6. Génération de mappings basés sur des étiquettes communes

```
Inputs : LIL, ILO1, ILO2, ICO2, Index_Mappings
Outputs :LCL, Index_Mappings /* Updated */
Begin
For Each (Label ∈ LIL) do
/* browse the labels of LIL */
        Entity_Name_1=Read(Label, ILO1)
        Entity_Name_2=Read(Label, ILO2)
        Found=0
    While (Entity_Name_2 hasValuesIs True) && (Found==0)
    /* Retrieving Entity_name2 not yet used */
    IF (Entity_Name_1,Entity_Name_2) not found in Index_ Mappings
        Add(Entity_Name_1,LLC)
        Add(Entity_Name_2,LLC)
        /* insert Entity_Name_1 and Entity_Name2 in LLC */
```

```
            Add(Entity_Name_1,Entity_Name_2, Index_Mappings)
            /*Update of the Index_Mappings*/
            Found=1 ;
        Else
            Entity_Name_2=Read(Label, ILO2)
        EndIf
        EndWhile
EndFor
Return(LCL, Index_Mappings)
End
```

Correspondances entre concepts basées sur le lexique Wordnet

Les techniques basées sur des chaînes de caractères ne sont pas suffisantes lorsque les concepts sont sémantiquement proches et que leurs noms sont différents. Interroger une ressource linguistique telle que WordNet [Miller and al.,1993] peut indiquer que les concepts sont similaires. Dans notre méthode d'alignement, en plus des méthodes lexicales, nous utilisons les relations de synonymie du lexique WordNet. L'avantage est que nous avons une plus grande couverture syntaxique et sémantique de chacune des deux ontologies à aligner. Toutefois, l'utilisation de cette ressource peut affecter la qualité des correspondances. En effet, les termes synonymes dans une ontologie n'impliquent pas que ces termes sont synonymes dans une autre. Pour cette raison, l'utilisation de la ressource WordNet externe est optionnelle. La recherche de synonymes ne concernera que les concepts non composés qui n'appartiennent pas à la liste des concepts communs aux deux ontologies.

Exemple de correspondances entre concepts basées sur le lexique Wordnet

Nous allons d'abord effectuer pour chaque ontologie l'opération de soustraction des concepts trouvés à l'étape de l'intersection des concepts.

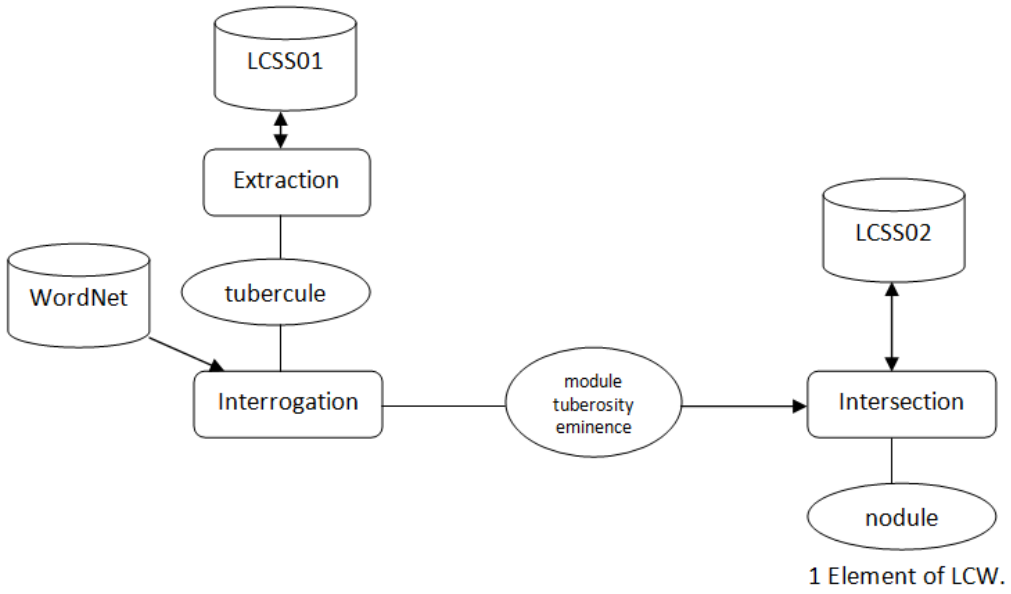

FIGURE 4.4 – *Génération de mappings en utilisant Wordnet*

La génération des correspondances basée sur WordNet est réalisée par Algorithme 7.

Algorithm 7. WordNet-based Mappings generation

Inputs : LCSSO1, LCSSO2, Index Mappings Outputs : LCW, Index Mappings /* Updated */
Begin
For Each Concept_1 (LCSSO1) do
/*It browses the List of Concepts for Searching Synonyms in ontology1 */
Entity_Name_1=Read(Concept_1,LCSSO1) Wordnet_List=Read(Entity_Name_1, Wordnet)
Add_List =Intersection(Wordnet_List,LCSSO2) Found=0
Read(Concept_2, Add_List)
While (Add_List is not empty) && (Found==0)
IF (Concept_1,Concept_2) $\notin$ *Index_Mappings*
Add(Concept_1, LCW)
Add(Concept_2, LCW)
/* add Concept_1 and Concept_2 in LCW */
Add(Concept_1, Index_Mappings)
Add(Concept_2, Index_Mappings)
/* add(Concept_1 and Concept_2) in Index_Mappings*/
Found=1 ;
Else
Read(Concept_2, Add_List)
EndIf

EndWhile
EndFor
Return(LCW, Index_Mappings)
End

Extraits

Les extraits suivants concernent les fichiers LIC (Liste Intersection Concepts) et LIL (Liste Intersection Labels)

```
<dc :name>aneurysmaortic</dc :name>
<dc :name>aortaascendingendarterectomy</dc :name>
<dc :name>etanercept</dc :name>
<dc :name>findingmelanomaskintnm</dc :name>
```

Extrait3 : Extrait du fichier LCO1.OWL contenant tous les concepts de l'ontologie oaei_SNOMED_small_ovelapping_NCI.owl

```
<dc :name>aneurysmaortic</dc :name>
<dc :name>changesmacrocystic</dc :name>
<dc :name>bchromogranin</dc :name>
<dc :name>hepatitis</dc :name>
<dc :name>angiofollicularcellhyperplasialymphoidplasmatype< /dc :name>
```

Extrait4 : Extrait du fichier LCO2.OWL contenant tous les concepts de l'ontologie oaei_NCI_small_overlapping_snomed.owl

```
<dc :name>aneurysmaortic</dc :name>
<dc :name>seizure</dc :name>
<dc :name>atypiacytologicsevere</dc :name>
<dc :name>ranibizumab</dc :name>
<dc :name>femoralhernia</dc :name>
```

Extrait5 : Extrait du fichier LIC.OWL

Les mêmes opérations que pour les concepts sont effectuées pour les labels, et nous obtenons le fichier LIL.OWL à partir de l'intersection des fichiers LLO1.OWL et LLO2.OWL.

```
<dc :name>methodscreening</dc :name>
<dc :name>relative</dc :name>
<dc :name>eyelidupper</dc :name>
<dc :name>hydrochloridenaphazoline</dc :name>
```

Extrait6 : Extrait du fichier LIL.OWL

Comme nous le remarquons, les concepts et les labels sont normalisés et tokenisés (les tokens sont triés et concaténés en supprimant le caractère ´ _ ˇ). L'extension des fichiers LIC et LIL est OWL. Ce choix est retenu pour bénéficier des avantages des opérateurs algébriques : l'intersection et la différence fournis par l'API Jena.

Diagnostic_laparoscopy
Diagnostic_Laparoscopy
Vertebral_part
Vertebral_Body
Ventricular_arrhythmia
Ventricular_Arrhythmiasex iodine
Vertebral_part
Vertebral
Sex_structure
Sex

Extrait8 : Extrait du fichier LCL.TXT .

4.2.4 Génération de l'alignement

L'alignement sera crée automatiquement à partir des listes LCC, LCL et LCW. L'algorithme de génération de l'alignement est illustré par l'algorithme 8.

Algorithm 8. Alignment Generation

```
Inputs : LCC, LCL, LCW
Outputs : F-ALIGN
Begin
For Each Concept ∈ LCC do
        Entity_Name_1=Read(Concept, LCC)
        Entity_Name_2=Read(Concept, LCC)
        Add(Entity_Name_1,F-ALIGNE)
        Add(Entity_Name_2,F-ALIGNE)
EndFor
For Each Concept ∈ LCL do
        Entity_Name_1=Read(Concept, LCL)
        Entity_Name_2=Read(Concept, LCL)
        Add(Entity_Name_1,F-ALIGNE)
        Add(Entity_Name_2,F-ALIGNE)
EndFor
For Each Concept ∈ LCW do
        Entity_Name_1=Read(Concept, LCW)
        Entity_Name_2=Read(Concept, LCW)
        Add(Entity_Name_1,F-ALIGNE)
        Add(Entity_Name_2,F-ALIGNE)
EndFor
Return (F-ALIGNE)
End
```

4.3 La compléxité du processus d'alignement

Les outils d'appariement d'ontologies devraient générer des correspondances de haute précision et de recall, peu importe la taille ou le type des ontologies d'entrée. Cependant, à mesure que la taille des ontologies d'entrée augmente, le nombre d'axiomes qu'un outil d'alignement d'ontologie sur lesquels il doit raisonner pour générer des correspondances précises et complètes, augmente aussi. Cette demande de puissance de raisonnement accrue augmente la complexité globale du processus d'alignement, ce qui entraîne une diminution considérable de la qualité des correspondances par rapport aux mesures standard de précision et de rappel. Ainsi, une augmentation de la taille des ontologies d'entrée pénalise la qualité des correspondances générées par un outil [Hamdi and al., 2009a]. Il est à noter que la qualité des correspondances de ONTEM ne souffre pas de ce type de problème.

4.3.1 Demande pour plus de mémoire

Pour mettre en correspondance les entités de deux ontologies d'entrée, une approche naïve consiste à comparer chaque entité de l'ontologie source à toutes les entités de l'ontologie cible. Cette approche produit cartésienne de l'alignement des entités de deux ontologies d'entrée donne lieu à une complexité spatiale de O(n^2) (en supposant que chaque ontologie d'entrée a n entités). La complexité spatiale de O(n^2) implique le maintien en mémoire de plusieurs valeurs de similarité entre les entités de l'ontologie source et de l'ontologie cible. Ce problème est aggravé par le fait que la plupart des outils d'alignement ontologique intègrent plusieurs mesures pour améliorer la qualité des correspondances qu'ils produisent. Par conséquent, le nombre de valeurs de similarité qui devront être conservées en mémoire sera k(n^2), k étant le nombre de mesures(matchers) utilisés dans un outil. Un processus d'appariement ontologique avec la complexité spatiale de O(n^2) peut facilement conduire à un dépassement de mémoire dans le cas d'un grand n.

Notre processus d'alignement ontologique a une complexité spatiale de O(n). Ainsi, nous évitons l'erreur de manque de mémoire dans le cas d'un grand n.

4.3.2 Augmentation du temps d'exécution du processus d'alignement

Comme pour la complexité spatiale, l'alignement de deux ontologies en entrée nécessite le produit cartésien des concepts de chaque ontologie, ce qui produit une complexité temporelle de O(n^2) si chaque ontologie a n entités. Si nous considérons l'efficacité d'un alignement et supposons que pour chaque calcul de similarité, l'alignement prend un temps t, alors la complexité temporelle globale du processus d'alignement est O($n^2 \times t$). Malheureusement, les utilisateurs sont généralement impatients et un processus qui a une complexité temporelle O($n^2 \times t$) exigerait d'eux d'attendre significativement les résultats de l'alignement même pour un petit nombre de n entités. La complexité temporelle globale du processus d'appariement d'ONTEM est O($n \times t$). Par conséquent, les utilisateurs n'attendraient pas très longtemps par rapport au temps cité ci-dessus.

4.4 Tableau comparatif des stratégies d'alignement

Le tableau 4.2 synthétise les principales différences des deux stratégies d'alignement éxistantes : le partitionnement et la modularisation, avec notre stratégie d'extraction (ONTEM).La comparaison entre les trois stratégies est basée sur les critères suivants :

CRITERES	PARTITIONNEMENT	MODULARISATION	EXTRACTION
ENTREES	2 Ontologies Larges	2 Ontologies Larges	2 Ontologies Larges
CHARGEMENT	PARTIEL	PARTIEL	TOTAL
INTERVENTION UTILISATEUR	OUI (fixe taille sous graphe)	OUI (fixe taille sous graphe)	NON
TEMPS EXECUTION	$\leq n^2 \times t$	$\leq n^2 \times t$	$n \times t$
ESPACE MEMOIRE K : nombre de matchers	$K \times n^2$	$K \times n^2$	n
CONNAISSANCE STRUCTURE ONTOLOGIE	NECESSAIRE	NECESSAIRE	OPTIONNELLE
PRINCIPE	Décomposer 2 ontologies larges indépendantes l'une de l'autre en des blocs ou partitions	Transformation des ontologies en graphe. Décomposition 2 ontologies larges en modules et déterminer les modules les plus similaires.	Extraction des entités communes aux 2 ontologies.
TECHNIQUES	- Utilisation de l'algorithme de partitionnement ROCK et d'un algorithme de division hiérarchique pour d'autres travaux pour la décomposition. - Utilisation des ancres et des documents virtuels pour la définition des partitions les plus similaires - Techniques de	- Utilsation de sous graphes. Utilisation des $\epsilon-$ connections pour la transformation des ontologies et un algorithme de partitionnement pour la décomposition. Détermination des modules les plus similaires (similarité lexicale grâce à l'utilisation de Edit distance).	Utilisation d'une table de mots de composition établit par un expert en linguistique. Utilisation d'opérations algébriques. Utilisation d'indexes pour la mémorisation des syntaxes des concepts. originelles et après transformation.

CRITERES	PARTITIONNEMENT	MODULARISATION	EXTRACTION
	calcul de distances de similarité.		
MATRICE SIMILARITE	OUI	NON	NON
CONSERVATION SEMANTIQUE ?	Graphique : NON Logique : OUI Clustering : NON	OUI	OUI
RAISONNEMENT ?	Graphique : NON Logique : OUI Clustering : NON	OUI	NON
RESSOURCES EXTERNES	OUI	OUI	OUI
SORTIES	Un ensemble de blocs ou de partitions similaires	Un ensemble de modules similaires	Aucun module Aucun bloc Un ensemble de correspondances obtenus soit directement entre concepts communs, soit à partir de labels communs, soit à partir de WordNet
AVANTAGES	Utilisation d'un algorithme qui passe à l'échelle Réduction de l'espace de recherche	Réduction de l'espace de recherche et donc un meilleur temps d'éxécution. Facilite la maintenance et le raisonnement	Aucun calcul de similarité donc temps de traitement optimisé. Réduction de l'espace de recherche . Stratégie qui peut être composée avec le partitionnement ou la modularisation.
INCONVENIENTS	La décomposition est faite à priori sans considérer l'objectif d'alignement en s'appliquant sur chaque ontologie indépenda-	L'utilisation des $\epsilon-$ connection n'est pas une approche fiable pour toutes les ontologies. Il y a en effet	A tester sur des ontologies réelles.

CRITERES	PARTITIONNEMENT	MODULARISATION	EXTRACTION
	mment l'une de l'autre. Il ya donc un risque que l'alignement ne comprenne pas toutes les correspondances souhaitées donc perte d'information. Le calcul des blocs à mettre en correspondance est coûteux (en temps de traitement).	un risque que certains nœuds ne soit pas assignés aux bons modules. Lors de la détermination des modules similaires, certains modules non-pertinents sont écartés. Cette étape peut engendrer une perte d'informations. L'approche n'a pas permis d'améliorer la qualité des résultats de matching (précision et recall).	

TABLE 4.2 – *comparatif des stratégies d'alignement*

Ces approches possèdent un point en commun, c'est qu'elles tentent toutes d'améliorer le processus de matching à large échelle en utilisant des techniques qui permettent de décomposer le problème de matching en des sous problèmes ou des techniques qui permettent d'optimiser l'espace de recherche et d'améliorer les performances du matching. En effet, dès que l'on passe à un contexte dynamique, les performances des matchers se réduisent et le temps d'exécution devient plus long, en plus du nombre de correspondances bruitées qui sont générées.

Le tableau 4.2 illustre les différences entre les approches de Partitionnement et Modularisation et Extraction en se basant sur les avantages et inconvénients de leurs stratégies.

4.5 Conclusion

Nous remarquons que les stratégies utilisées dans les approches partitionnement et modularisation ont comme inconvénient le risque de perte des bonnes correspondances contrairement à la stratégie ONTEM qui offre des valeurs élevées de fiabilité.
Malgré que les approches partitionnement et modularisation ont pour avantage d'améliorer les performances en termes de temps d'exécution et ceci en limitant l'espace de recherche grâce aux stratégies qui décomposent les ontologies. Alors que la stratégie ONTEM s'appuie plus sur l'aspect conservation (sans perte d'information).

Nous avons également déterminé les avantages et les inconvénients de chacune des deux stratégies existantes. L'avantage principal consiste à réduire l'espace de recherche des correspondances afin d'améliorer les performances et ceci grâce à la décomposition. Cependant, les deux stratégies présentent une limite principale qui est le risque de perte des bons candidats au matching et la sélection des mauvais candidats.

Expérimentation 5

Sommaire

5.1 Introduction

Les principales ontologies biomédicales telles que SNOMED CT[1], NCI[2] et FMA[3] sont grandes et sont au format OWL. Elles sont très demandées pour divers projets de recherche dans le domaine biomédical. Ces ontologies sont complexes et sont basées sur diverses vues et vocabulaires de modélisation. Pour souligner la validité de notre méthode, nous devons comparer les alignements produits par ONTEM avec des alignements de référence [Zerhouni and Benslimane,2019b].

Un alignement de référence est un alignement considéré comme "correct" entre deux ontologies. Notre objectif est de valider la méthode d'extraction ONTEM en traitant les ontologies volumineuses contenues dans la section OAEI LargeBio Track 2018[4]. Nos résultats seront comparés aux alignements de référence de cette section.

5.2 La Description du Dataset et de l'environnement d'expérimentation

Dans le registre évolutif, le dataset LargeBio track 2018 consiste à rechercher des alignements entre le modèle d'anatomie (FMA), la SNOMED CT et le National Cancer Thesaurus Institute (NCI), qui sont sémantiquement riches et contiennent des dizaines de milliers de concepts. Le Dataset LargeBio track 2018 a trois problèmes d'alignement ontologique, à savoir : (FMA, NCI), (FMA, SNOMED) et (SNOMED, NCI). Les ontologies FMA.OWL, NCI.OWL et SNOMED.OWL contiennent respectivement 78989, 66724 et 122464 classes.

Le prototype ONTEM a été développé sur la plate-forme Eclipse Helios, en utilisant le langage de programmation java et l'API jena2.4, ainsi que le langage de lecture de graphes sémantiques SPARQL. La machine sur laquelle les travaux ont été effectués dispose d'un processeur Intel® Core TM (2) Duo CPU E75002, 93 GHz 2, 94 GHz, 4, 0 Go de RAM, système d'exploitation 32 bits, Windows 7 Professional N. Les opérations d'évaluation ont utilisé OWL3 API [Horridge and Bechhofer,2011]. La bibliothèque OntoBridge[5] a été utilisé pour la lecture des ontologies OWL, ainsi que pour d'autres opérations de manipulation.

En tant que serveur local, nous avons utilisé le serveur APACHE installé grâce à l'application WAMPSERVER.

5.3 Comparaison de ONTEM avec les autres méthodes

Les paramètres d'évaluation Precision, Recall et F-measure ont été utilisés pour comparer notre méthode ONTEM avec d'autres méthodes pionnières dans le domaine, notamment : AML, FCAMapX, LogMapBio, LogMap, LogMapLt, XMap, DOME, ALDO2Vec, POMAP+++ [Laadhar and al., 2017], KEPLER [Kachroudi and al., 2017].

1. Systematized Nomenclature of Medicine-Clinical Terms. http ://www.snomed.org/snomed-ct/
2. National Cancer Institute Thesaurus. https ://ncit.nci.nih.gov/
3. Foundational Model of Anatomy. http ://si.washington.edu/projects/fma
4. http ://www.cs.ox.ac.uk/isg/projects/SEALS/oaei/2018/
5. http ://gaia.fdi.ucm.es/grupo/projects/ontobridge/index.html

La méthode proposée est validée au moyen d'un vaste ensemble d'expériences sur de petits et de grands ensembles de données. Dans l'OAEI LargeBio Track 2018, sept systèmes ont été en mesure de réaliser les six tâches de l'ensemble de données.
Les résultats illustrés dans le tableau 5.1, montrent que ONTEM, avec la stratégie d'extraction, surpasse les autres grandes stratégies d'alignement des ontologies en termes de temps de traitement tout en conservant la même qualité.
Nous notons ici que ONTEM réalise un très bon temps d'exécution dans les six tâches.

Système	Tâche 1	Tâche 2	Tâche 3	Tâche 4	Tâche 5	Tâche 6	Moyenne	Tâches
LogMapLt	1	6	1	9	5	11	6	6
ONTEM	1	7	3	18	9	14	9	6
DOME	2	12	2	20	10	24	12	6
LAB	24	55	68	94	346	168	126	6
Xmap	7	65	26	299	124	427	158	6
LogMap	6	51	33	287	123	475	163	6
FCAMapX	40	881	91	1736	833	2377	993	6
LogMapBio	701	1072	890	1840	1500	2942	1491	6
POMAP+++ POMAP	254		779		6312		2448	3
ALOD2Vec	342		1400		9209		3650	3
KEPLER	588		4163				2376	2

TABLE 5.1 – *Durée*$(10 \times 3s)$ *d'exécution du système et achèvement des tâches*

Nous tenons à souligner que l'ONTEM a été en mesure d'accomplir les six tâches et même de se placer dans quatre tâches différentes au milieu de la liste des concurrents. Les résultats obtenus sont illustrés par les figures 5.1 à 5.6 et les tableaux 5.2 à 5.7, et discutés dans la section suivante.

5.3.1 Tâche 1 de LargeBio Track 2018

Système	Mappings	Précision	Rappel	F-Mesure
LAB	2723	0,958	0,910	0,933
FCAMapX	2828	0,948	0,911	0,929
LogMapBio	2776	0,941	0,902	0,921
LogMap	2747	0,944	0,897	0,920
ALOD2Vec	2528	0,972	0,839	0,910
KEPLER	2506	0,960	0,831	0,891
POMAP+++ POMAP	2414	0,979	0,814	0,889
LogMapLt	2480	0,967	0,819	0,887
Xmap	2315	0,977	0,783	0,869
ONTEM	2666	0.923	0.814	0.865
DOME	2248	0,985	0,764	0,861

TABLE 5.2 – *Tâche* 1. *FMA-NCI small fragments*

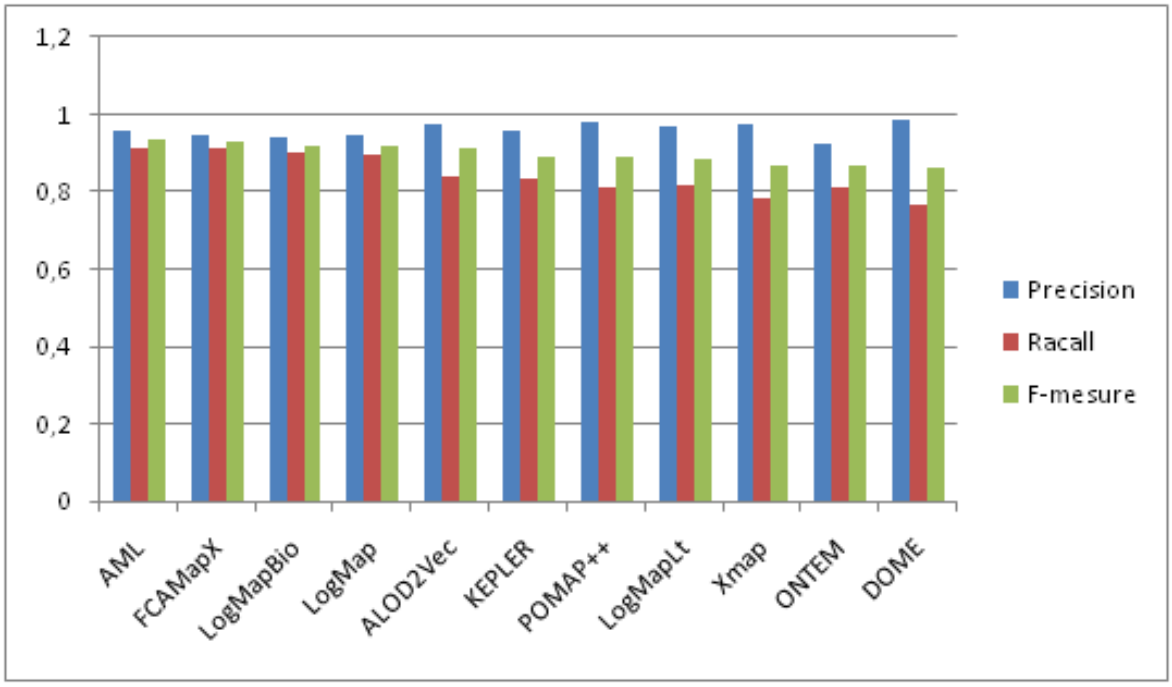

FIGURE 5.1 – *Résultats comparatifs FMA-NCI small fragments*

5.3.2 Tâche2 de LargeBio Track 2018

Système	Mappings	Précision	Rappel	F-Mesure
LAB	2968	0,838	0,872	0,855
LogMap	2701	0,856	0,808	0,831
LogMapBio	2860	0,830	0,831	0,830
Xmap	2415	0,878	0,742	0,804
FCAMapX	3607	0,665	0,841	0,743
LogMapLt	3458	0,676	0,819	0,741
DOME	2383	0,803	0,668	0,729
ONTEM	3371	0,662	0,738	0,698

TABLE 5.3 – *Tâche* 2. *FMA-NCI whole ontologies*

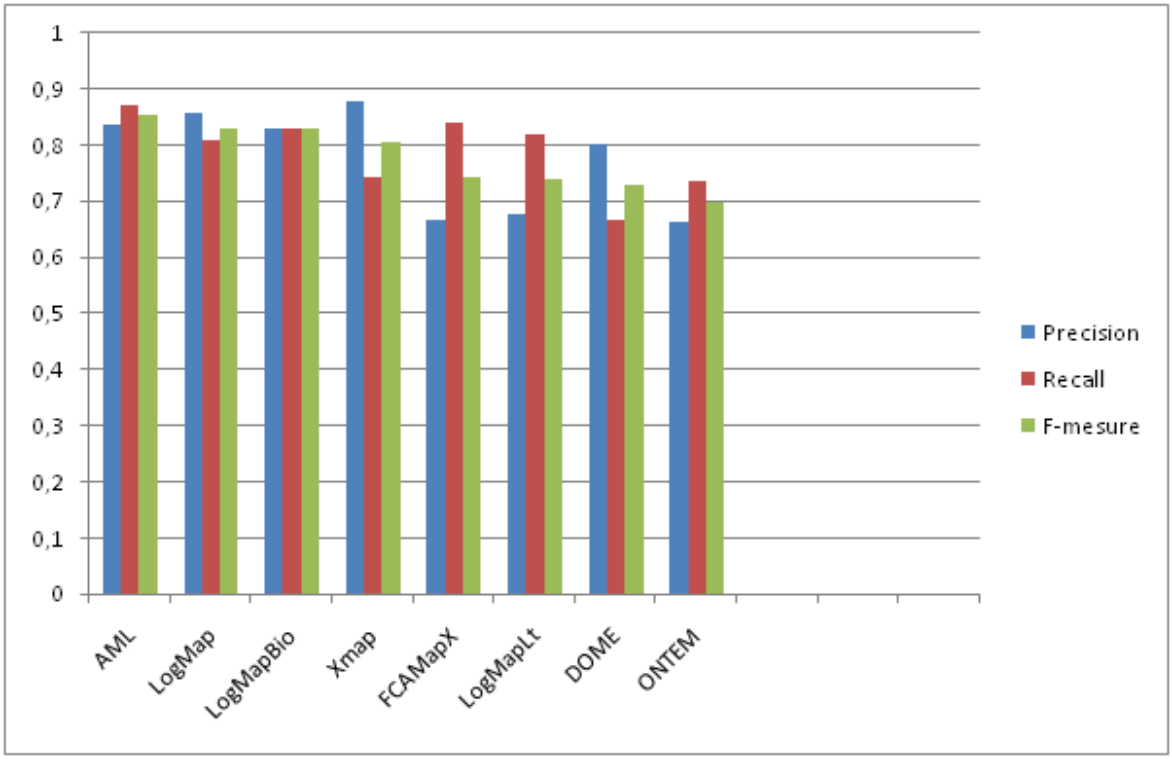

FIGURE 5.2 – *Résultats comparatifs : NCI Whole ontology with SNOMED large fragment*

5.3.3 Tâche 3 de LargeBio Track 2018

Système	# Mappings	Précision	Rappel	F-Mesure
FCAMapX	7582	0,955	0,815	0,879
LAB	6988	0,923	0,762	0,835
LogMapBio	6319	0,947	0,693	0,800
LogMap	6282	0,947	0,690	0,798
Xmap	5815	0,962	0,647	0,774
ONTEM	6387	0,914	0,647	0,758
KEPLER	4005	0,822	0,424	0,559
POMAP+++ POMAP	2163	0,906	0,260	0,404
ALOD2Vec	1727	0,941	0,213	0,347
LogMapLt	1642	0,968	0,208	0,342
DOME	1530	0,988	0,198	0,330

TABLE 5.4 – *Tâche 3. FMA-SNOMED small fragments*

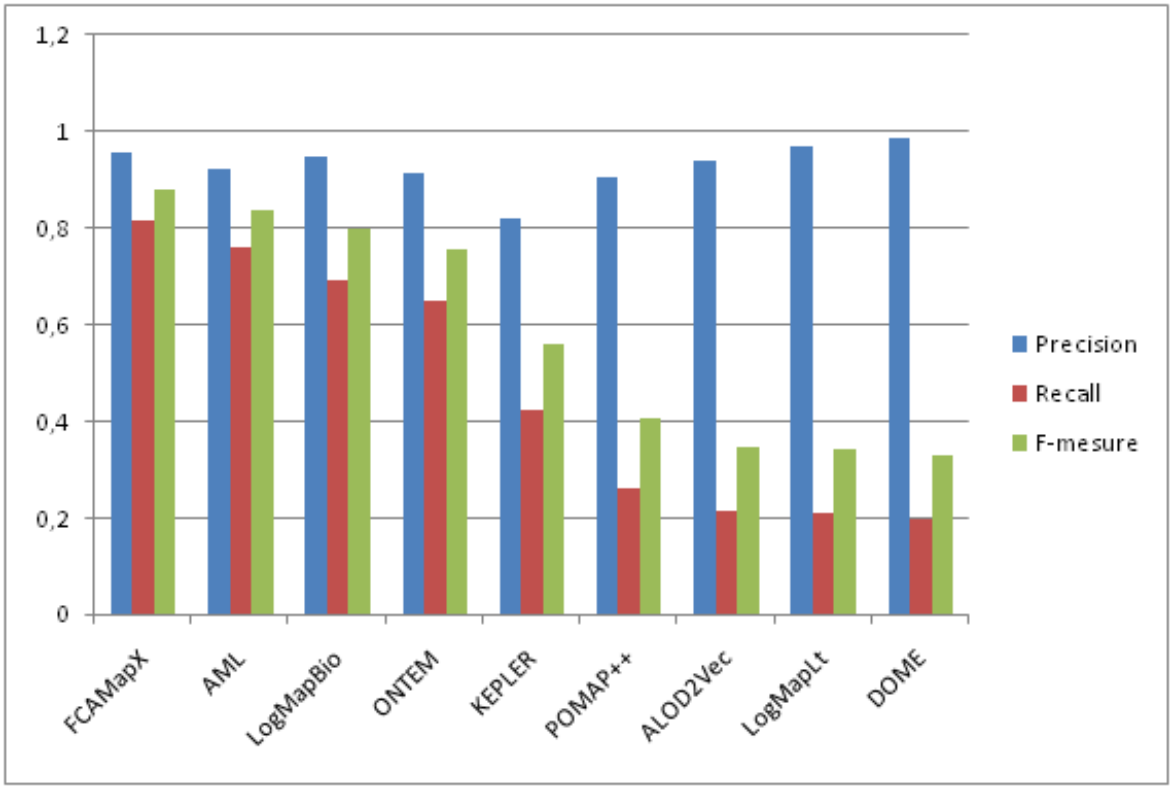

FIGURE 5.3 – *Résultats comparatifs FMA-SNOMED small fragments*

5.3.4 Tâche 4 de LargeBio Track 2018

Systâme	# Mappings	Précision	Rappel	F-Mesure
FCAMapX	7971	0,819	0,762	0,789
LAB	6571	0,882	0,687	0,772
LogMapBio	6471	0,834	0,650	0,731
LogMap	6393	0,840	0,645	0,730
ONTEM	6817	0,827	0,626	0,713
Xmap	6749	0,723	0,608	0,661
LogMapLt	1820	0,851	0,208	0,334
DOME	1588	0,941	0,197	0,326

TABLE 5.5 – *Tâche 4. : FMA Whole Ontology with SNOMED large fragments*

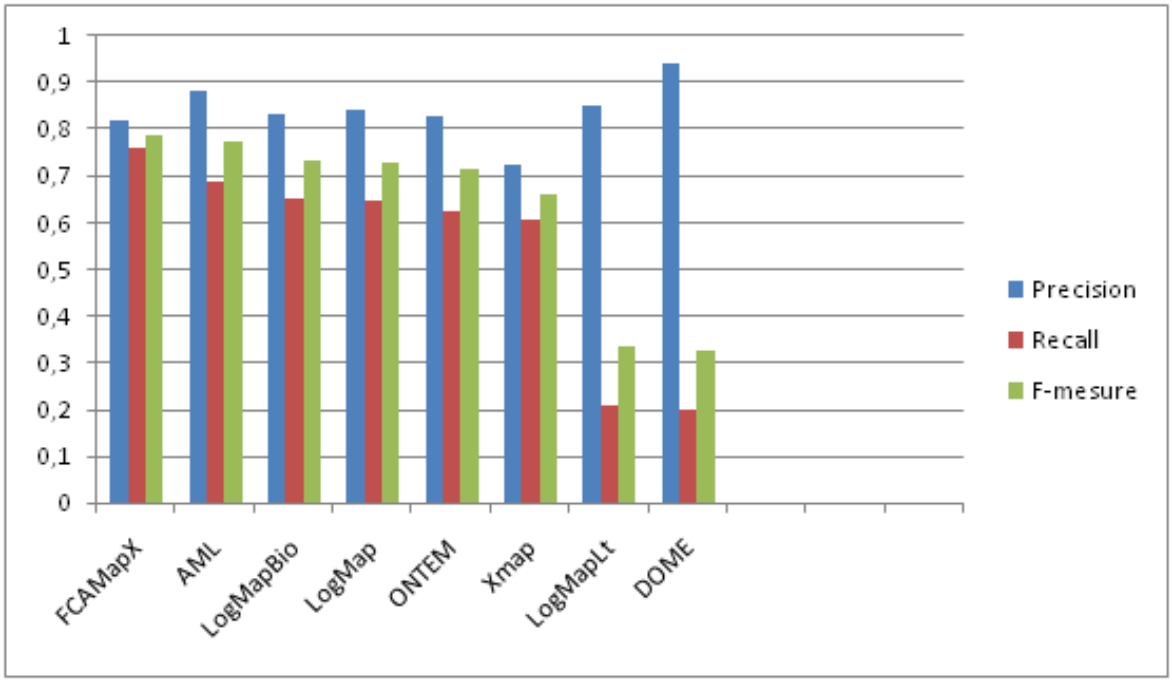

FIGURE 5.4 – *Résultats comparatifs : FMA Whole Ontology with SNOMED large fragments*

5.3.5 Tâche 5 de LargeBio Track 2018

Système	# Mappings	Précision	Rappel	Mesure F
LAB	14435	0,878	0,737	0,801
FCAMapX	13789	0,878	0,703	0,781
LogMapBio	12678	0,912	0,672	0,774
LogMap	12414	0,922	0,665	0,773
ONTEM	11958	0,921	0,585	0,715
LogMapLt	10921	0,893	0,566	0,693
Xmap	12125	0,835	0,588	0,690
POMAP+++ POMAP	10895	0,889	0,563	0,689
DOME	9321	0,922	0,499	0,648
ALOD2Vec	12882	0,743	0,556	0,636

TABLE 5.6 – *Tâche 5. SNOMED-NCI small fragments*

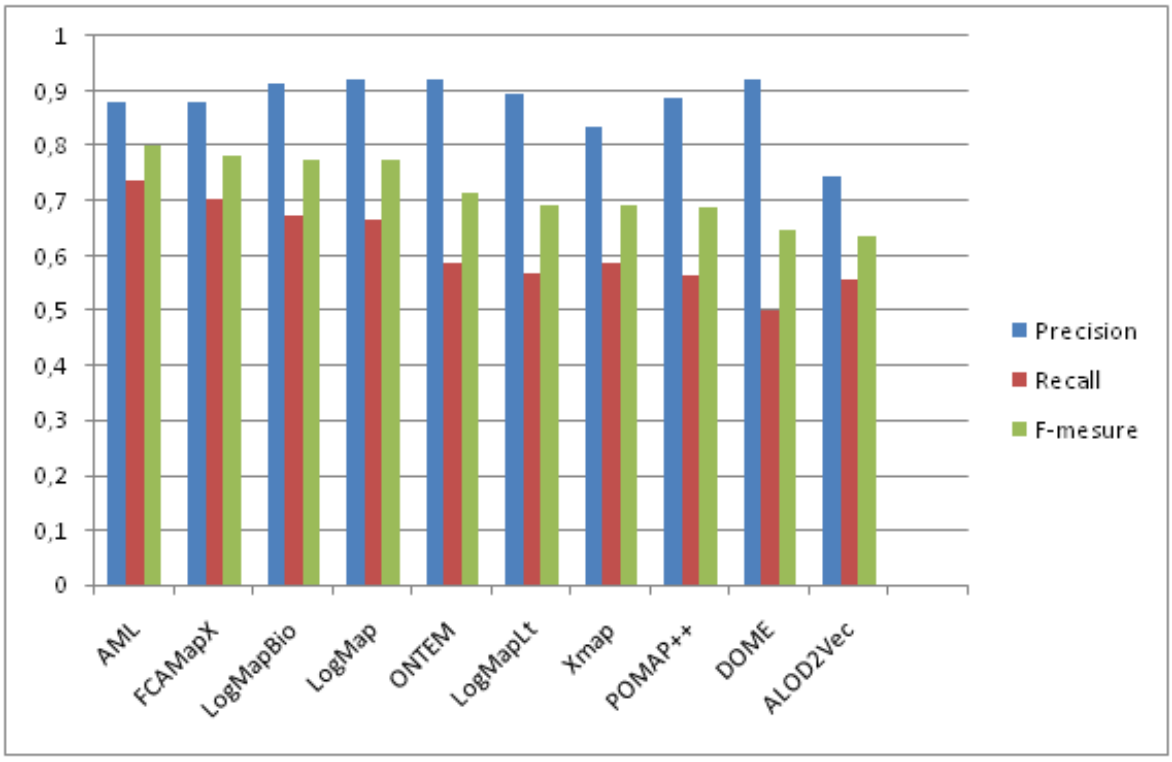

FIGURE 5.5 – *Résultats comparatifs SNOMED-NCI small fragments*

5.3.6 Tâche 6 de LargeBio Track 2018

Système	# Mappings	Précision	Rappel	Mesure F
LAB	13176	0,904	0,668	0,768
FCAMapX	15383	0,796	0,680	0,733
LogMapBio	13098	0,854	0,627	0,723
LogMap	12276	0,867	0,596	0,706
ONTEM	9971	0.893	0.531	0.666
LogMapLt	12864	0,798	0,566	0,662
DOME	9702	0,907	0,485	0,632
Xmap	16271	0,640	0,582	0,610

TABLE 5.7 – *Tâche 6. : Comparative results NCI Whole ontology with SNOMED large fragment*

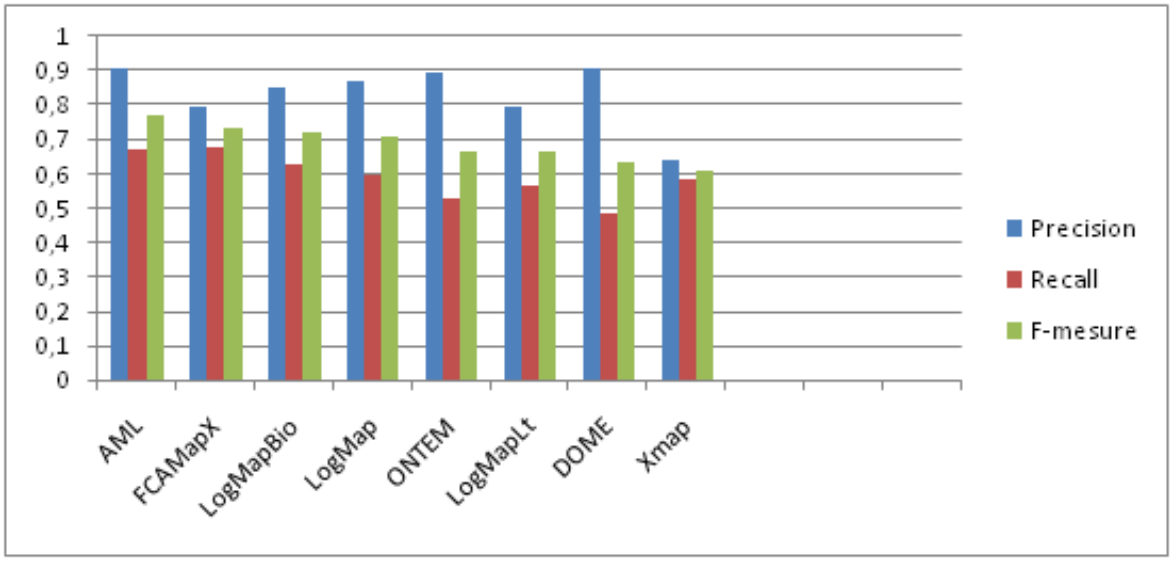

FIGURE 5.6 – *Résultats comparatifs NCI Whole ontology with SNOMED large fragment*

5.4 Discussion

Dans ce chapitre, nous avons élaboré une série d'expérimentations pour évaluer notre stratégie d'alignement des grandes ontologies OWL.

Nous avons montré que si la taille des ontologies d'entrée augmente, les méthodes actuelles sont obligées de décomposer les grandes ontologies en blocs ou en modules.

La stratégie ONTEM ne divise pas les ontologies en blocs ou en modules. La qualité des correspondances d'ONTEM est meilleure que les stratégies de partitionnement et de modularisation en ce qui concerne les mesures standard de précision et de rappel. De plus, lors de nos différents tests, nous avons constaté que ONTEM réalise une précision sémantique très prometteuse.

A titre d'exemple, nous avons trouvé que l'ancre (Outer_plexiform_layer_of_ retina, Outer_Plexiform_Layer) située au niveau de l'alignement "OAEI_FMA2NCI_ UMLS_ mappings_with_flagged_repairs.rdf" est sémantiquement moins précise que celle trouvée par ONTEM (Outer_plexiform_layer_of_retina, Structure_of_outer_plexiform_layer_of_retina). En effet, le mot-clé "retina" manque dans l'ancre donné par l'alignement du Benchmark LargeBio Track 2018. Nous notons que l'effet de ces absences de mots-clés a une influence significative sur l'échelle de l'évaluation des deux alignements.

De plus, les méthodes actuelles de calcul des similarités chargent les deux grandes ontologies à traiter en mémoire. Elles sont confrontées à des difficultés de gestion de la mémoire et généralement, le message d'exception "Java Heap Space" apparaît. Nous avons montré que le processus d'alignement a une complexité spatiale de O(n). ONTEM évite l'erreur de dépassement de mémoire dans le cas d'une grande valeur de n (la valeur maximale de n est fonction de la taille maximale de la mémoire autorisée par le système).

D'autre part, la complexité temporelle globale du processus d'alignement de la méthode ONTEM est $O(n \times t)$. Par conséquent, les utilisateurs n'attendraient pas très longtemps par rapport aux autres méthodes. Nous avons comparé la méthode d'alignement ONTEM avec les résultats obtenus selon OAEI LargeBio Track 2018, correspondant aux modalités d'évaluation de la plate-forme SEALS.

Plusieurs observations au sujet de ces résultats ont été soulignées, y compris l'incidence de l'élimination de tout calcul des mesures de similarité dans les délais de traitement. Nous avons montré dans les runtimes du système que ONTEM est au deuxième rang. Nous soulignons que ONTEM peut accomplir les six tâches, et même se placer dans quatre tâches différentes au milieu de la liste des concurrents.

ONTEM n'effectue aucun calcul de similarité. Aucune opération de comparaison n'est effectuée entre les mots situés dans les entités (nom de classe ou labels) car nous utilisons l'opération de tri. Le temps d'alignement est considérablement réduit. Les résultats que nous avons obtenus sont plus que satisfaisants. Le spécialiste en linguistique peut étendre la liste de mots de composition que nous avons dressé afin d'améliorer les résultats de la méthode ONTEM.

Nous avons obtenu des résultats prometteurs par rapport aux alignements de référence contenus dans la section LargeBio Track 2018. Nous montrons à travers ONTEM que même si les ontologies sont volumineuses, les paramètres de précision et de F-mesure augmentent. Il n'est donc pas nécessaire de limiter la taille des ensembles de concepts à l'entrée de l'outil d'alignement. ONTEM est une méthode entièrement automatique et ne nécessite aucune intervention de l'utilisateur pendant le processus d'alignement. Elle permet aux utilisateurs non experts d'aligner facilement toutes les ontologies écrites en langage OWL.

Nous notons à ce niveau, que l'utilsation de WordNet dans un environnement spécialisé tel que les ontologies biomédicales est déconseillé. En effet, WordNet est une ressource lexi-

cale généraliste. Nous pensons utiliser à la place de WordNet, le lexique spécialisé UMLS Specialist [Bodenreider ,2004].
Nous sommes convaincus que ONTEM fournira une nouvelle base pour toutes les autres méthodes basées sur des calculs de similarité. En effet, ces calculs ne concerneront que des concepts non traités par notre méthode. Les différents systèmes seront dispensés de ces calculs et les délais de traitement seront donc réduits.

5.5 Conclusion

Le prototype ONTEM traite les 6 tâches du benchmark OAEI LargeBio Track 2018 (voir annexes). Le prototype reste à améliorer . Nous prévoyons une interface graphique pour faciliter au maximum l'intervention des utilisateurs.

Conclusion et Perspectives 6

Dans cette thèse nous avons proposé une stratégie originale d'intéropérabilité des systèmes d'information à base d'alignement des ontologies (ONTEM). Nous avons montré que notre stratégie assure l'intéropérabilité syntaxique et sémantique. La taille des ontologies n'est plus un handicap pour assurer celle-ci.
Pour ce faire, nous nous sommes concentrés sur la question de l'alignement ontologique à grande échelle pour le Web sémantique. En effet, la variété des ontologies d'un même domaine dans le web sémantique a conduit à l'hétérogénéité et donc au développement des méthodes d'alignement ontologique. Depuis plus d'une vingtaine d'annÃ©es, les méthodes d'alignement ontologique tentent de résoudre les problèmes d'hétérogénéité et de correspondance ontologique. Aujourd'hui, dans de nombreuses applications réelles comme dans le domaine médical, la taille des ontologies est très importante et les méthodes d'alignement actuelles sont confrontées à de nombreux défis tels que le manque de mémoire et des longs temps de traitement. Nous avons montré que notre méthode ONTEM se distingue des méthodes existantes par son originalité. Aucune opération de décomposition n'est effectuée, la conservation sémantique est donc assurée. De plus, aucun calcul de similarité n'est nécessaire avec la méthode ONTEM. Elle est basée uniquement sur des opérations algébriques. La méthode ONTEM est validée au moyen d'un vaste ensemble d'expériences sur de petits et de grands ensembles de données contenus dans l'OAEI LargeBio Track 2018. Le prototype développé confirme la validité de la méthode ONTEM. Il permet de construire de nouvelles architectures à partir des méthodes existantes qui permettront à ONTEM d'obtenir de meilleurs résultats car l'objectif de tous ces travaux est de faciliter l'interopérabilité des systèmes d'information ontologiques. À cette fin, le présent document propose une solution appropriée à ce type de problème.
Un prototype a été mis au point pour appuyer l'approche proposée. Ce prototype nous a permis d'évaluer nos commentaires et de les comparer avec d'autres méthodes reconnues dans ce domaine, comme la section LargeBio Track de l'OAEI de la campagne 2018. ONTEM est un travail en cours, qui, grâce à son idée innovante, pourra donner de nouvelles pistes de recherche.
Dans nos travaux futurs, nous prévoyons consolider notre méthode afin de mieux soutenir l'alignement des ontologies à grande échelle. Nous avons déjà commencé à nous pencher sur cette question, mais la mise à jour de la base de données de tests pose d'autres défis, en termes de langages ontologiques utilisés et de formalisme sémantique de description en évolution. De plus, pour réduire le temps d'exécution de l'alignement ontologique à grande échelle, la tâche de prétraitement des deux ontologies peut être réalisée de manière parallèle.

Annexes 7

7.1 Prototype ONTEM

Nous présentons ci-dessous les fenêtres de chargement des ontologies et opération mesure de l'alignement [Zerhouni and Benslimane,2019b].

7.1.1 Chargement des ontologies OWL

Figure 7.1 – *Chargement des ontologies OWL*

7.1.2 Opération mesure de l'alignement

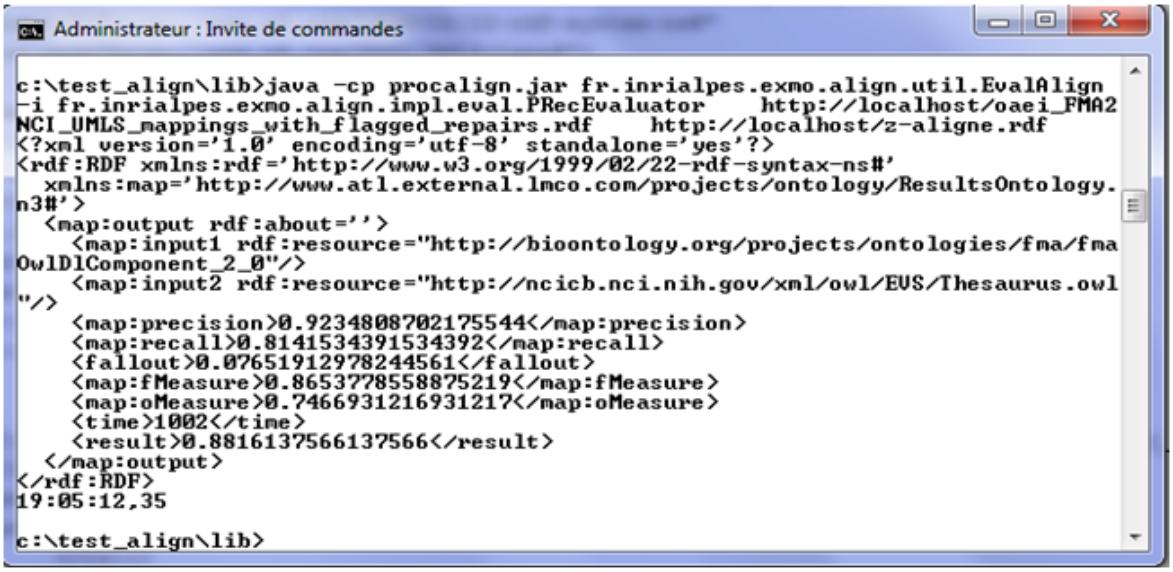

FIGURE 7.2 – *Opération mesure de l'alignement*

7.1.3 Extraits des ontologies fma_small_overlapping_nci.owl et nci_small_overlapping_fma.owl

Extrait de l'ontologie oaei_fma_small_overlapping_nci.owl

```
<?xml version="1.0" ?>
<!DOCTYPE rdf :RDF [
<!ENTITY owl "http ://www.w3.org/2002/07/owl" >
<!ENTITY xsd "http ://www.w3.org/2001/XMLSchema" >
<!ENTITY rdfs "http ://www.w3.org/2000/01/rdf-schema" >
<!ENTITY rdf "http ://www.w3.org/1999/02/22-rdf-syntax-ns" >
]>
<rdf :RDF xmlns="http ://bioontology.org/projects/ontologies/fma/
fmaOwlDlComponent_2_0#"
<!–
////////////////////////////////////////
/
//
// Classes
//
////////////////////////////////////////
/
–>
<!– http ://bioontology.org/projects/ontologies/fma/fmaOwlDlComponent_2_0 #Em-
bryonic_cell –>

<owl :Class
rdf :about="http ://bioontology.org/projects/ontologies/fma/fmaOwlDl
Component_2_0#Embryonic_cell">

<rdfs :label xml :lang="en">Embryonic cell</rdfs :label>
<rdfs :subClassOf
rdf :resource="http ://bioontology.org/projects/ontologies/fma/fmaOwlDl
Component_2_0#Embryonic_structure"/>
</owl :Class>

<!– http ://bioontology.org/projects/ontologies/fma/fmaOwlDlComponent
_2_0#Abducens_nerve –>
<owl :Class
rdf :about="http ://bioontology.org/projects/ontologies/fma/fmaOwlDl
Component_2_0#Abducens_nerve">

<rdfs :label xml :lang="en">Abducens nerve</rdfs :label>
<rdfs :label xml :lang="en">Abducens nerve [VI]</rdfs :label>
<rdfs :label xml :lang="en">Abducens nerve tree</rdfs :label>
<rdfs :label xml :lang="en">Abducent nerve</rdfs :label>
```

```
<rdfs :label xml :lang="en">Abducent nerve [VI]</rdfs :label>
<rdfs :label xml :lang="en">Lateral rectus nerve</rdfs :label>
<rdfs :label xml :lang="en">Nervus abducens</rdfs :label>
<rdfs :label xml :lang="en">Sixth cranial nerve</rdfs :label>
<rdfs :label xml :lang="en">Nerve VI</rdfs :label>
<rdfs :label xml :lang="en">Nervus abducens [VI]</rdfs :label>
<rdfs :subClassOf
rdf :resource="http ://bioontology.org/projects/ontologies/fma/fmaOwlDl
Component_2_0#Cranial_nerv
```

Extrait de l'ontologie oaei_nci_small_overlapping_fma.owl

```
<?xml version="1.0" ?>
<!DOCTYPE rdf :RDF [
<!ENTITY owl "http ://www.w3.org/2002/07/owl" >
<!ENTITY xsd "http ://www.w3.org/2001/XMLSchema" >
<!ENTITY rdfs "http ://www.w3.org/2000/01/rdf-schema" >
<!ENTITY rdf "http ://www.w3.org/1999/02/22-rdf-syntax-ns" >
<!ENTITY Thesaurus "http ://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl" >
]>
<rdf :RDF xmlns="http ://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl"
xml :base="http ://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl"
xmlns :Thesaurus="http ://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl"
xmlns :rdfs="http ://www.w3.org/2000/01/rdf-schema"
xmlns :owl="http ://www.w3.org/2002/07/owl"
xmlns :xsd="http ://www.w3.org/2001/XMLSchema"
xmlns :rdf="http ://www.w3.org/1999/02/22-rdf-syntax-ns">
<owl :Ontology rdf :about="http ://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus
.owl"/>
<!– ////
//////////////////////////////////
//
// Classes
//
//////////////////////////////////
–>
<!– http ://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owlBlastomere –>
<owl :Class rdf :about="http ://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl
#Blastomere">
<rdfs :label xml :lang="en">Blastocytes</rdfs :label>
<rdfs :label xml :lang="en">Blastomere</rdfs :label>
<rdfs :label xml :lang="en">Blastomeres</rdfs :label>
<rdfs :label xml :lang="en">Embryo stage 2</rdfs :label>
<rdfs :subClassOf
rdf :resource="http ://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owlEmbryonic_
Cell"/>
```

```
<rdfs :subClassOf
rdf :resource="http ://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#Totipotent
_Stem_Cell"
/>
<rdfs :subClassOf>
<owl :Restriction>
<owl :onProperty
rdf :resource="http ://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#Anatomic
_Structure_Is
_Physical_Part_Of"/>
<owl :someValuesFrom
rdf :resource="http ://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owlBlastocyst"
/>
</owl :Restriction>
</rdfs :subClassOf>
</owl :Class>
<!– http ://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owlEmbryonic_Cell –>
<owl :Class
rdf :about="http ://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#Embryonic
_Cell">
<rdfs :label xml :lang="en">Blastomere</rdfs :label>
<rdfs :label xml :lang="en">Embryonic Cell</rdfs :label>
<rdfs :label xml :lang="en">Embryonic Cells</rdfs :label>
<rdfs :subClassOf
rdf :resource="http ://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#Cell"/>
<rdfs :subClassOf
rdf :resource="http ://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owlEmbryonic
structure"/ >
<rdfs :subClassOf>
<owl :Restriction>
<owl :onProperty
rdf :resource="http ://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#Anatomic _Struc-
ture_Is_Physical_Part_Of"/>
<owl :someValuesFrom
rdf :resource="http ://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#Embryonic
_Tissue"/>
</owl :Restriction>
</rdfs :subClassOf>
</owl :Class>
```

7.1.4 Extrait de l'alignement généré.

```
<?xml version="1.0" encoding="utf-8" standalone="no" ?>
<rdf :RDF xmlns="http ://knowledgeweb.semanticweb.org/heterogeneity/
alignment"
xml :base="http ://knowledgeweb.semanticweb.org/heterogeneity/alignment"
xmlns :rdf="http ://www.w3.org/1999/02/22-rdf-syntax-ns"
xmlns :xsd="http ://www.w3.org/2001/XMLSchema">
<Alignment>
<xml>yes</xml>
<level>0</level>
<type>**</type>
<time>1002</time>
<method>fr.inrialpes.exmo.align.impl.method.StringDistAlignment</method>
<onto1>http ://localhost/oaei_fma.owl</onto1>
<onto2>http ://localhost/oaei_nci.owl</onto2>
<uri1>http ://bioontology.org/projects/ontologies/fma/fmaOwlDlComponent
_2_0</uri1>
<uri2>http ://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl</uri2>
<map>
<Cell>
<entity1
rdf :resource="http ://bioontology.org/projects/ontologies/fma/fmaOwlDl
Component_2_0#B
lastomere"/>
<entity2 rdf :resource="http ://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl
#Embryonic _Cell"/>
<measure rdf :datatype="http ://www.w3.org/2001/XMLSchemafloat">1.0
</measure>
<relation>=</relation>
</Cell>
</map>
<map>
<Cell>
<entity1
rdf :resource="http ://bioontology.org/projects/ontologies/fma/fmaOwlDl
Component_2_0#Brachial_artery"/>
<entity2
rdf :resource="http ://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl
Brachial_Artery"/>
<measure rdf :datatype="http ://www.w3.org/2001/XMLSchem#float">1.0
</measure>
<relation>=</relation>
</Cell>
</map>
```

Bibliographie

[Academie française,2005] Dictionnaire de l'Académie française, neuvième édition. Journal officiel, Paris, 2005. 13

[Anam and al., 2014] Anam, S., Kim, Y., Kang, B., & Liu, Q. Evaluation of Terminological Schema Matching and Its Implications for Schema Mapping. Proceedings of the PRICAI 2014 : Trends in Artificial Intelligence. Springer International Publishing. 107

[Angele and Shnurr,2005] Jürgen ANGELE et Hans-Peter SCHNURR. Do not use this gear with a switching lever ! automotive industry experience with semantic guides. In GI Jahrestagung (1), pages 48 – 52, 2005. 44, 49, 50, 114

[Annane and al., 2018] Annane, A., Bellahsene, Z., Azouaou, F., & Jonquet, C. Building an effective and efficient background knowledge resource to enhance ontology matching. Journal of Web Semantics, 51, 51 – 68. doi :10.1016/*j*. websem.2018.04.001

[Ardjani and al.,2015] Ardjani, F., Bouchiha, D., & Malki, M. Ontology-Alignment Techniques : Survey and Analysis. International Journal of Modern Education and Computer Science, 7(11), 67 – 78. doi :10.5815/ijmecs.2015.11.08 59

[Bachimont,2000] Bruno BACHIMONT. Engagement Sémantique et Engagement Ontologique : Conception et Réalisation D'ontologies En Ingénierie Des Connaissances, chapter 19, pages 305 – 324. Eyrolles, 2000. 30

[Barwise and Seligman,1997] Jon BARWISE et Jerry SELIGMAN. Information Flow : The Logic of Distributed Systems. Number 44. Cambridge University Press, 1997. 46

[Baeza and Ribeiro, 1999] R.Baeza-Yates et B.Ribeiro-Neto. Modern Information Retrieval. ACM Press ; Addison-Wesley : New York ; Harlow, England ; Reading, Mass., 1999. 103

[Benoit ,2005] V. BENOIT :Modularisation et intégration d'ontologies dans le domaine de la bioinformatique, thèse de Master, univrsité du QUEBEC à Montréal. 2005 86, 87

[Berners-Lee and all,2001] J. Hendler, O. Lassila, "The Semantic Web", Scientic American 284, (2001), *pp*.34 – 43. 2

[Bodenreider ,2004] Bodenreider O. The Unified Medical Language System (UMLS) : integrating biomedical terminology. Nucleic Acids Res. 2004; 32(Database issue) :267 – 70. 135

[Borgida and Giunchiglia, 2007] Borgida, A. et Giunchiglia, F. Importing from functional knowledge basesa preview. Dans B. C. Grau, V. Honavar, A. Schlicht , et F. Walter (dir.) . Proceedings of the 2*nd* International Workshop on Modular Ontologies, WoMO 2007, volume 315 de CEUR Vvorkshop Proceedings, Whistler, Canada. 95

[Bouquet and al.,2004] Paolo BOUQUET, Fausto GIUNCHIGLIA, Frank van HARMELEN, Luciano SERAFINI et Heiner STUCKENSCHMIDT. Contextualizing ontologiesstar, open. Web Semantic : Science, Services and Agents on the World Wide Web, 1(4) : 325343, 2004. 44, 49

[Brisson,2004] Laurent BRISSON. Mesures d'intérêt subjectif et représentation des connaissances. Rapport technique, Laboratoire I3S, Université Sophia Antipolis, Nice (France), Octobre 2004. 21

[Bruijn and al.,2006] Jos de BRUIJN, Marc EHRIG, Cristina FEIER, Francisco MARTIN-RECUERDA, FrançoisSCHARFFE et Mortiz WEITEN. Semantic Web Technologies, trends and research in ontology-based systems, chapter Ontology Mediation, Merging, and Aligning, pages 95113. WILEY, 2006. 42

[Charlet and al.,2000] Jean CHARLET, Manuel ZACKLAD, Gilles KASSEL et Didier BOURIGAULT. Ingénierie des connaissances : Evolution récentes et nouveaux défis. Eyrolles, 2000. systems. In IIAI'05, 2005. 33

[Charlet and al.,2002] CHARLET, CORDONNIER et GIBAUD. Interopérabilité en médecine : quand le contenu interroge le contenant et l'organisation. Information - Interaction - Intelligence, 2(2) : 37 − 62, 2002. 36

[Cheatham and Hitzler, 2013] Cheatham, M., & Hitzler, P. String similarity metrics for ontology alignment. In The Semantic WebâISWC 2013. Springer. 108

[D'Aquin and al.,2006] D'Aquin, M., Sabou, M., & Motta, E. Modularization : A Key for the Dynamic Selection of Relevant Knowledge Components. Proceedings of the 1*st* International Workshop on Modular Ontologies. Academic Press. vii, 88

[D'Aquin and al., 2007] d'Aquin, M., Doran, P. , Motta, E. et Tamma, V. Towards a parametric ontology modularization framework based on graph transformation. Dans B. C. 95

[D'Aquin and al., 2009] d'Aquin, M., Schlicht, A. , Stuckenschmidt, H. et Sabou, M. Criteria and evaluation for ontology modularization techniques. Dans Modular Ontologies, volume 5445 de Lecture Notes in Computer Science, 67 − 89. Springer-Verlag. 60

[Diallo, 2014] Diallo, G. An effective method of large scale ontology matching. Journal of Biomedical Semantics, 5(1), 44.*doi* : 10.1186/2041 − 1480 − 5 − 44 114

[Ding and al., 2013] G. Ding, T. Sun, Y. Xu, âMulti-Schema Matching Based On Clustering Techniques. In the 10*th* International Conference on Fuzzy Systems and Knowledge Discovery (FSKD). 2013. 83

[Doan and al., 2003] AnHai DOAN, Jayant MADHAVAN, Pedro DOMINGOS et Alon HALEVY. Ontology matching : A machine learning approach. 2003. 54, 55

[Doran and al., 2007] Doran P., Tamma V., Iannone L. Ontology module extraction for ontology reuse : An ontology engineering perspective. In Proceedings of the sixteenth acm conference on conference on information and knowledge management, p. 61 − 70. New York, NY, USA, ACM. 85, 89, 90

[Doran and al., 2008] Doran, P., Palmisano, I. et Tamma, V. Somet : algorithm and tool for sparql based ontology module extraction. Dans U. Sattler et A. Tamilin (dir.) . Proceedings of the 2008 ESWC International Workshop on Ontologies : Reasoning and Modularity (WORM- 08), volume 348 de CEUR Workshop Proceedings, Tenerife, Spain. vii, 95, 97

[Dou and al.,2002] Dejing DOU, Drew MCDERMOTT et Peishen QI. Ontology translation by ontology merging and automated reasoning, 2002. 51, 52

[Dupont and al., 2006] Dupont, P., Callut, J., Dooms, G., Monette, J. et Deville, Y. Relevant subgraph extraction from random walks in a graph. Rapport technique, Université catholique de Louvain, UCL/ INGI. 101

[Ehrig , 2007] Ehrig, M. Ontology Alignment : Bridging the Semantic Gap. Semantic Web and Beyond Computing for Human Experience $(pp.1250)$. Springer. 113

[Ehrig and Staab,2004] Marc EHRIG et Steffen STAAB. Qom quick ontology mapping. In International Semantic Web Conference $(ISWC2004)$, Japan, November 2004. 53, 54, 56

[Enric and al.,2000] Motta ENRIC, BUCKINGHAM et DOMINGUE. Ontology-driven document enrichment :principles, tools and applications. $52(6) : 1071 - 1109$, June 2000. 24

[Euzenat, 2004] Jérôme Euzenat. An API for Ontology Alignment. In The Semantic Web - ISWC 2004 : Third International Semantic Web Conference,Hiroshima, Japan, November $7 - 11, 2004$. Proceedings, pages 698_712.184 105, 106

[Euzenat and Valtchev, 2004] Euzenat, J., & Valtchev, P. Similarity-based ontology alignment in OWL-lite. Proceedings of the European Conference on Artificial Intelligence (ECAI) $(pp.333337)$. Academic Press. 3, 108

[Euzenat and Shvaiko,2013] Euzenat, J. ,Shvaiko, P. J. Ontology matching : State of the art and future challenges. IEEE Transactions on Knowledge and Data Engineering, $25(1), 158 - 176$. doi :$10.1109/TKDE.2011.253$ 4, 6

[Feldman and al.,2005] FELDMAN, DUMONTIER, LING, HAIDER et HOGUE. Co : A chemical ontology for identification of functional groups and semantic comparison of small molecules. FEBS Lett, $579(21) : 468591, 2005$. 30

[Fernandez and al.,1997] Mariano FERNANDEZ, Asuncion GOMEZ-PEREZ et Natalia JURISTO. Methontology : from ontological art towards ontological engineering. In Proceedings of the AAAI97 Spring Symposium Series on Ontological Engineering, pages 3340, Stanford, USA, March 1997. 21, 22

[Flatter,2003] David FLATER. Sumo2loom documentation.2003 24

[Fürst,2002] Frédéric FÜRST. L'ingénierie ontologique. Rapport technique, Institut de recherche en Informatique de Nantes, 2002. 15, 30

[Furst,2004] Frédéric FÜRST. Contribution à l'ingénierie des ontologies : une méthode et un outil d'opérationnalisation. Thèse d'Informatique, Université de Nantes. 86

[Gaëlle,2002] Lortal GAËLLE. État de lâart ontologies et intégration/fusion dâontologies, 2002. 17

[Giunchiglia and al.,2005] Fausto GIUNCHIGLIA, Pavel SHVAIKO et Mikalai YATSKEVICH. S-match : an algorithm and an implementation of semantic matching. In Y. KALFOGLOU, M. SCHORLEMMER, A. SHETH, S. STAAB et M. USCHOLD, réds., Semantic Interoperability and Integration, number 04391 in Dagstuhl Seminar Proceedings. Internationales Begegnungs und Forschungszentrumfuer Informatik (IBFI), Schloss Dagstuhl, Germany, 2005. 53, 54, 55

[Gómez-Pérez and al., 1995] Asunciòn GÒMEZ-PÉREZ, Natalia JURISTO et Juan PAZOS. Evaluation and assessment of the knowledge sharing technology. In Towards very large knowledge bases, pages $289 - 296$. IOS Press, 1995. 22, 31

[Gómez-Pérez, 1999] Gòmez-Pérez, A. Ontological engineering : A state of the art. Expert Update : Knowledge-Based Systems and Applied Artificial Intelligence, 2(3), 3343. 82, 108

[Guha and al., 1999] . Guha, R.Rastogi, and Shim, K. ROCK, -A Robust Clustering Algorithm for Categorical Attributes| |. In Proceedings of the 15*th* International Conference on Data Engineering (Sydney, Australia. March 23 – 261999.

[Guha and al.,2000] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. ROCK : A Robust Clustering Algorithm for Categorical Attributes. Information Systems, 25 : 345_366.185 66, 67

[Grau and al.,2005] Grau Bernardo Cuenca, Parsia Bijan, Sirin Evren and Kalyanpur Aditya. Automatic Partitioning of OWL Ontologies Using $\epsilon - Connections$. In : Horrocks Ian, Sattler Ulrike and Wolter Frank. Proceedings of the 2005 International Workshop on Description Logics $(DL2005)$, 26 – 28 July, 2005 Edinburgh, Scotland, UK. (CEUR Workshop Proceedings, 147). 61, 62, 65, 84

[Grau and al., 2006] Bernardo Cuenca Grau, Bijan Parsia, Evren Sirin, and Aditya Kalyanpur. Modularity and Web Ontologies. In Patrick Doherty, John Mylopoulos, and Christopher A. Welty (eds), Proceedings of *KR*2006 : the 20*th* International Conference on Principles of Knowledge Representation and Reasoning, Lake District, UK, June 2_5, 2006, pages 198_209. 61, 65, 66

[Grau and al., 2007a] Grau, B. C., Horrocks, I., Kazakov, Y. et Sattler, U. (2007). Just the right amount : Extracting modules from ontologies. Dans Proceedings of the 16*th* International Conference on World Wide Web, 717 – 726. , Banff, Alberta, Canada. ACM. 94

[Grau and al., 2007b] Grau, B. C., Horrocks, I., Kazakov, Y., & Sattler, U. A logical framework for modularity of ontologies. Proceedings of the IJCAI International Joint Conference on Artificial Intelligence $(pp.298303)$. Academic Press.

[Grau and al., 2008] Grau, B. C., Horrocks, I., Kazakov, Y., & Sattler, U. Modular reuse of ontologies : Theory and practice. Journal of Artificial Intelligence Research, 31, 273318. doi :10.1613/jair.2375 94

[Grau and al.,2009] Grau, B. C., Horrocks, I., Kazakov, Y. et Sattler, U. Extracting modules from ontologies : A logic-based approach. In M odular Ontologies 159 – 186. Springer. 94

[Gruber,1993] Thomas GRUBER. A translation approach to portable ontology specifications. Knowledge Acquisition, 5(2) : 199220, 1993. 14, 24, 37

[Gruber and Olsen,1994] Thomas GRUBER et Gregory OLSEN. An ontology for engineering mathematics. In KR, pages 258269, 1994. 30

[Guarino and Giaretta,1995] Nicola GUARINO et Pierdaniele GIARETTA. Ontologies and knowledge bases : Towards a terminological clarification. In N MARS, réd., Towards Very Large Knowledge Bases : Knowledge Building and Knowledge Sharing, pages 2532. IOS Press, 1995. vii, 14, 28, 29

[Hamdi and al., 2009a] FayÃ§al Hamdi, Brigitte Safar, Nobal B. Niraula, and Chantal Reynaud. TaxoMap in the OAEI 2009 Alignment Contest. In Proceedings of the 4*th* International Workshop on Ontology Matching $(OM - 2009)$ collocated with the 8*th* International Semantic Web Conference $(ISWC - 2009)$ Chantilly, USA, October 25, 2009. 59, 120

[Hamdi and al., 2009b] Fayçal Hamdi, Brigitte Safar, Chantal Reynaud, and Haïfa Zargayouna. Alignment-Based Partitioning of Large-Scale Ontologies. In Fabrice Guillet, Gilbert Ritschard, Djamel Abdelkader Zighed, and Henri Briand (eds), EGC (best of volume), volume 292 of Studies in Computational Intelligence, pages 251_269. Springer. 61, 62, 67

[Hamdi and al., 2009c] FayÃ§al Hamdi, Brigitte Safar, Haïfa Zargayouna, and Chantal Reynaud. Partitionnement d'ontologies pour le passage à l'échelle des techniques d'alignement. In Jean-Gabriel Ganascia and Pierre GanÃ§arski (eds), EGC, volume RNTI-E-15 of Revue des Nouvelles Technologies de l'Information, pages 409_420. Cépaduès-Éditions. 61, 62, 67

[Hartung and al.,2012] Hartung, M., Gross, A., Kirsten, T., & Rahm, E. Effective mapping composition for biomedical ontologies. eProcceedings of Semantic Interoperability in Medical Informatics ESWC. Academic Press.

[Horridge and Bechhofer,2011] Horridge, M., & Bechhofer, S. The OWL API : A Java API for OWL ontologies. Semantic Web, 2(1), 1121. International Journal of Strategic Information Technology and Applications 126

[Hu and al.,2006a] Hu Wei and Qu Yuzhong. Block Matching for Ontologies. In : F. Cruz Isabel, Decker Stefan, Allemang Dean, Preist Chris, Schwabe Daniel, Mika Peter, Uschold Michael and Aroyo Lora. In : Proceedings of the $5th$ International Semantic Web Conference, $5-9$ November, 2006, Athens, GA, USA. Springer, 2006, $pp300-313$ (Lecture Notes in Computer Science, 4273). 59, 61, 62

[Hu and al.,2006b] Hu Wei, Zhao Yuanyuan and Qu Yuzhong. Partition-Based Block Matching of Large Class Hierarchies. In : Mizoguchi Riichiro, Shi Zhongzhi and Giunchiglia Fausto. Proceedings of the First Asian Semantic Web Conference, $3-7$ September, 2006, Beijing, China. Springer, $pp72-83$ (Lecture Notes in Computer Science, 4185). 59, 60, 66, 67, 82

[Hu and al.,2008] Hu Wei, Qu Yuzhong and Cheng Gong. Matching large ontologies : A divideand- conquer approach. In journal of Data Knowledge Engineering, 2008, vol.67, $n°1$, pp $140-160$. 59, 61

[Huang and Lai, 2006] X.Huang, W.LAI. Clustering graphs for visualization via node similarities Journal of Visual Languages and Computing 17(2006)225253 82

[Inaba and al., 2000] INABA, SUPNITHI, IKEDA, MIZOGUCHI et TOYODA. An overview of "learning goal ontology". In ECAI2000 Workshop on Analysis and Modelling of Collaborative Learning Interactions, pages $23-30, 2000$. 28

[Ivanova and al.,2015] Ivanova, V., Lambrix, P., & Aberg, J. Requirements for and evaluation of user support for large-scale ontology alignment. Proceedings of the European Semantic Web Conference (pp. 3â20). Academic Press ; . doi :$10.1007/978-3-319-18818-8\_1$

[Jiang and al.,2006] Z. Jiang, S. Qingguo T. Tang, Li Y ongiang,-An aggregation cache replacement algorithm based on ontology clusteringâ. Journal of natural sciences. Vol. 11 NO.51141 $-$ 1146.2006 83

[Jiménez and al., 2008] Jiménez-Ruiz, E., Grau, B. C. , Sattler, U. , Schneider, T. et Berlanga, R. Safe and economie re-use of ontologies : A logic-based methodology and tool support. In The Semantic Web : Research and Applications, volume 5021 de LNCS $185-199$. Springer. 94

[Jiménez-Ruiz and Grau,2011] Jiménez-Ruiz, E., & Grau, B. C. Log map : Logic-based and scalable ontology matching. In The Semantic WebâISWC 2011. Springer.

[Jimenez-Ruiz and al., 2012] Jimenez-Ruiz, E., Grau, B., Zhou, Y., & Horrocks, I. Large-scale interactive ontology matching : Algorithms and implementation. In Frontiers in Artificial Intelligence and Applications (pp. 444449). IOS Press.

[Jimenez-Ruiz and al.,2018] imenez-Ruiz, E., Agibetov, A., Samwald, M., & Cross, V. We Divide, You Conquer : From Large-scale Ontology Alignment to Manageable Subtasks with a Lexical Index and Neural Embeddings. Proceedings of the 17th International Semantic Web Conference (ISWC 2018), Monterey, CA. Academic Press.

[Jouanot,2000] Fabrice JOUANOT. Un modèle sémantique pour l'interopérabilité de systèmes d'information. In INFORSID, pages 347364, 2000. 36, 41

[Jurish and Igler,2018] Jurisch, M., & Igler, B. RDF2Vec-based Classification of Ontology Alignment Changes. Proceedings of the First Workshop on Deep Learning for Knowledge Graphs and Semantic Technologies (DL4KGS) collocated with the 15*th* Extended Semantic Web Conference (ESWC 2018), Heraklion, Crete, Greece (pp. $40-45$). Academic Press.

[Kachroudi and al.,2013] Kachroudi, M., Hassen, W., Zghal, S., & Ben Yahia, S. Large Ontologies Partitioning for Alignment Techniques Scaling. In WEBIST $(pp.165-168)$. Academic Press.

[Kachroudi and al., 2017] Kachroudi, M., Diallo, G., & Yahia, S. B. OAEI 2017 results of KEPLER. Proceedings of the 12*th* International Workshop on Ontology Matching (pp. 138145). Academic Press. 126

[Kalfoglu and Shorlemmer,2003] Yannis KALFOGLOU et Marco SCHORLEMMER. If-map : an ontology mapping method based on information flow theory. Journal on Data Semantics, $1(1)$: 98127, october 2003. vii, 44, 46, 47

[Karol and Mangat, 2013] S. Karol, V. Mangat. âEvaluation of a Text Document Clustering Approach based on Particle Swarm Optimization| |. IJCSNS International Journal of Computer Science and Network Security, Vol.13 No.7, July 2013. 81

[Kengue and al.,2008] Jean-François Djoufak-Kengue, Jérôme Euzenat, Petko Valtchev et al. Alignement d'ontologies dirigé par la structure. Dans Actes 14*e* journées nationales sur langages et modèles à objets (LMO), pages 4357, 2008. 5

[Kolli, 2008] R. Kolli. -Scalable Matching Of Ontology Graphs Using Partitioningâ. M.S.Thesis, University of Georgia. Kunjir MP, 2008. 82

[Klein and Loebbecke,2000] KLEIN et LOEBBECKE. The transformation of pricing models on the web : examples from the airline industry,. In 13*th* International Bled Electronic Commerce Conference, pages 1921, June 2000. 24

[Konev and al.,2008] Konev, B., Lutz, C., Walther, D., & Wolter, F. Formal Properties of Modularization. In H. Stuckenschmidt & S. Spaccapietra (Eds.), Ontology Modularization. Springer.

[Kunger and Puraji, 2009] M.P. Kunjir, MD. Pujari, Project Report on Effective and Efficient computation of Cluster Similarity. M.S. Thesis, Indian Institute of Science, Bangalore 2009. 82

[Kuśnierczyk ,2008] Kuśnierczyk, W. Taxonomy-based partitioning of the Gene Ontology. Journal of Biomedical Informatics, $41(2), 282292$. doi :10.1016/j.jbi.2007.07.007

[Kutz and al., 2004] Kutz, O ., Lutz, C. , Wolter, F. et Zakharyaschev, M. E-connections of abstract description syst ems. Artificial Intelligence, $156(1)$,$1 - 73$.. 65

[Laadhar and al., 2017] Laadhar, A., Ghozzi, F., Megdiche, I., Ravat, F., Teste, O., & Gargouri, F. POMap : An Effective Pairwise Ontology Matching System. In KEOD $(pp.161 - 168)$. Academic Press. 126

[Lambrix and Kaliyaperumal,2013] Lambrix, P., & Kaliyaperumal, R. (2013). A session-based approach for aligning large ontologies. In 10th Extended Semantic Web Conference (pp. 4660). Academic Press ; . doi :$10.1007/978 - 3 - 642 - 38288 - 8_4$

[Luke and al.,1997] Sean LUKE, Lee SPECTOR, David RAGER et James HANDLER. Ontology-based web agents. In W. Lewis JOHNSON et Barbara HAYES-ROTH, réds., Proceedings of the First International Conference on Autonomous Agents (Agents'97), pages 5968, Marina delRey, CA, USA, 1997. ACM Press. 24

[Lutz and al., 2007] Lutz, C., Walther, D. et Walter, F. Conservative extensions in expressive description logics. Dans Proceedings of the 20*th* International Joint Conference on Artificial Intelligence, volume $7, 453 - 458$. , Hyderabad, India. 93

[Maedche and al.,2002] Alexander MAEDCHE, Boris MOTIK, Nuno SILVA et Raphael VOLZ. MAFRA - A Mapping Framework for Distributed Ontologies. In Proc. of 13*th* European Conference on Knowledge Engineering and Knowledge Management (EKAW), Siquenca, Spain, 2002. vii, 44, 45

[McGuinness and al.,2000] Deborah MCGUINNESS, Richard FIKES, James RICE et Steve WILDER. The chimaera ontology environment. pages 11231124. AAAI Press / The MIT Press.2000 51, 52

[Minsky,1975] Marvin MINSKY. A framework for representing knowledge. In P.M WINSTON, réd., ThePsychology of Computer Vision, pages 211277. McGraw Hill, New York, 1975. 19

[Mizoguchi and Bourdeau,2000] Riichiro MIZOGUCHI et Jacqueline BOURDEAU. Using ontological engineering to overcome common ai-ed problems. IJAIED, 2000. 29

[Miller and al.,1993] Miller, G., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. Introduction to WordNet : An on-line lexical database. MIT Press. 116

[Moawed and al.,2019] Moawed, S., Algergawy, A., Sarhan, A., & Eldosouky, A. A Framework for Efficient Matching of Large-Scale Metadata Models. Arabian Journal for Science and Engineering, $44(4), 31173135.doi$: $10.1007/s13369 - 018 - 3443 - 4$

[Namyoun and al.,2006] Choi NAMYOUN, Song IL-YEOL et Han HYOIL. A survey on ontology mapping. SIGMOD Rec., $35(3)$: 3441, September 2006. 50, 53

[Noy,2004] Natalya NOY. Semantic integration : a survey of ontology-based approaches. SIGMOD Rec., $33(4)$: $6570, 2004$ 43

[Noy and Musen,2000a] Natalya NOY et Mark MUSEN. Anchor-prompt : Using non-local context for semantic matching, 2000a. 53, 54

[Noy and Musen,2000b] Natalya Fridman NOY et Mark MUSEN. Prompt : Algorithm and tool for automated ontology merging and alignment. pages 450455, 2000b. 51

[Noy and Musen, 2003] Noy, N. F.et Musen, M. A. The prompt suite : interactive tools for ontology merging and mapping. International Journal of Human-Computer Studies, $59(6), 983 - 1024$. 91

[Noy and Musen, 2004] Noy, N. et Musen, M. Specifying ontology views by traversal. The Semantic Web-ISWC 2004, 713 − 725. 90, 91

[Oliveira and Pesquita, 2018] .Oliveira, D., Pesquita, C. Improving the interoperability of biomedical ontologies with compound alignments. Journal of biomedical semantics, 9(1), 1 : 1 − 1 : 13.

[Omelayenko,2003] Borys OMELAYENKO. Rdft : A mapping meta-ontology for web service integration. Pages 137153.2003. 44, 48

[Palmiz et al., 2009] Palmisano, I., Tamma, V., Payne, T. et Doran, P. Task oriented evaluation of module extraction techniques. In The Semantic Web-ISWC 2009130 − 145. Springer. 87

[Patrick, 2014] PATRICK,T.P. Modularisation des ontologies. Mémoire préparé comme exigence partielle de la maitrise en informatique. Université du Quebec à montréal. 2014 63, 65, 89, 92, 101

[Parent and Spaccapietra,2009] Parent, C. et Spaccapietra, S. An overview of modularity. Dans Modular Ontologies,volume 5445 de Lecture Notes in Computer Science,24 − 55.Springer Verlag. 84

[Pereira and al.,2017] Pereira, S., Cross, V., & Jimenez-Ruiz, E. On partitioning for ontology alignment. Proceedings of the International Semantic Web Conference (Posters & Demonstrations). Academic Press. 59

[Pesquita and al.,2014] Pesquita, C., Faria, D., Santos, E., Neefs, J.-M., & Couto, F. M. Towards visualizing the alignment of large biomedical ontologies. Proceedings of the 10*th* International Conference on Data Integration in the Life Sciences (*pp*.104111). Academic Press ; . doi :10.1007/978 − 3 − 319 − 08590 − 6_10

[Pim and al.,1997] Borst PIM, Akkermans HANS et Top JAN. Engineering ontologies. International Journal of Human-Computer Studies, 46(2/3) : 365406, 1997. 30

[Pinto and al., 1999] Pinto, H. S., Gòmez-Pèrez, A.,and Martins, J. P. Some issues on ontology integration.In Proceedings of the Workshop on Ontologies and Problem Solving Methods during IJCAI-99, Stockholm, Sweden. 86, 87

[PSYCHÉ and al.,2003] Valéry PSYCHÉ, Olavo MENDES et Jacqueline BOURDEAU. Apport de l'ingénierie ontologique aux environnements de formation à distance. Revue STICEF, 10, 2003. vii, 25, 32

[Pinto and al., 2000] H. Sofia Pinto, J.P. Martins « Reusing Ontologies », Portugal, 2000 86

[Qu and al.,2006] Qu Yuzhong, Hu Wei and Cheng Gong. Constructing Virtual Documents for Ontology Matching. In : Carr Les, David De Roure, Iyengar Arun, A. Goble Carole and Dahlin Michael. In : Proceedings of the 15*th* International Conference on World Wide Web, 2326 May, 2006, Edinburgh, Scotland. New York, NY : ACM Press, 2006, pp 23 − 31. 59

[Rahm and Bernstein, 2001] Erhard RAHM et Philip BERNSTEIN. A survey of approaches to automatic schema matching. The VLDB Journal, 10(4) : 334350, 2001. 53

[Rector and al.,2005] Rector, A. L., Napoli, A., Stamou, G., Stoilos, G., Wolger, H., Pan, J., & Tzouvaras, V. Report on modularization of ontologies [Technical report]. Knowledge Web.

[Roche,2005] Christophe ROCHE. Terminologie et ontologie. Revue Langages, numéro 157,Mars 2005. 12, 14, 37

[Rtk and al., 1995] P.J. Rousseeuw, E. Trauwaert, and L. Kaufman. Fuzzy clustering with high contrast. J. Comput. Appl. Math., 64 : 8190, November 1995. 81

[Ruder and al., 2018] Ruder, S., Vulić, I., SÙgaard, A. A survey of cross-lingual word embedding models. Journal of Artificial Intelligence Research. 108

[Salton and McGill, 1983] G.Salton et M. J.McGill, Introduction to modern information retrieval. McGraw-Hill. New York, 1983. 103

[Santos and al.,2015] Santos, E., Faria, D., Pesquita, C., & Couto, F. M. Ontology alignment repair through modularization and confidence-based heuristics. PLoS One, 10(12), e0144807. doi :10.1371/journal.pone.0144807 59

[Saruladha and al., 2012a] K. Saruladha, G. Aghila, B. Sathiya, -A Partitioning Algorithm for Large Scale Ontologies| |. International Conference on Recent Trends In Information Technology (ICRTIT), 2012 82

[Saruladha and al., 2012b] aruladha, K., Aghila, G., & Sathiy, B. LOMPT : An efficient and Scalable Ontology Matching Algorithm. Procedia Engineering, 38, 22722287. doi :10.1016/j.proeng.2012.06.274

[Schlicht and Stuckenschmidt ,2007] Schlicht, A., & Stuckenschmidt, H. Criteria-based partitioning of large ontologies. Proceedings of the 4th International Conference on Knowledge Capture (pp. 171172). Academic Press ; . doi :10.1145/1298406.1298439

[Shvaiko and Euzenat,2005] Shvaiko Pavel, Euzenat Jerome. A Survey of Schema-based Matching approaches. Journal on Data Semantics IV , 2005, vol.3730, pp 146 − 171

[Shank and Abelson,1988] SCHANK et ABELSON. Scripts, Plans, Goals and Understanding. Kaufmann, San Mateo,CA, 1988. 20

[Seidenberg and Rector, 2005] Seidenberg, J. et Rector, A. Techniques for segmenting large description logic ontologies. Dans Workshop on Ontology Management : Searching, Selection, Ranking, and Segmentation. 3rd International Conference on Knowledge Capture, 49 − 56. vii, 91, 92

[Seidenberg and Rector, 2006] Seidenberg, J. et Rector, A. Web ontology segmentation : analysis, classification and use. Dans Proceedings of the 15*th* International Conference on World Wide Web, 13 − 22., NY, USA. ACM. vii, 93

[Setti Ahmed and al., 2011] Soraya Setti Ahmed, Mimoun Malki, Sidi Mohamed Benslimane. Extracting Views From Domain Ontology : An Existential Dependency Driven Approach. International Conference on Knowledge Engineering and development, KEOD 2011. 26 − 28 Octobre, 2011, Paris â France. 95

[Setti Ahmed and Benslimane, 2012] Soraya Setti Ahmed, Sidi Mohamed Benslimane : Reverse Engineering Process for Extracting Views from Domain Ontology. International Conference of Web and Information Technologies (*ICWIT*'2012), 29 − 30 April, 2012, Sidi Bel Abbes, Algeria. 84

[Setti Ahmed and al., 15] Soraya Setti Ahmed, Mimoun Malki, Sidi Mohamed Benslimane. Ontology Partitioning : Clustering Based Approach. International Journal of Information Technology and Computer Science, 2015, 06, 1 − 11 Published Online May 2015 in MECS (http ://www.mecs-press.org/) DOI : 10.5815/ijitcs.2015.06.01

[Shabolt and al.,2006] Nigel Shabolt,Christopher Brewxter, Harith Alalani Proceeding ISWC'06 Proceedings of the 5th international conference on The Semantic Web Pages 1 − 15 Athens, GA â November 05 − 09, 2006 3

[Sowa,1992] John SOWA. Conceptual graphs summary. pages 351, 1992. 17

[Sowa,2000] John SOWA. Knowledge Representation : Logical, Philosophical, and Computational Foundations. Brooks/Cole, August 2000. vii, 14, 18, 26

[Stoilos and al.,2018] Stoilos, G., Geleta, D., Shamdasani, J., & Khodadadi, M. A novel approach and practical algorithms for ontology integration. In Proceedings of ISWC. Academic Press ; . doi :10.1007/978 − 3 − 030 − 00671 − 6_27 114

[Stuckenschmidt and Klein, 2004] Heiner Stuckenschmidt and Michel C. A. Klein 2004. Structure-Based Partitioning of Large Concept Hierarchies. In Sheila A. McIlraith, Dimitris Plexousakis, and Frank van Harmelen (eds), International Semantic Web Conference, volume 3298 of Lecture Notes in Computer Science, pages 289_303. Springer. 61, 62, 63, 89

[Stumme and Maedche,2001] Gerd STUMME et Alexander MAEDCHE. Fca-merge : Bottom-up merging of ontologies. In IJCAI, pages 225234, 2001. 51, 52

[Stuckenschmidt and Schlicht , 2009] Stuckenschmidt, H. et Schlicht, A. Structure-based partitioning of large ontologies. Dans Modular Ontologies, volume 5445 de Lecture Notes in Computer Science, 187 − 210. Springer-Verlag. vii, 63

[Thanh LE and al.,2004] Bach Thanh LE, Rose DIENG-KUNTZ et Fabien GANDON. On ontology matching problems- for building a corporate semantic web in a multi-communities organization. ICEIS (4), pages 236243, 2004. 54, 56

[Ushold and Grüninger,1996] Mike USCHOLD et Michael GRÜNINGER. Ontologies : principles, methods, and applications. Knowledge Engineering Review,11(2) : 93155, 1996. 14, 15, 20, 31, 33

[Vernadat,2007] FranÃ§ois VERNADAT. Interoperable enterprise systems : Architectures, methods and metrics. Rapport technique, LGIPM, Université de Metz, France, 2007. 36, 41

[Wang and al.,2006] Wang Zongjiang, Wang Yinglin, Zhang Shensheng, Shen Ge and Du Tao. Effective Large Scale Ontology Mapping. In : Lang Jérôme, Lin Fangzhen and Wang Ju. Proceedings of the First International Conference Knowledge Science, Engineering and Management, 5 − 8 August, 2006, Guilin, China. Springer, 2006, pp 454 − 465. 59, 84

[Wang and al.,2018] Wang, L., Bhagavatula, C., Neumann, M., Lo, K., Wilhelm, C., & Ammar, W. Ontology Alignment in the Biomedical Domain Using Entity Definitions and Context. Proceedings of the BioNLP workshop, Melbourne, Australia (pp. 4755). Academic Press.

[Wang and al.,2011] Wang, P., Zhou, Y., & Xu, B. Matching large ontologies based on reduction anchors. Proceedings of the 22*nd* International Joint Conference on Artificial Intelligence (pp. 2243 − 2348). Academic Press.

[Webster,2004] Merriam-Webster dictionnary. 2004. 2, 13

[Winkler, 1999] William E. Winkler 1999. The state of record linkage and current research problems. Technical Report *RR*/1999/04, Statistics Research Division, U.S. Bureau of the Census. 68

[Wu and Palmer., 1994] Z. Wu et M. Palmer. Verb semantics and lexical selection. In Proceedings of the 32*nd* Annual Meeting of the Associations for Computational Linguistics, *pp*133 − 138.1994. 68

[Zerhouni and Benslimane,2019a] ONTEM : Extraction based Alignment Method for Large Ontologies April 2019. Conference : Third Edition of the National Study Day on Research on Computer Sciences : JERI'2019At : Saida, Algeria. 106

[Zerhouni and Benslimane,2019b] Large-Scale Ontology Alignment- An Extraction Based Method to Support Information System Interoperability. International Journal of Strategic Information Technology and Applications Volume 10 Issue 2 April-June 2019 84 – 59. 107, 126, 137

[Zhao and Zhang, 2016] Zhao, M., & Zhang, S. Identifying and validating ontology mappings by formal concept analysis. In ISWC Conference on Ontology Matching (OM) (pp. 6172). Academic Press ; Retrieved from www.CEUR-WS.org 114

RÉSUMÉ

La compétitivité des entreprises est profondément liée à la capacité de partager et échanger les connaissances et le savoir-faire avec l'ensemble de ses collaborateurs. Ce besoin d'échanger des connaissances oblige les entreprises d'évoluer leurs systèmes d'informations hétérogènes afin de les rendre interopérables. Pour faire face à l'hétérogénéité sémantique, les approches basées sur les ontologies sont largement utilisées en raison de l'évolution rapide des technologies connexes du web sémantique et les avantages qu'elles apportent pour faciliter l'interopérabilité sémantique. L'alignement ontologique est un moyen important d'établir l'interopérabilité entre les applications Web sémantiques qui utilisent des ontologies différentes mais connexes. De nos jours, le partitionnement et la modularisation sont les deux stratégies principales pour décomposer les grandes ontologies en blocs ou en modules d'ontologies afin de les aligner. Si pour les ontologies de petite taille le problème d'alignement ne se pose pas, les méthodes d'alignement des ontologies larges restent toujours posées. Cette thèse a comme objectif de proposer une nouvelle stratégie d'alignement basée sur l'extraction des entités ontologiques pour résoudre les différents problèmes techniques tels l'optimisation de l'espace mémoire et du temps d'exécution de l'opération d'alignement des ontologies larges. Notre stratégie consiste à reformuler le problème d'alignement entre deux ontologies comme étant un problème d'optimisation et, par conséquent, de concevoir une méthode efficace pour le résoudre. Ainsi, nous proposons ONTEM, une méthode d'alignement basée sur les opérations algébriques permettant d'obtenir automatiquement des alignements pour les ontologies larges sans aucun calcul de distance de similarité. Le benchmark OAEI LargeBioTrack 2018 a servi comme support de comparaison de notre prototype avec l'ensemble des concurrents présents dans cette édition. Les résultats expérimentaux obtenus sont prometteurs et ont révélé qu'ONTEM a manifesté globalement une bonne performance en termes de précision, de Rappel, de F-mesure et de temps d'exécution et d'occupation mémoire(aucun dépassement de capacité).
Mots-clés : Alignement, extraction, ontologies larges, modularisation, partitionnement, interopérabilité sémantique.

ملخص

ترتبط القدرة التنافسية للّشركات ارتباطا عميقا بالقدرة على مشاركة و تبادل المعرفة و الدراية الفنية مع جميع المتعاونين. هذه الحاجة الى تبادل المعرفة تجبر الشركات على تطوير أنظمة المعلومات غير المتجانسة لجعلها قابلة للّتشغيل المتبادل. للّتعامل مع عدم التجانس الدلالي، يتم استخدام الاساليب القايئمة على الانطلوجيا على نطاق واسع بسبب التطور السريع لتقنيات الويب الدلالية ذات الصلة و الفوايئد التي تجلبها لتسهيل التشغيل البيني الدلالي. المحاذات الأنطولوجية هي وسيلة مهمة لاثبات امكانية التشغيل البيني بين تطبيقات الويب الدلالية التي تستخدم الأنطولوجيا المختلفة و ذات الصلة. في الوقات الحاضر، يعتبر التقسيم و النمذجة الاستراتجيتين الريئيسيتين لتقسيم الأنطولوجيا الضخمة الى كتل أو وحدات الأنطولوجيا لمحاذاتها. لا يوجد أي مشكل في عملية محاذات أو انحياز أنطولوجيا ذات الحجم الصغير بينما هناك مشاكل عند عملية انحياز أنطولوجيا ذات الحجم الكبير.

تهدف هذه الرسالة الى اقتراح استراتيجية محاذاة جديدة تستند الى استخراج العناصر الأنطولوجية لحل المشكلات الفنية المختلفة مثل تحسين سعة الذاكرة و وقت تنفيذ عملية المحاذاة. بالنسبة للّأنطولوجيا الواسعة الحجم.
تّتمثل استراتجيتنا في اعادة صياغة مشكلة التوافق بين اثنين من الأنطولوجيا كمشكلة تحسين، و بالتالي، لتصميم طريقة فعالة لحلها. انّنا نقترح *ONTEM* ، و هي طريقة محاذاة تستند الى العمليات الجبرية التي تسمح بالحصول على محاذاة تلقايئية لأنطولوجيا واسعة الحجم دون أي حساب لمسافة التشابه.
كان معيار *OAEILargeBioTrack*2018 بمثابة دعم مقارنة لنموذجنا الأولي مع جميع المنافسين الموجودين في هذه الطبعة. النتايئج التجريبية التي تم الحصول عليحا واعدة و كشفت أن *ONTEM* قد أظهرت أداءجيدا على مستوى العام من حيث الدقة و الاسترجاع و وقت القياس و التنفيذ و شغل الذاكرة (لا يوجد تجاوز سعة الذاكرة).

كلمات المفاتيح : المحاذاة، استخراج، التقسيم ، النمذجة، الأنطولوجيا الواسعة، التجانس الدلالي.

Abstract
The competitiveness of business enterprises is deeply linked to the ability to share and exchange knowledge and know-how with all of its collaboraters. This need to exchange knowledge forces business enterprises to evolve their heterogeneous information systems to make them interoperable. To deal with semantic heterogeneity, ontology-based approaches are widely used because of the rapid evolution of related semantic web technologies and the benefits they bring to facilitate semantic interoperability. Ontological alignment is an important means of establishing interoperability between semantic Web applications that use different but related ontologies. Nowadays, partitioning and modularization are the two main strategies for breaking down large ontologies into blocks or ontology modules in order to align them. If for small ontologies the problem of alignment does not arise, the methods of alignment of large ontologies remain always posed.
This thesis aims at proposing a new alignment strategy based on the extraction of ontological entities to solve the different technical problems such as optimization of the memory space and the execution time of the alignment operation of large ontologies. Our strategy is to reformulate the alignment problem between two ontologies as an optimization problem and, therefore, to design an effective method to solve it. Thus, we propose ONTEM, an alignment method based on the algebraic operations allowing to obtain automatically alignments for the large ontologies without any calculation of distance of similarity. The OAEI LargeBioTrack 2018 benchmark served as a comparison support for our prototype with all the competitors present in this edition. The experimental results obtained are promising and have revealed that ONTEM has globally exhibited a good performance in terms of accuracy, recall, F-measurement and execution time and memory occupancy (no memory overflow).
Keywords : Alignment, extraction, large ontologies, modularization, partitioning, semantic interoperability.