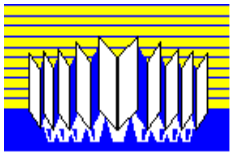
UNIVERSITE DE TUNIS EL MANAR — FACULTÉ DES SCIENCES DE TUNIS

UNIVERSITE D'ARTOIS — CRIL-CNRS, LENS



MÉMOIRE DE THÈSE

présenté en vue de l'obtention du

Diplôme de Doctorat en Informatique

Sami ZGHAL

DEA en Modélisation et Informatique de Gestion

I.S.G. Tunis - Tunisie

CONTRIBUTIONS À L'ALIGNEMENT D'ONTOLOGIES OWL PAR AGRÉGATION DE SIMILARITÉS

Soutenu le 21 décembre 2010 devant le jury composé de :

Président

Mohamed Mohsen GAMMOUDI, MCF-HDR, Université du 7 novembre à Carthage - Tunisie

Rapporteurs

Sylvie DESPRES, Professeur, Université de Paris 13 - France

Faiez GARGOURI, Professeur, Université de Sfax - Tunisie

Examinateur

Gilles GONCALVES, Professeur, Université d'Artois - France

Co-Directeurs

Sadok BEN YAHIA, MCF-HDR, Université de Tunis El Manar -Tunisie

Engelbert MEPHU NGUIFO, Professeur, Université Blaise Pascal - France

2010-2011

À mes parents

À ma femme

À ma sœur et mon frère

À toute ma famille

Remerciements

Je tiens à remercier Sadok Ben Yahia et Engelbert Mephu Nguifo qui ont dirigé mes travaux de thèse. Je tiens à leurs exprimer ma gratitude pour la qualité de leurs conseils, leurs disponibilités ainsi que le degré de responsabilisation de leur encadrement qui m'a permis de développer mon goût pour la recherche.

Je tiens à remercier Monsieur Mohamed Mohsen Gammoudi pour avoir accepter de présider le jury et d'évaluer mon travail.

Je remercie mes deux rapporteurs de thèse Madame Sylvie Despres et Monsieur Faiez Gargouri pour l'intérêt qu'ils ont porté à ce travail en acceptant de le juger et pour le temps qu'ils ont consacré pour la lecture de cette thèse.

Je tiens à remercier également Monsieur Gilles Goncalves de m'avoir honoré en acceptant d'examiner ma thèse.

Je remercie Monsieur Khaled Besaies de m'avoir accueilli dans son unité de recherche URPAH (Unité de Recherche en Programmation, Algorithmique et Heuristique) ainsi que tous les membres de l'unité et en particulier Tarek Hamrouni pour son assistance et aide fructueuse.

Je souhaite également remercier tous les membres du CRIL (Centre de Recherche en Informatique de Lens), doctorants et permanents, de m'avoir accepté au sein du centre et soutenu mes travaux de recherche.

Table des matières

Résumé

Les ontologies sont de plus en plus utilisées dans plusieurs domaines et en particulier dans la représentation des connaissances et le Web sémantique. Dans ces deux domaines, une multitude d'ontologies ont été développées pour la modélisation des connaissances. Cependant, ces ontologies peuvent comporter plusieurs formes d'hétérogénéité à savoir : l'hétérogénéité syntaxique, l'hétérogénéité terminologique, l'hétérogénéité conceptuelle et l'hétérogénéité sémiotique. Dans ce cadre, l'alignement s'impose comme une solution *sine qua non* aux problèmes d'hétérogénéité. Dans la littérature, il existe plusieurs travaux (par exemple l'alignement, la fusion, l'intégration, la production de ponts d'axiomes, la traduction, etc.) qui visent à réduire cette hétérogénéité selon ses aspects terminologiques et/ou conceptuels. Les méthodes actuelles d'alignement d'ontologies reposent sur l'utilisation d'un seul type de mesure de similarité et l'exploitation restreinte des structures internes des ontologies. Les méthodes d'alignement ne sont pas complètement satisfaisantes et le problème d'alignement reste ouvert pour la proposition de nouvelles contributions.

Dans le cadre de cette thèse, nous avons proposé plusieurs méthodes d'alignement à savoir : la méthode EDOLA (Extended Diameter OWL-Lite Alignment), la méthode SODA (Structural Ontology OWL-DL Alignment) et la méthode OACAS (Ontologies Alignment using Composition and Aggregation of Similarities). Ces méthodes reposent sur l'agrégation et la composition de similarités, ainsi que l'exploration étendue de la structure des ontologies à aligner. La méthode EDOLA permet d'aligner deux ontologies OWL-Lite, tandis que les deux autres méthodes, SODA et OACAS, considèrent des ontologies OWL-DL. Les trois méthodes procèdent dans une première phase à la transformation des deux ontologies à aligner sous la forme d'un graphe, nommé O-Graph, pour chaque ontologie. Ces graphes permettent la représentation fidèle des ontologies OWL sous une forme facile à l'exploitation lors du processus d'alignement. Ainsi, les graphes obtenus reflètent toutes les informations des ontologies à savoir les entités, les relations entre ces dernières et les instances (individus).

La méthode EDOLA est une approche se basant sur un modèle de calcul des similarités locale et globale. Ce modèle suit la structure du graphe O-Graph pour calculer les mesures de similarité entre les nœuds des deux ontologies. Le module d'alignement associe pour chaque catégorie de nœuds une fonction d'agrégation. Cette dernière prend en considération toutes les mesures de similarités entre les couples de nœuds voisins au couple de nœuds à apparier. Ainsi, cette fonction exploite toute l'information descriptive de ce couple.

La méthode SODA est une amélioration de la méthode EDOLA. En effet, la méthode SODA opère sur des ontologies OWL-DL, pour les aligner, au lieu d'ontologies décrites en OWL-Lite. La méthode SODA est une approche structurelle pour l'alignement d'ontologies OWL-DL. Elle opère en trois étapes successives et exploite la structure des ontologies à travers leurs graphes O-Graphs. La première étape permet de calculer la similarité linguistique à travers des mesures de similarité plus adaptées aux descripteurs des constituants des ontologies à apparier. La seconde étape détermine la similarité structurelle en exploitant la structure des deux graphes O-Graphs. La troisième étape déduit la similarité sémantique, en prenant en considération les deux types de similarités déjà calculées, afin de fournir l'alignement résultat.

La méthode d'alignement, OACAS, opère en trois étapes successives pour produire l'alignement. La première étape permet de calculer la similarité linguistique composée. Cette dernière prend en considération tous les descripteurs des entités ontologiques à aligner. La seconde étape détermine la similarité de voisinage par niveau. Le processus de calcul de la similarité de voisinage opère sur deux niveaux successifs. La troisième étape du processus d'alignement agrège les composants de la similarité linguistique composée et la similarité de voisinage par niveau pour déterminer la similarité agrégée. La similarité agrégée représente le résultat de la méthode d'alignement OACAS.

Les trois méthodes sont évaluées en utilisant les bases d'évaluation proposées par les différentes campagnes de l'OAEI (Ontology Alignment Evaluation Initiative). Les résultats obtenus par les différentes méthodes proposées sont présentés et discutés par rapport aux méthodes pionnières de la littérature. Nous discutons aussi l'apport de chacune des méthodes proposées.

Abstract

Ontologies are the kernel of semantic Web. They allow the explicitation of the semantic purpose for structuring different fields of interest. In order to harmonize them and to guarantee the interoperability between these resources, the topic of alignment of ontologies has emerged as an important process to reduce their heterogeneity and improve their exploitation. Between different ontologies of the same field may exist some heterogeneity : syntactic heterogeneity, terminological heterogeneity, conceptual heterogeneity and semiotic heterogeneity. Ontology alignment can be the solution for reducing heterogeneity problem. Many ontology alignment methods use only one similarity measure and handle a restricted internal structure of the ontologies to be aligned.

In this thesis, we have proposed three ontology alignment methods : EDOLA (Extended Diameter OWL-Lite Alignment) method, SODA (Structural Ontology OWL-DL Alignment) method and OACAS (Ontologies Alignment using Composition and Aggregation of Similarities) method. These methods rely on aggregation and composition of similarities and check the spread structure of the ontologies to be aligned. EDOLA method allows to align OWL-Lite ontologies whereas SODA and OACAS consider OWL-DL ontologies. The three proposed methods operate in a first step by transforming both ontologies to aligned as a graph, named O-Graph, for each ontology. This graph reproduces OWL ontologies to be easily manipulated during the alignment process. The obtained graphs describe all the information contained in the ontologies : entities, relations between entities and instances.

Besides, the EDOLA method is a new approach that computes local and global similarities using a propagation technique of similarities through the O-Graphs. This model explores the structure of the considered O-Graphs to compute the similarity values between the nodes of both ontologies. The alignment model associates for each category of nodes an aggregation function. This function takes in consideration all the similarity measures of the couple of nodes to match. This aggregation function explores all descriptive

information of this couple. EDOLA operates in two successive steps. The first step computes the local similarity (*i.e.*, terminological one), whereas the second step computes the global one.

The SODA method is an improved version of EDOLA. In fact, it uses OWL-DL ontologies. SODA method is a structures approach for OWL-DL ontologies. The method operates in three successive steps and explores the structure the ontologies using O-GRAPHS. The first step computes linguistic similarity using appropriate similarity measures corresponding to the descriptors of ontological entities. The second step allows to compute structural similarity using the two graphs O-GRAPHS. The third step deduces the semantic similarity, by combining both similarities already computed, in order to outperform the alignment task.

The OACAS method introduces a new alignment algorithm handling OWL-DL ontologies. The main thrust of this method is the application of the most suitable similarity measure depending of the category of the node in the ontology. In addition, the OACAS method explores a wider neighborhood than do the pioneering methods of the literature. OACAS method lays on a composition of similarity computation based model. The method starts by exploring the O-GRAPH structure. An aggregation function combines the similarity measures and the node's structures of the nodes to be aligned. Thus, this function considers all the descriptive information of this couple (name, comment and label) as well as its neighborhood structure. The OACAS method operates into three successive steps. The first one computes the composed linguistic similarity, the second one computes neighborhood similarity by level whereas the third one computes the aggregation similarity.

The three methods have been evaluated using Benchmark bases proposed by the OAEI (Ontology Alignment Evaluation Initiative) campaign. The obtained results of different methods proposed are presented and compared with others pionniers method in the literature.

Liste des figures

Liste des tableaux

Liste des algorithmes

Introduction générale

Actuellement, plus de 4.2 milliards de pages sont recensés sur le réseau Internet. Ces pages sont majoritairement présentées dans un format inexploitable automatiquement par les machines. En revanche, elles sont lisibles et compréhensibles uniquement par l'homme. Par conséquent, les machines, ou plutôt les agents logiciels, ne peuvent effectuer que des recherches purement syntaxiques dans le contenu de ces documents. Ils sont incapables d'interpréter et d'analyser les informations contenues dans ces pages Web.

Afin de remédier aux insuffisances du Web actuel, Berners-Lee *et al.* (2001) ont proposé une nouvelle vision du Web, appelée *le Web sémantique*. Le principal pilier du Web sémantique est l'utilisation des ontologies. Elles permettent de décrire la structure et la sémantique des données contenues dans un document. Les ontologies assurent l'organisation des informations sous la forme d'une taxonomie de concepts et des relations entre ces derniers (Chandrasekaran *et al.* (1999)). Dans le cas où les documents sont décrits ou annotés par l'intermédiaire d'ontologies, les machines et en particulier les agents logiciels peuvent interpréter leurs sémantiques. Cette interprétation localise et intègre les données pour diverses applications.

Le Web sémantique permet l'uniformisation et la standardisation des objectifs. Il assure la normalisation de la description sémantique des informations à travers l'utilisation du formalisme XML (eXtensible Markup Language). Il organise les connaissances sous la forme d'ontologies permettant la description d'un domaine d'étude ainsi que ses aspects sémantiques (Gruber (1995)).

Dans cette perspective, il s'avère nécessaire de recourir à l'accès aux informations ou aux ressources distribuées, et de gérer les entités distinctes et hétérogènes. Il est très difficile de définir un consensus autour des ontologies. Les difficultés proviennent des formalismes utilisés pour la représentation des ontologies. L'exploitation de ces derniers engendre des problèmes au niveau de l'interopérabilité des applications.

L'interopérabilité des ontologies nécessite la comparaison des ces dernières. Les tech-

niques d'alignement d'ontologies assurent un moyen de reconciliation et de rapprochement entre les différentes sources d'informations (Euzenat et Shvaiko (2007)). Le Web comporte plusieurs ressources décentralisées. Ces dernières sont représentées sous la forme d'ontologies. Elles peuvent décrire des domaines semblables, dites *ontologies de domaine*, et utilisent des terminologies différentes. L'intégration des ontologies distinctes, d'un même domaine, prend en considération les correspondances sémantiques entre les éléments constituant ces ontologies.

Les techniques d'alignement doivent être robustes et évolutives. La robustesse des résultats assure que des erreurs mineures au niveau des résultats obtenus ne doivent pas avoir des conséquences importantes. Les techniques d'alignement doivent évoluer, car elles doivent produire des résultats dans un temps raisonnable avec des ontologies, qui peuvent contenir des centaines de concepts sémantiques.

Afin de réaliser l'alignement d'ontologies, il est nécessaire d'impliquer aussi bien les aspects syntaxiques que les aspects sémantiques des ontologies. Le processus d'alignement est un des éléments fondamentaux du processus d'intégration d'ontologies. Il permet, d'une part, d'analyser le contenu des ontologies à aligner, d'autre part, de comparer des ontologies pour déterminer les correspondances entre leurs constituants. Il permet aussi la détection d'éventuels conflits existants. Le résultat obtenu du processus d'alignement est un ensemble de liens de correspondances entre les entités ontologiques détectées lors du processus d'alignement.

Les correspondances obtenues, lors processus d'alignement, permettent d'assurer l'interopérabilité entre les applications exploitant plusieurs sources d'ontologies hétérogènes. L'alignement d'ontologies est encore, en grande partie, réalisé manuellement par les experts des domaines. Par conséquent, l'alignement d'ontologies, prenant en considération les aspects sémantiques, est désormais devenu une tâche cruciale dans le déploiement d'une large variété d'applications de gestion et de manipulation de l'information.

Problématique

Depuis le début des années 2000, l'alignement d'ontologies a pris de l'ampleur dans le domaine de l'intelligence artificielle et en particulier dans l'ingénierie des connaissances. La littérature recense une grande panoplie de méthodes (Euzenat et Shvaiko (2007)). Ces dernières sont issues de diverses communautés telles que : la recherche d'information, les bases de données, l'apprentissage, l'ingénierie des connaissances, le traitement automa-

tique du langage, etc (Euzenat et Shvaiko (2007)). La grande majorité de ces méthodes d'alignement repose sur l'exploitation des mesures de similarité. Par ailleurs, d'autres méthodes, exploitent des ressources complémentaires dans leur processus d'alignement. De ce fait, plusieurs limites majeures se présentent, à savoir :

- La sémantique des alignements produits est limitée à la prise en charge des relations d'équivalence. En effet, les mesures de similarités, de part leur nature symétrique, ne peuvent être utilisées que pour détecter des relations d'implication (ou de subsomption) ;
- Les méthodes d'alignement reposent sur un processus d'appariement, qui n'exploite pas la totalité des informations des entités contenues dans les ontologies. Ces descripteurs offrent des degrés d'expressivité pouvant influencer grandement la qualité de l'alignement produit ;
- L'utilisation d'une seule mesure de similarité pour tous les différents types de descripteurs des entités peut produire des valeurs de similarité faibles. En effet, il est nécessaire d'adapter la mesure de similarité au type du descripteur considéré ;
- Une grande partie des méthodes d'alignement ne prend pas en considération l'aspect structurel des ontologies à aligner. Ces méthodes n'exploitent pas (ou exploitent d'une manière restreinte) la structure des entités ontologiques à aligner ;

Objectifs

L'objectif du présent travail est de proposer quelques méthodes capables d'aligner des ontologies. Les méthodes d'alignement permettent d'identifier les correspondances syntaxiques, sémantiques et même structurelles entre les constituants des ontologies. Ces correspondances soutiennent l'interopérabilité des systèmes d'information. Les objectifs de cette thèse sont énumérés dans ce qui suit :

- **Objectif 1 :** nous cherchons à résoudre le problème d'hétérogénéité relatif au langage de représentation d'ontologies. Nous choisirons d'utiliser OWL comme langage de représentation ;
- **Objectif 2 :** les ontologies à aligner sont rarement identiques et ne sont pas complètes. D'où, la nécessité de mettre à la disposition des utilisateurs des méthodes d'alignement capables d'identifier les correspondances de tous les (ou une partie des) éléments des ontologies à aligner. En d'autres termes, les méthodes doivent être capables de déterminer un nombre maximal de correspondances entre les différentes

entités contenues dans les ontologies à aligner ;

- **Objectif 3 :** la littérature comporte un nombre considérable de méthodes d'alignement d'ontologies. Certaines méthodes reposent sur des modèles de calcul de similarité entre les constituants des ontologies. D'autres méthodes, proposent d'exploiter les structures internes des ontologies à aligner. Les méthodes existantes parviennent dans certains cas à atteindre un nombre considérable de correspondances correctes. Dans d'autres cas, les résultats obtenus sont médiocres. De plus, ces valeurs de similarité n'exploitent pas la structure et la richesse sémantique des ontologies à aligner. Dans ce cadre, il y a donc un effort à produire afin d'augmenter la fiabilité de l'alignement obtenu indépendamment de la conceptualisation des ontologies. Les méthodes proposées analysent et exploitent les ressources des ontologies afin de prendre en charge la sémantique et améliorer la qualité des résultats obtenus ;
- **Objectif 4 :** les méthodes proposées exploitent, d'une part, des mesures de similarité adaptées aux différents descripteurs des entités contenues dans les ontologies. D'autre part, elles exploitent la sémantique incorporée dans la structure des ontologies. Les méthodes explorent le voisinage des entités à aligner avec des degrés différents en propageant les similarités ;
- **Objectif 5 :** la phase de validation permet de comparer les résultats obtenus par les méthodes proposées avec les résultats d'autres méthodes pionnières de la littérature. Ce processus de validation repose sur l'utilisation d'un ensemble de métriques d'évaluation. Ces métriques permettent d'une part d'apprécier la qualité de l'alignement obtenu ; et d'autre part, elles assurent une démarche de comparaison avec les résultats obtenus par d'autres méthodes d'alignement.

Principales contributions

En prenant en considération les objectifs fixés, les méthodes proposées se composent d'une phase d'analyse, d'une phase de développement et d'une phase de test et d'évaluation. L'ensemble de ces phases inclut les étapes suivantes :

1. Une étude du domaine de l'alignement d'ontologies. Pour ce faire, une recherche est nécessaire sur les méthodes ainsi que sur les solutions existantes dans le domaine de l'alignement des ontologies OWL ;
2. L'étude des mesures de similarité existantes dans la littérature sont applicables pour la résolution de l'hétérogénéité terminologique. Ces mesures exploitent soit des in-

formations auxiliaires, soit des fonctions mathématiques. Les informations auxiliaires sont capables d'identifier les liens linguistiques entre les termes. Les fonctions mathématiques permettent le calcul de la distance syntaxique entre les termes ;

3. Le développement des méthodes d'alignement inclut la description détaillée des mesures de similarité basées sur : la comparaison terminologique ou lexicale ainsi que la structure des concepts ontologiques ;
4. La phase d'évaluation comporte l'expérimentation et l'évaluation des différentes mesures de similarité exploitées par les différentes méthodes d'alignement développées.

En prenant en considération les étapes précédentes et des objectifs cités auparavant, les principales contributions de la thèse peuvent être récapitulées comme suit :

1. **Présentation des ontologies** : les ontologies OWL à aligner sont transformées dans une première étape sous la forme de graphes. Cette représentation permet de décrire toutes les informations contenues dans une ontologie OWL . Les graphes obtenus sont exploités dans une seconde étape par le processus d'alignement ;
2. **Mesures de similarité** : les méthodes d'alignement reposent sur l'utilisation de plusieurs mesures de similarité. À cet effet, une panoplie de mesures est implémentée et testée pour dégager les plus pertinentes en fonction des informations décrivant les constituants des ontologies ;
3. **Méthodes d'alignement** : les méthodes d'alignement proposées, dans le cadre de cette thèse, se basent sur le principe d'agrégation et composition de mesures de similarité. Dans une première phase, les mesures de similarité (linguistique, syntaxique et terminologique) sont déterminées. La seconde phase permet de prendre en charge les structures des ontologies à aligner à travers la notion de voisinage pour incorporer la sémantique ;
4. **Automatisation du processus d'alignement** : la plupart des méthodes d'alignement d'ontologies nécessitent l'intervention de l'utilisateur. En effet, des méthodes automatisent une partie du processus d'alignement. Les méthodes proposées sont des approches ne nécessitant pas l'intervention des utilisateurs ;
5. **Expérimentation** : l'utilisation des métriques d'évaluation d'alignement permet d'estimer la qualité des alignement obtenus. Les expérimentations sont réalisées sur les bases de test *Benchmark* mises à la disposition de la communauté de l'alignement d'ontologies.

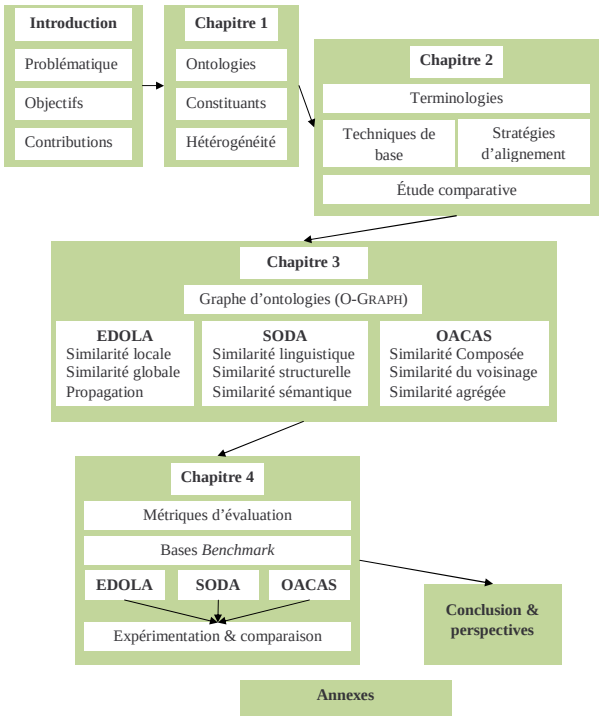


Figure 1 — Structure du manuscrit

Structure du document

Cette thèse est organisée en quatre chapitres. La Figure 1 présente cette organisation. Dans le premier chapitre, les concepts de base des ontologies sont présentés. Il propose une classification ainsi qu'une présentation des constituants d'une ontologie. Ce chapitre introduit ensuite les principaux langages dédiés pour la description des ontologies et propose une étude comparative de ces langages.

Le chapitre 2 introduit les concepts de base de l'alignement d'ontologies. Il présente les techniques de base et les stratégies d'alignement. Ce chapitre propose une étude comparative des principales méthodes d'alignement en se basant sur les caractéristiques internes et externes ainsi que sur les stratégies employées lors du processus d'alignement.

Le chapitre 3 est dédié à la présentation des contributions proposées dans le cadre de l'alignement d'ontologies OWL par agrégation de similarités. Il développe, dans une première étape, une description du processus de construction des graphes ontologies O-GRAPH. Dans une seconde étape, le chapitre présente en détail les trois méthodes d'alignement d'ontologies EDOLA (Extended Diameter OWL-Lite Alignment, Zghal *et al.* (2007e)), SODA (Structural Ontology OWL-DL Alignment, Zghal *et al.* (2007b)) et OA-

CAS (Ontologies Alignment using Composition and Aggregation of Similarities, Zghal *et al.* (2009)) en mettant en exergue les apports de chaque méthode.

Dans le chapitre 4, l'évaluation et l'expérimentation des trois méthodes proposées sont présentées. Cette expérimentation est menée en exploitant des bases de test *Benchmark*. L'expérimentation est effectuée par l'intermédiaire d'un ensemble de métriques d'évaluation, qui permettent d'apprécier la qualité des alignements obtenus. Le chapitre développe aussi une étude comparative des méthodes proposées par rapport aux méthodes pionnières appartenant à la littérature.

En conclusion, un bilan du travail est dressé. La conclusion générale contient aussi une analyse des principales contributions. Elle contient également des perspectives et des ébauches des travaux futurs.

CHAPITRE 1

Fondements des ontologies

1.1 Introduction

Les ontologies sont aujourd'hui au cœur du Web sémantique. Les ontologies proposent des représentations sémantiques des connaissances, susceptibles d'être manipulées par les machines. Elles participent dans les dimensions scientifiques et techniques du domaine de l'Intelligence Artificielle et en particulier dans la branche de l'ingénierie des connaissances (Bachimont (2000)).

Le champ d'application des ontologies ne cesse de s'étendre et couvre les systèmes conseillers (systèmes d'aide à la décision, systèmes d'enseignement assisté par ordinateur e-learning, etc.), les systèmes de résolution de problèmes et les systèmes de gestion et la manipulation des connaissances (dans plusieurs domaines : agricole, médical et biologique). Le principal objectif des ontologies est de doter le Web d'une couche sémantique assurant la recherche d'informations aussi bien sur le niveau syntaxique qu'au niveau sémantique. L'objectif est de créer un Web intelligent permettant de rendre compte de la sémantique à travers la prise en charge du sens des informations par l'intermédiaire des ontologies.

Le reste du chapitre est organisé somme suit : la section 1.2 rappelle les principales définitions du terme "ontologie" dans plusieurs domaines. Dans la section 1.3, une description détaillée des constituants est dressée. Dans la section 1.4, plusieurs classifications d'ontologies sont proposées selon plusieurs niveaux (conceptualisation, granularité, complétude, etc.). Les langages permettant de représenter les ontologies ainsi qu'une étude comparative sont récapitulés dans la section 1.5. L'étude des différents types d'hétérogénéité fait l'objet de la section 1.6.

1.2 Ontologies

Le terme ontologie est apparu au 17 ème siècle. Il a été introduit séparément par les deux philosophes Lorhard (1606) et Goclenius (1613). Le mot ontologie est difficile à définir d'une manière précise et claire. En effet, ce mot est exploité dans différents contextes appartenant à plusieurs domaines tels que la philosophie, la linguistique, l'intelligence artificielle (en particulier l'ingénierie des connaissances) et le Web sémantique.

1.2.1 Étude de ce qui est

Le mot ontologie est la réunion de deux termes *"onto"* et *"logie"*. Le premier terme *"onto"* (ou *"ontif"*) signifie *"je suis"*. Le second terme *"logie"* (ou *"logos"*) est un suffixe grec qui signifie *"science"* ou *"discours"*. Dans le domaine philosophique ARISTOTE, a défini l'ontologie comme la science de l'Être. L'ontologie est une partie de la métaphysique (McCarthy et Hayes (1969)) qui s'applique à *"l'Être en tant qu'Être"*, indépendamment de ses déterminations particulières. L'ontologie est habituellement comprise comme une science des *étants* (Être en tant que phénomène) que comme une science de l'Être en tant qu'Être. Elle s'intéresse d'avantage à ce qui existe (les étants ou existants) qu'aux principes de ce qui existe (l'Être).

L'ontologie en tant que domaine est la partie de la philosophie, qui s'intéresse à la nature ainsi qu'à l'organisation à priori de la réalité (Guarino et Poli (1995)). La nature et l'organisation sont indépendantes de la forme des connaissances. De même, elles sont aussi indépendantes des utilisations ou des champs d'application.

Sowa (1999) fait de l'ontologie, en tant que domaine, l'étude des catégories d'entités abstraites et concrètes qui existent ou peuvent exister. Par conséquent, l'ontologie porte sur une abstraction structurée de la réalité.

Le champ d'application de l'ontologie est l'étude du sens de ces connaissances. L'ontologie est distincte de l'épistémologie. Cette dernière représente l'étude de la nature des connaissances et de leur justification (Sowa (1999)). Elle se base sur la structure des connaissances pour en produire de nouvelles.

1.2.2 Conceptualisation

L'ontologie, au sens philosophique, a été une source d'inspiration pour plusieurs domaines. Ces derniers incluent l'acquisition, la représentation théorique et le partage des connaissances (Chandrasekaran *et al.* (1999)), ainsi que pour le traitement du langage

naturel. Dans ces domaines, une ontologie est généralement synonyme de modèle conceptuel. Welty et Guarino (2001) définissent un modèle conceptuel comme l'implémentation d'une ontologie. Cette implémentation doit satisfaire les contraintes imposées par l'application exploitant l'ontologie. En revanche, une ontologie est indépendante des contraintes d'exécution, son objectif principal est de spécifier la conceptualisation du monde qui est sous-jacente à l'application (Welty et Guarino (2001)). Ils rejoignent la première définition proposée par Gruber (1993), qui définit l'ontologie comme *"une spécification explicite d'une conceptualisation"*. Le terme *conceptualisation* représente un modèle abstrait d'un domaine d'intérêt, dont les concepts pertinents sont identifiés et recensés. Le mot *explicite* implique que les types de concepts ainsi que les contraintes exprimées sur les concepts soient explicitement définies. Le terme *formelle* indique que l'ontologie doit être décrite dans un format (ou un langage) compréhensible par la machine. Le mot *partagé* fait référence au fait que l'ontologie doit capturer une partie consensuelle de la connaissance acceptée par toute ou une large partie de la communauté derrière l'ontologie. Hafner et Fridman (1996), Noy et Hafner (1997) et Chandrasekaran *et al.* (1999) définissent l'ontologie comme une théorie du contenu sur les sortes d'objets, les propriétés de ces objets ainsi que les relations possibles dans un domaine spécifique de connaissances. Une ontologie fournit les termes pour décrire les connaissances sur ce domaine.

La validité d'une ontologie dans le domaine philosophique est absolue, tandis que celle du domaine de l'ingénierie des connaissance dépendait d'un consensus (c'est à dire une validité restreinte). Ce point amène Gruber et Olsen (1994) à proposer une seconde définition d'une ontologie comme *"une spécification explicite d'une conceptualisation partagée"*. Le terme *partagé* fait référence au fait que l'ontologie doit capturer une partie consensuelle de la connaissance acceptée par toute, ou une large partie de, la communauté de l'ontologie. En conséquence, dans un même domaine, plusieurs ontologies concurrentes peuvent co-exister. Guarino et Giaretta (1995) ajoutent que la conceptualisation ne peut être que partielle et proposent la définition suivante : *"une ontologie est une conceptualisation explicite et partagée"*. Gruber (2009) améliore ses propres définitions (Gruber (1993), Gruber et Olsen (1994)) et fournit la définition suivante : *"dans le contexte des sciences de l'informatique et de l'information, une ontologie définit un jeu de primitives représentatives avec lequel un domaine de connaissances ou un univers de discours peut être modélisé"*. En effet, le jeu de primitives évoqué, n'est qu'une traduction, la plus fidèle que possible du monde réel à représenter sous forme d'une ontologie.

Une ontologie peut être synonyme d'un modèle conceptuel. Certains auteurs emploient

également ce terme pour désigner une représentation de modèle, dans le sens du modèle conceptuel de Welty et Guarino (2001). Sowa (1999) formalise une ontologie comme un catalogue des types de choses appartenant à un domaine. De même, Rector (1999) retient aussi cette définition. Il analyse la difficulté de définir des formalismes pour la représentation des concepts et leur peuplement avec des connaissances. Deux conceptualisations peuvent avoir la même représentation dans le cas où cette dernière ne prend pas compte des nuances. Dans le cas inverse, une conceptualisation peut admettre plusieurs représentations différentes (Guarino et Giaretta (1995)).

1.2.3 Ontologie formelle

Dans le cas où une ontologie est une représentation d'une conceptualisation, la rigueur et l'utilisation d'ontologies par des programmes poussent à définir des principes portant à la fois sur la conceptualisation et sur sa représentation. L'adoption de principes différents dans les deux cas contribue au caractère relatif des ontologies développées dans le domaine de l'ingénierie des connaissances.

Les principes rigoureux de la modélisation des connaissances répondent aux besoins de partager et d'utiliser les connaissances dans différents contextes. Une ontologie formelle est une théorie des distinctions formelles entre les éléments d'un domaine, indépendamment de leur réalité (Guarino (1997)).

Les principes de représentation permettent une organisation précise des connaissances. En effet, une ontologie formelle est composée d'une collection de noms pour les types de concepts et de relations. Ils sont organisés dans un ordre partiel par la relation *"type/sous-type"* (Sowa (1999)), par opposition à une ontologie informelle constituée d'un catalogue de types qui sont soit non définis, soit définis par des assertions en langage naturel. Guarino et Giaretta (1995) évoquent alors la théorie ontologique pour désigner une théorie logique cherchant à exprimer des connaissances ontologiques.

Guarino et Poli (1995) reprennent la définition d'une ontologie formelle proposée par Cocchiarella (1991) et définissent l'ontologie comme : *"le développement systématique, formel et axiomatique de la logique de toutes les formes et les modes d'existences"*. Cette définition combine les deux nuances de *"formel"* à la fois comme synonyme de rigueur et de description de la forme. Aussi, une ontologie formelle s'intéresse moins à l'existence de certains individus qu'à une description rigoureuse de leur forme. Une ontologie est donc une théorie des distinctions a priori entre les entités du monde et entre les catégories utilisées pour la modélisation du *monde* (Guarino et Poli (1995)). Lorsque la modélisation

et la représentation d'une ontologie reposent sur des principes mathématiques, les deux aspects de *"formel"* sont pris en compte. La section suivante s'intéressera à la description détaillée des différents constituants d'une ontologie.

1.3 Constituants d'une ontologie

Chandrasekaran *et al.* (1999) décrivent les éléments constituants l'ontologie comme suit : les connaissances et les domaines de connaissances, les concepts, les relations et les connaissances inférentielles. Les concepts de l'ontologie sont caractérisés par des propriétés (attributs) qui les décrivent. Ces concepts peuvent être associés via des relations. Les concepts et les relations peuvent évoluer au cours du cycle de vie de l'ontologie. Ces évolutions mettent en jeu des événements et des processus, éventuellement associés par une relation de causalité.

1.3.1 Connaissances et domaines de connaissances

Une ontologie ne peut être construite que dans le cadre d'un domaine précis de la connaissance avec une sémantique qui doit être intégrée. Un domaine de connaissances est constitué par les objets du domaine et par un contexte d'usage de ces objets. Délimiter rigoureusement un domaine de connaissances, peut se révéler ardu. Cette difficulté provient de la nature holistique de la connaissance. Certaines connaissances sont utilisées dans plusieurs domaines. De plus, les connaissances humaines se déploient suivant plusieurs aspects. Les connaissances peuvent être développées non seulement sur la réalité mais également sur un domaine de connaissances (méta-connaissances ou connaissances sur les connaissances). Borner le domaine de connaissances à représenter, reviendrait à une délimitation précise de l'objectif opérationnel de l'ontologie.

Dans le domaine de l'ingénierie de connaissances, le terme connaissance possède un sens restreint. Les connaissances considérées sont susceptibles d'être formalisées (Bachimont (2000)). La manipulation automatique de connaissances ne paraît pertinente que dans le cas des connaissances objectives, *i.e.*, elle fait l'objet d'un consensus large. Plus précisément, la connaissance n'est considérée en tant que telle que dans le cas où l'information stockée dans la machine prend un sens pour l'utilisateur. Ce dernier peut établir un lien entre cette information et celle qu'il possède déjà et ce sens doit être le même pour tous les utilisateurs (Charlet (2001)). Cette notion de lien sémantique explique d'ailleurs le développement des modèles de représentation de types réseaux sémantiques.

Les connaissances manipulées dans les systèmes à base de connaissances doivent être des informations *"actives"*. Ces dernières sont susceptibles d'influencer le déroulement du processus de la production de nouvelles informations ou de la prise de décisions (Kayser (1997)). Ce type de connaissances *"productives"* offre un intérêt pour l'automatisation. La construction d'une ontologie doit se faire à partir d'un champ de connaissances bien délimité par un objectif opérationnel clair. Elle doit aussi porter sur des connaissances objectives, dont la sémantique peut être exprimée rigoureusement et formellement. De ce fait, plusieurs types d'ontologies peuvent être distinguées en fonction des différents objectifs opérationnels recensés.

Au sein d'une même ontologie, différents types de connaissances cohabitent. En effet, chaque domaine de connaissances intègre sa propre terminologie, des faits connus, des règles intangibles, des heuristiques, etc. Les ontologistes s'accordent pour considérer que les primitives cognitives de base d'une ontologie sont les concepts, les relations entre ces derniers et les connaissances inférentielles. La sous-section suivante décrit les concepts de l'ontologie ainsi que les principales propriétés de ces derniers.

1.3.2 Concepts

Les connaissances du domaine portent sur des objets. Les objets sont référés à travers des concepts. Un concept peut représenter un objet matériel, une notion ou encore une idée (Uschold et King (1995)). Un concept peut être divisé en trois parties : un terme ou plusieurs, une notion et un ensemble d'objets. Les termes sont aussi appelés *labels* ou *étiquettes* du concept. La notion est appelée aussi *intension du concept*. Elle contient la sémantique du concept. Elle est exprimée en termes de propriétés et d'attributs, de règles et de contraintes. L'ensemble d'objets est appelé aussi *extension du concept*. Il regroupe les objets manipulés à travers le concept. Ces objets sont les instances du concept. Par exemple, le terme "table" renvoie à la fois à la notion de table comme objets de type "meuble" possédant un plateau et des pieds, et à l'ensemble des objets de ce type.

Les deux aspects d'un concept (intension et extension) sont assez différents. En effet, deux extensions peuvent ne pas être disjointes, alors que deux intensions s'excluent mutuellement par au moins une propriété. Bachimont (2000) utilise le terme *concept formel* pour désigner l'extension d'un concept et *de concept sémantique* pour désigner l'intension du concept. Un concept est doté d'une sémantique référentielle et d'une sémantique différentielle. La sémantique référentielle est imposée par l'extension du concept tandis que la sémantique différentielle est imposée par l'intension du concept.

Il est à noter qu'un concept peut avoir une extension vide. Ce concept est un concept générique. Il correspond à une notion abstraite (par exemple, la "vérité", prise dans le sens de "ce qui est vrai" et non pas du "degré de vérité"). Deux concepts peuvent partager la même extension sans pour autant avoir la même intension. Ce cas est illustré par les concepts "étoile du matin" et "étoile du soir", désignent tous les deux l'astre Vénus. De plus, des concepts partageant la même extension mais pas leur intension peuvent être désignés par le même terme. Ceci correspond à des points de vue différents sur un même objet. Par exemple, les chiens peuvent être considérés comme des animaux de compagnie, ou comme des ressources culinaires.

Le langage naturel contient de nombreux termes désignant plusieurs concepts sémantiques différents. Cependant, de telles ambiguïtés ne sont pas gérables par machine. En effet, la machine identifie généralement un concept à l'aide de ses termes. La restriction à un domaine de connaissances permet généralement d'éviter l'homonymie de concepts. Il apparaît par contre souhaitable de gérer les synonymies et de permettre la désignation d'un concept par plusieurs termes. Ceci assure une plus grande souplesse d'utilisation de l'ontologie (Gómez-Pérez *et al.* (1996)). D'une manière générale, l'articulation entre les aspects référentiels et différentiels des concepts est délicat. Certains auteurs estiment que l'intension d'un concept permet à elle seule d'en préciser le sens (Guarino et Welty (2000)). En revanche, d'autres jugent indispensable de spécifier un concept à travers les deux paradigmes sémantiques (Kassel et Perpette (1999)).

Certaines propriétés sont essentielles à la caractérisation d'un concept. En effet, la suppression de ces propriétés entraîne la disparition du concept (Bouaud *et al.* (1995)). D'autres propriétés peuvent être considérées pour la caractérisation du concept dans un contexte donné. Ces propriétés ne sont vraies que dans ce cadre et leur disparition ne modifie pas intrinsèquement le concept. Guarino et Giaretta (1995) ne considèrent comme ontologique que les propriétés nécessaires. Kassel (2002) admet dans les ontologies des propriétés incidentes. Ces propriétés incidentes ne sont valides que dans le cadre applicatif. Ces propriétés incidentes sont nécessaires à la complétude de la définition du concept, qui est plus spécifique que celui considéré dans le cadre général.

Les concepts manipulés dans un domaine de connaissances sont organisés au sein d'un réseau de concepts. L'ensemble des concepts est structuré hiérarchiquement et les concepts sont liés par des propriétés conceptuelles. La propriété utilisée pour la structuration de la hiérarchie des concepts est la subsomption. Un concept C_1 subsume un concept C_2 si toute propriété sémantique de C_1 est aussi une propriété sémantique de C_2, c'est à dire

C_2 est plus spécifique que C_1. L'extension d'un concept est forcément plus réduite que celle d'un concept qui le subsume tandis que son intension est par contre plus riche en instances.

Les principales propriétés sur les concepts peuvent être classées en deux groupes : les propriétés sur un seul concept et les propriétés sur deux concepts.

Les propriétés portant sur un seul concept sont (Guarino *et al.* (1994a)) :

- **La généricité** : un concept est générique s'il n'admet pas d'extension, c-à-d., d'instances (par exemple "la vérité" est un concept générique) ;
- **L'identité** : Un concept a une propriété d'identité s'il dispose d'une propriété permettant de distinguer une instance d'une autre (par exemple le concept "étudiant" porte une propriété d'identité liée au numéro de l'étudiant, deux étudiants sont identiques s'ils possèdent le même numéro) ;
- **La rigidité** : un concept est rigide si toute instance de ce concept en reste instance dans tous les domaines (par exemple le concept "humain" est rigide, par contre le concept "étudiant" est non rigide) ;
- **L'anti-rigidité** : un concept est anti-rigide si toute instance de ce concept est essentiellement définie par son appartenance à l'extension d'un autre concept (par exemple le concept "étudiant" est un concept anti-rigide car l'étudiant est avant tout un "humain") ;
- **L'unité** : un concept est unité si, pour chacune de ces instances, les différentes parties de l'instance sont reliées par une relation qui ne lie pas d'autres instances de concepts (par exemple les deux parties d'un couteau, manche et lame sont liées par une relation "enmanché" qui ne lie que cette lame et cette manche).

Les propriétés portant sur les deux concepts sont les suivantes :

- **L'équivalence** : deux concepts sont équivalents s'ils possèdent la même extension ;
- **La disjonction** : deux concepts sont disjoints si leurs extensions sont disjointes (par exemple le concept "homme" et le concept "femme" sont deux concepts disjoints) ;
- **La dépendance** : deux concepts sont dépendants si pour toute instance du premier concept il existe une instance du second concept qui ne soit ni partie ni constituant du premier concept (par exemple le concept "parent" est un concept dépendant du concept "enfant" et vice-versa).

Certaines propriétés peuvent s'appliquer aux propriétés elles-mêmes (Guarino et Welty (2000)). Welty et Guarino (2001) considèrent les concepts et les relations comme des propriétés. Les propriétés précédentes sont définies comme des méta-propriétés portant

sur n'importe quelle propriété. Welty et Guarino (2001) définissent, à l'aide de ces méta-propriétés, des types de concepts à savoir les sortes et les non sortes.

Les sortes se décomposent en (Welty et Guarino (2001)) :

- **Le type** : est concept rigide et portant sa propre identité (par exemple les concepts "homme", "chat" et "eau" sont des types) ;
- **Le quasi-type** : est un concepts anti-rigide et portant sa propre identité (par exemple le concept "herbivore") ;
- **Le modèle matériel** : est un concept anti-rigide et dépendant (par exemple les concepts "étudiant" et "nourriture" sont des modèles matériels) ;
- **La phase** : est un concept anti-rigide, portant sa propre identité et indépendant (par exemple le concept "papillon") ;
- **Le mixte** : est un concept non rigide et portant sa propre identité (par exemple le concept "male").

Les non sortes se déclinent en trois ensembles (Welty et Guarino (2001)) :

- **La catégorie** : est un concept rigide et ne portant pas sa propre identité (par exemple les concepts "chose" et "lieu") ;
- **Le modèle formel** : est un concept anti-rigide et dépendant (par exemple les concepts "acteur" et "patient") ;
- **L'attribution** : est un concept non rigide (par exemple le concept "rouge").

Il est possible de classer les propriétés à l'aide d'autres critères. Welty et Guarino (2001) proposent de distinguer les propriétés intrinsèques et les propriétés extrinsèques. Les propriétés intrinsèques ne sont liées qu'au concept lui même, comme la généricité. Les propriétés extrinsèques font intervenir d'autres concepts dans leur définition. Des modalités peuvent être introduites au niveau des propriétés qui ne sont pas de nature modale, comme l'identité. Un concept peut porter possiblement (dans certains mondes) ou nécessairement (dans tous les mondes possibles) sur un critère d'identité.

En plus des propriétés, l'intension d'un concept peut contenir des attributs. Par exemple, un chien possède quatre attributs pattes et un attribut queue. Un attribut peut être une instance de concept. Par exemple, "un président de la République Tunisienne" a toujours le Palais de Carthage comme "résidence officielle". D'autre part, il est possible d'affecter des attributs (concepts ou instances) à une instance d'un concept (Gómez-Pérez *et al.* (1996)). Par exemple, "Ferrari" est une instance de "voiture" qui porte un attribut "couleur rouge" instance du concept "couleur".

Certains liens conceptuels existant entre les concepts peuvent s'exprimer à l'aide de

propriétés portées par les concepts. D'autres liens doivent être représentés à l'aide de relations autonomes. Ces relations ainsi que leurs propriétés sont décrites dans la sous-section suivante.

1.3.3 Relations

Une relation permet de lier des instances de concepts, ou des concepts génériques. Une relation est caractérisée par un terme (voir plusieurs) et une signature. Les termes sont les labels ou les étiquettes de la relation. La signature précise le nombre d'instances de concepts, que la relation lie, leurs types et l'ordre des concepts (c'est à dire la façon dont la relation doit être lue). Par exemple, la relation "écrit" lie une instance du concept "personne" et une instance du concept "texte", dans cet ordre.

La décision de la représentation d'une notion sous forme d'un concept ou d'une relation peut parfois s'avérer nécessaire. Le choix dépend essentiellement de l'objectif opérationnel de l'ontologie et du contexte d'utilisation de la notion. De même, il convient de choisir entre l'utilisation de relations ou d'attributs pour la représentation des liens entre concepts.

Tout comme les concepts, les relations peuvent être caractérisées par des propriétés. Les relations sont organisées de manière hiérarchisée à l'aide de la propriété de subsomption décrite précédemment. Les propriétés des relations sont classées en trois groupes : les propriétés intrinsèques, les propriétés liant deux relations et les propriétés liant une relation et des concepts.

Les propriétés intrinsèques sont les suivantes :

- **Les propriétés algébriques** : comme la symétrie, la réflexivité et la transitivité ;
- **La cardinalité** : indique le nombre possible de relation de ce type entre les mêmes concepts ou instances du concept.

Les propriétés liant deux relations sont :

- **L'incompatibilité** : deux relations sont incompatibles si elles ne peuvent lier les mêmes instances de concepts (par exemple les relations "être rouge" et "être vert" sont incompatibles) ;
- **L'inverse** : deux relations binaires sont inverses l'une de l'autre si, l'une lie deux instances I_1 et I_2, l'autre lie I_2 et I_1 (par exemple les relations "a pour père" et "a pour enfant" sont inverses l'une de l'autre) ;
- **L'exclusivité** : deux relations sont exclusives si, l'une lie des instances de concepts, l'autre ne lie pas ces instances, et vice-versa (par exemple "l'appartenance" et "la non appartenance" sont deux relations exclusives). L'exclusivité entraîne l'incompatibilité.

Kassel (2002) propose les propriétés liant une relation et des concepts. Ces propriétés sont comme suit :

- **Le lien relationnel** : il existe un lien relationnel entre une relation R et deux concepts C_1 et C_2. Pour tout couple d'instances des concepts C_1 et C_2, il existe une relation de type R qui lie les deux instances de C_1 et C_2. Un lien relationnel peut être contraint par une propriété de cardinalité, ou porter directement sur une instance de concept (par exemple il existe un lien relationnel entre les concepts "texte" et "auteur" d'une part et la relation "a pour auteur" d'autre part) ;
- **La restriction de relation** : Il existe une restriction de relation entre une relation R et un concept C, si pour tout concept de type C, et toute relation de type R liant C, les autres concepts liés par la relation sont d'un type imposé, (*e.g*, si la relation "mange" portant sur une "personne" et un "aliment" lie une instance de "végétarien", concept subsumé par "personne", l'instance de "aliment" est forcément instance de "végétarien").

Les propriétés des concepts et des relations complètent la sémantique différentielle de l'ontologie. Elles contribuent à préciser les liens et les différences entre les primitives cognitives du domaine de connaissances. Les aspects différentiels et référentiels sont cependant fortement imbriqués.

En plus des paradigmes différentiels et référentiels, Bachimont (2000) définit le paradigme opérationnel d'une ontologie. Au niveau opérationnel, les concepts et les relations sont définis par les opérations qu'il est possible d'appliquer. C'est à ce niveau que l'aspect dynamique des connaissances (connaissances inférentielles) va être représenté. La sous section suivante présentera les différents types des connaissances inférentielles.

1.3.4 Connaissances inférentielles

La description des connaissances en termes de concepts, de relations entre ces concepts et des propriétés sur ces concepts et relations ne suffit pas pour atteindre l'objectif opérationnel d'un système à base de connaissances. En effet, le centre d'intérêt du modélisateur n'est pas les connaissances en elles-mêmes mais leur concrétisation dans une action pour atteindre un objectif (Teuliet et Girard (2001)). Il s'agit de bénéficier de la spécificité du support informatique par rapport au support traditionnel, c'est à dire son aspect dynamique. Comparé à l'oral qui n'offre pas de support, et à l'écrit traditionnel, un système informatique peut manipuler des connaissances pour en inférer de nouvelles (Bachimont (1999)).

Les connaissances opérationnelles peuvent être des faits, des règles, ou des contraintes. Un fait est un énoncé vrai et implicatif. Un fait est un axiome qui participe à la description du monde cognitif dans lequel s'inscrit le système à base de connaissances. Une règle permet d'inférer de nouvelles connaissances. Elle contient donc une implication. Les contraintes permettent la spécification des impossibilités ou des obligations. Le choix entre l'utilisation d'une règle ou d'une contrainte pour la représentation d'une connaissance n'est pas toujours facile. Il dépend de la façon dont les connaissances vont être utilisées au niveau opérationnel (ou applicatif).

Certains auteurs considèrent que ces types de connaissances doivent être intégrés dans les propriétés des concepts, des relations ou des instances de concepts. En effet, un fait apparaît comme une propriété d'une instance de concept. Une règle apparaît plutôt comme une propriété d'un concept ou d'une relation car elle ne porte pas que sur les instances. Par exemple, une règle peut servir à la détermination de la valeur d'un attribut d'un concept (Gómez-Pérez *et al.* (1996)). Les contraintes peuvent porter aussi bien sur des concepts ou relations que sur des instances de concepts. Maedche *et al.* (2000) proposent d'intégrer les axiomes dans les ontologies comme des objets liés aux concepts et relations, de manière à apparaître comme des propriétés.

Ces choix de représentation des connaissances de raisonnement sont ajoutés à un autre choix portant sur le mode d'utilisation de ces connaissances. En effet, lors de la manipulation des connaissances, une bonne partie de nos réflexions sont implicites. Les connaissances inférentielles se classent en connaissances implicites et en connaissances explicites. Les connaissances implicites sont utilisées par le système mais non exprimées à l'intention des utilisateurs. Les connaissances explicites doivent être exprimées aux utilisateurs. Le choix est lié au contexte opérationnel de l'ontologie.

Les différents éléments utilisés pour la représentation des connaissances forment un ensemble complexe. Leur mise en œuvre demande que des choix clairs soient pris avant tout travail d'ontologisation. Ces choix de représentation doivent servir l'objectif opérationnel de l'ontologie et être validés dans le cadre d'usage de l'ontologie.

Les ontologies représentées sous forme d'une taxonomie de concepts et de relations sont des ontologie légères, *lightweight ontologies* (Gómez-Pérez *et al.* (2004)). Les ontologies formées d'une taxonomie à laquelle s'ajoute des connaissances inférentielles sont des ontologies lourdes, *heavyweight ontologies* (Gómez-Pérez *et al.* (2004)). Cette première classification est en fonction des constituants des ontologies. La section suivante s'intéressera à d'autres types de classification des ontologies selon d'autres critères de classification.

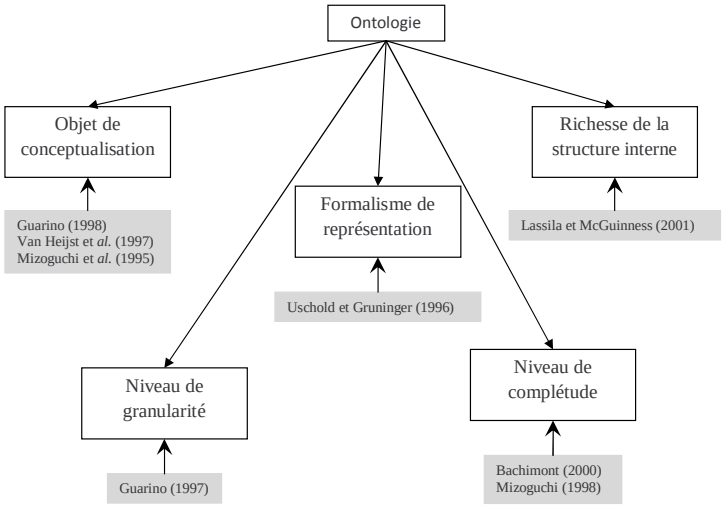


Figure 1.1 — Différentes dimensions de classification d'ontologies

1.4 Classification d'ontologies

La classification d'ontologies peut se faire selon plusieurs dimensions. Ces dimensions regroupent principalement cinq classes : objets de conceptualisation, niveau de granularité, formalisme de représentation, niveau de complétude et la richesse de la structure interne. La Figure 1.1 illustre les dimensions permettant la classification d'ontologies.

1.4.1 Classification selon l'objet conceptualisation

Initialement, Mizoguchi *et al.* (1995) ont proposé une classification d'ontologies en quatre catégories à savoir :

- **Les ontologies de contenu** : permettent la réutilisation des connaissances. Ces ontologies incluent d'autres sous-catégories telles que les ontologies de tâches, les ontologies de domaine, les ontologies génériques et communes ;
- **Les ontologies de communication** : (informer et s'informer) assurent le partage des connaissances ;
- **Les ontologies d'indexation** : permettent la recherche d'informations ;
- **Les méta-ontologies** : sont équivalentes aux ontologies de représentation des connaissances.

La classification de Van Heijst *et al.* (1997), illustrée par la Figure 1.2, repose sur deux dimensions orthogonales à savoir : la structure et le sujet d'une conceptualisation. Dans la première dimension, les auteurs distinguent trois catégories d'ontologies :

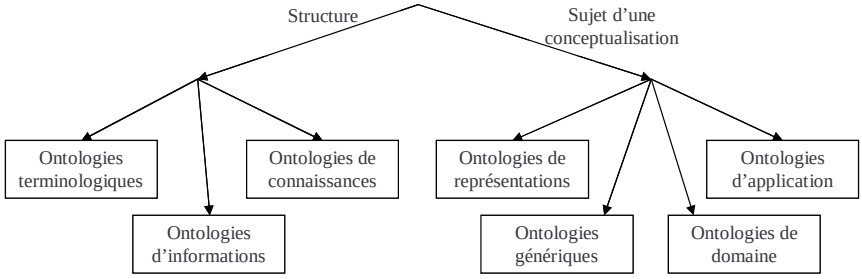


Figure 1.2 — Classification d'ontologies proposée par Van Heijst *et al.* (1997)

- **Les ontologies terminologiques** (par exemple les lexiques) ;
- **Les ontologies d'information** (par exemple les schémas de bases de données) ;
- **Les ontologies de modélisation de connaissances** (par exemple les conceptualisations de connaissances).

Dans la deuxième dimension, quatre catégories sont identifiées :

- **Les ontologies de représentation** : modélisent les représentations primitives utilisées pour la formalisation des connaissances sous un paradigme donné,
- **Les ontologies génériques** : comportent des concepts abstraits (généraux) subsumant les concepts existant dans différents domaines ;
- **Les ontologies de domaine** : expriment des conceptualisations spécifiques à un domaine particulier (par exemple les ontologies du e-learning) ;
- **Les ontologies d'application** : sont les ontologies les plus spécifiques, elles contiennent des concepts dépendants d'un domaine et d'une tâche particulière (par exemple dans le contexte du e-learning).

Guarino (1998) classifie les ontologies selon leur niveau de dépendance à une tâche particulière ou à un point de vue. Ainsi, il distingue quatre type d'ontologies (voir figure 1.3) :

- **Les ontologies de haut niveau** : expriment des conceptualisations valables dans différents domaines et décrit des concepts généraux ;
- **Les ontologies de domaine** : expriment des conceptualisations spécifiques à un domaine particulier ;
- **Les ontologies de tâches** : fournissent un vocabulaire systématisé de termes employé pour la résolution de problèmes liés aux tâches, qui peuvent appartenir ou non au même domaine ;
- **Les ontologies d'applications** : sont les ontologies les plus spécifiques, elles

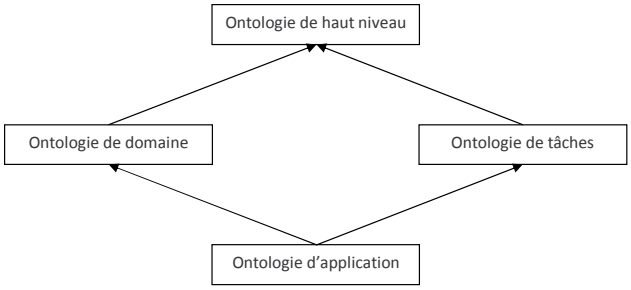


Figure 1.3 — Classification d'ontologies proposée par Guarino (1998)

contiennent des concepts dépendants d'un domaine et d'une tâche particulière ;

Une distinction est établie entre les ontologies de domaine portant sur des concepts renvoyant à des objets matériels ou à des concepts de bas niveau, *lower ontologies*, c-à-d., n'offrant que des possibilités limitées de raffinement, et les ontologies portant sur des concepts de haut niveau, *upper ontologies*. Ces dernières décrivent des notions générales comme les notions d'objet, de propriété, d'état, de valeur, de moment, d'événement, d'action, de cause et d'effet (Sowa (2000)). Parmi les ontologies de haut niveau, il existe des ontologies qui décrivent les notions utilisées dans toutes les ontologies pour la spécification des connaissances, telles que les sortes, les substances, les concepts, les relations (Guarino (1998)), etc. Ces ontologies de représentation sont indépendantes des différents domaines de connaissances. Elles décrivent des primitives cognitives communes aux différents domaines (Guarino *et al.* (1994b)).

1.4.2 Classification selon le niveau de granularité

La granularité est un autre critère pour la classification d'ontologies. La granularité représente le niveau de détail de la conceptualisation de l'ontologie. En fonction de l'objectif opérationnel, une connaissance plus ou moins fine du domaine est nécessaire. Des propriétés considérées comme secondaires dans certains contextes peuvent se révéler indispensables pour d'autres applications. Deux catégories peuvent être identifiéeés à savoir la granularité fine et la granularité large.

La granularité fine correspond à une ontologie très détaillée. Cette ontologie possède ainsi un vocabulaire plus riche. Ce vocabulaire assure une description détaillée des concepts pertinents d'un domaine particulier (ou d'une tâche particulière). Ce niveau de granularité s'avère utile lorsqu'il s'agit d'établir un consensus entre les agents qui manipulent l'ontologie.

La granularité large correspond à une ontologie avec un vocabulaire moins détaillé. Il coïncide avec des scénarios spécifiques d'utilisation, où les différents utilisateurs sont préalablement d'accord à propos de la conceptualisation sous-jascente (Guarino (1997)). Les ontologies génériques possèdent une granularité large. En effet, compte tenu des concepts qu'elles modélisent sont normalement raffinées subséquemment dans d'autres ontologies de domaine ou d'application. De même, les ontologies de haut niveau possèdent une granularité large. Elle provient du fait que les notions sur lesquelles elles portent peuvent être raffinées par des notions plus spécifiques.

1.4.3 Classification selon le niveau de formalisme de représentation

Les ontologies sont aussi classées en plusieurs niveaux d'expression selon leur utilisation (Uschold et Gruninger (1996), Mhiri *et al.* (2006)) :

- **L'ontologie très informelle** : est exprimée dans un langage naturel (sémantique ouverte) ;
- **L'ontologie semi-informelle** : est exprimée sous une forme restreinte et structurée de langage naturel pour augmenter la clarté et pour réduire l'ambiguïté ;
- **L'ontologie semi-formelle** : est exprimée en langage formel ;
- **L'ontologie rigoureusement formelle** : est définie avec une sémantique formelle à travers un langage artificiel formel, permettant les théorèmes et les preuves telles que la robustesse et l'exhaustivité.

1.4.4 Classification selon le niveau de complétude

La classification selon le niveau de complétude est proposée, respectivement, par Bachimont (2000) et Mizoguchi (1998). Bachimont (2000) propose une classification orientée sur la construction de la signification. Cette classification repose sur trois niveaux : le niveau sémantique, le niveau formel et le niveau opérationnel.

Le niveau sémantique (ou impératif) caractérise une ontologie régionale (Bachimont (2000)). Les concepts de ce type d'ontologies sont caractérisés par des libellés. Ils respectent les quatre principes différentiels *(i)* communauté avec l'ancêtre ; *(ii)* différence de spécification par rapport à l'ancêtre ; *(iii)* communauté avec les concepts frères qui sont situés sur le même niveau ; et *(iv)* différence par rapport aux frères. Ces principes correspondent à l'engagement sémantique qui assure que chaque concept possède un sens

univoque.

Le niveau formel (ou référentiel) caractérise une ontologie référentielle (Bachimont (2000)). Les concepts référentiels (ou formels) se caractérisent, en plus des quatres caractéristiques différentielles (déjà énoncées), par un terme/libellé dont la sémantique est définie par une extension d'objets. L'engagement ontologique spécifie que les objets du domaine peuvent être associés au concept, conformément à la signification formelle. Deux concepts formels sont considérés comme identiques s'ils possèdent la même extension.

Le niveau opérationnel (ou computationnel) caractérise une ontologie computationnelle (Bachimont (2000)). Les concepts au niveau opérationnel ou computationnel se caractérisent, en plus des caractéristiques des concepts formels, par les opérations qu'il est possible de leur appliquer afin de générer des inférences (engagement computationnel). Deux concepts opérationnels sont identiques s'ils possèdent le même potentiel d'inférence.

Mizoguchi (1998) quant à lui propose une classification axée sur le processus d'ingénierie ontologique. Cette classification est basée sur trois niveaux : conceptuel, formel et opérationnel.

Le niveau conceptuel est spécifié en faisant abstraction de toute contrainte informatique et sert de support à l'acquisition des connaissances (Mizoguchi (1998)). Il représente une collection structurée de termes. La tâche fondamentale dans la construction d'une ontologie est l'articulation du monde d'intérêt, c'est à dire l'extraction des concepts et leur identification dans des hiérarchies. À ce stade, la modélisation est informelle, une spécification en langage naturel complétée de graphique est acceptable et des définitions minimales des concepts sont réalisées.

Le niveau formel est formalisé au moyen de langage de représentation interprétable par une machine (Mizoguchi (1998)). Outre des définitions du niveau conceptuel, des définitions formelles sont ajoutées pour empêcher des interprétations inattendues des concepts. De même, les relations ou les contraintes nécessaires sont également traduites en langage interprétable par la machine, pour former un ensemble d'axiomes. À ce stade, les relations sont plus riches que celles du niveau précédent. Les définitions sont déclaratives et formelles pour permettre aux ordinateurs de les interpréter. L'interprétation d'une ontologie à ce niveau permet aux ordinateurs de répondre aux questions sur les modèles construits à partir de l'ontologie.

Le niveau opérationnel est encodé au moyen d'un langage de programmation et la spécification obtenue est exécutable (Mizoguchi (1998)). L'ontologie est exécutable dans le sens où les modèles construits à partir de l'ontologie fonctionnent en utilisant des

modules fournis par les codes abstraits associés aux concepts de l'ontologie.

1.4.5 Classification selon la richesse de la structure interne

Lassila et McGuinness (2001) proposent une classification d'ontologies en fonction des données que l'ontologie décrit et la richesse de sa structure interne. Lassila et McGuinness (2001) proposent les catégories suivantes (voir figure 1.4) :

- **Le vocabulaire contrôlé** : est un ensemble de termes définis par un groupe de personnes ou une communauté (par exemple les catalogues) ;
- **Le glossaire** : représente un ensemble de termes avec leur signification ;
- **Le thésaurus** : est défini par un ensemble de termes organisés suivant un nombre restreint de relations. Ces relations pouvent être entre termes synonymes ou entre termes préférés. Afin d'uniformiser leur format de représentation, différentes normes spécifient les thésaurus monolingues et multilingues ;
- **L'hiérarchie informelle** : organise des catégories à partir de la notion générale de généralisation / spécification (par exemple l'hiérarchie proposée par *Yahoo* qui représente une catégorisation des thèmes de recherches). L'hiérarchie n'est pas formelle car l'hiérarchisation des catégories ne respecte pas la stricte notion de subsomption (Lassila et McGuinness (2001)) ;
- **L'hiérarchie formelle** (notée H.F.) : est une hiérarchie dont la structure est déterminée par des relations de généralisation ;
- **L'hiérarchie formelle avec instances** : est similaire à la catégorie précédente mais elle inclut des instances ;
- **Le *frame*** : représente une ontologie contenant des classes avec des propriétés pouvant être héritées ;
- **L'ontologie avec restrictions de valeurs** : est une ontologie contenant des restrictions sur les valeurs des propriétés ;
- **L'ontologie avec contraintes logiques** : est une ontologie pouvant contenir des contraintes entre les constituants, définies dans un langage logique.

Les ontologies peuvent être décrites à travers plusieurs langages. Ces langages offrent différents niveaux d'expressivité et la richesse de l'ontologie. Un certain nombre de langages offre des mécanismes de raisonnement. La section suivante passera en revue les principaux langages de représentation des ontologies.

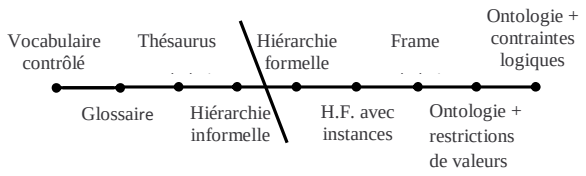


Figure 1.4 — Classification de Lassila et McGuinness (2001)

1.5 Langages de représentation

Dans la littérature plusieurs langages ont été utilisés pour la description des ontologies. Ces langages regroupent le langage de représentation des hiérarchies XML (eXtensible Markup Language, Marsh (2001)), le langage RDF (Resource Description Framework, Klyne et Carroll (2004)), le langage DAML+OIL (Darpa Modeling Language of Ontology + Ontology Inference Layer, Connolly *et al.* (2001)) et le langage OWL (Ontology Web Language, Smith *et al.* (2004)). Ces langages offrent différents niveaux d'expressivité.

1.5.1 XML

Le langage XML possède une syntaxe pour la description des documents structurés. Il ne permet pas d'imposer des contraintes sémantiques à la signification des documents décrits. Le langage XML est un langage à balises.

Un schéma XML est une description du type d'un document XML. Le schéma contient un ensemble de règles. Ces règles sont des contraintes sur la structure et le contenu du document XML. Un document XML doit respecter le XML schema qui lui est dédié afin d'assurer la validité du document selon son schéma. Le DTD (*Document Type Definition*) permet de définir une grammaire permettant de vérifier la conformité du document XML. Le DTD et le schéma XML sont développés pour exprimer des schémas XML.

Un document XML contient l'espace de noms (*namespace*) qui est un contexte ou un conteneur abstrait. L'espace des noms contient des noms, des termes, des mots qui représentent des objets ou des concepts dans le monde réel. Un nom défini dans un espace de noms correspond à un et un seul objet. Deux concepts ou objets différents sont référencés par deux noms différents dans un même espace de noms. Le langage XML ne permet de présenter que la structure des documents. Il ne prend pas en charge la sémantique de l'information. La sous-section suivante décrit le langage de description de ressources RDF.

1.5.2 RDF

Le langage RDF est un modèle de méta-données pour référencer des objets (ressources) et les liens entre ces objets. Dans ce langage, les ressources sont identifiées par les URIs (*Uniform Ressource Identifier*). Ces ressources sont décrites par un triplet (sujet - prédicat - objet). Le sujet représente la ressource à décrire. Le prédicat est un attribut ou un aspect de la ressource. Il exprime un lien ou une relation entre le sujet et l'objet. L'objet est la valeur du prédicat. Les triplets RDF peuvent être représentés sous une forme graphique à travers un graphe ou décrits en XML.

Le schéma RDF est un langage qui permet de décrire des vocabulaires, des propriétés et des classes de ressources dans le modèle RDF. Le schéma RDF est une extension sémantique du langage RDF. Il assure des mécanismes pour la description des groupes de ressources similaires (classes) et des relations entre ces ressources (propriétés). Les descriptions du vocabulaire du schéma RDF sont écrites en RDF. Les deux langages RDF et schéma RDF sont référencés par RDF(S). Le RDF(S) possède des mécanismes pour la description des ressources ainsi que leurs caractéristiques.

Une des caractéristiques les plus importantes du schéma RDF est la possibilité de définir les liens de "subsomption" entre des classes et des relations. Ces liens sont assurés par les primitives `rdfs:subClassOf` et `rdfs:subPropertyOf`. Ces primitives sont des liens de *"spécialisation"* ou *"is_a"*. Elles permettent aux classes et aux relations d'hériter des caractéristiques définies dans les classes (ou des relations) parentes. Ce mécanisme assure le raisonnement dans RDF(S). La sous-section suivante présentera le langage de description des ontologies DAML+OIL.

1.5.3 DAML+OIL

DAML+OIL (Connolly *et al.* (2001)) est la fusion entre deux langages DAML et OIL. Le langage DAML a pour objectif de fournir les fondations pour la génération du Web sémantique. Le langage a adopté d'abord RDF(S) comme langage ontologique pour garantir l'interopérabilité sémantique. Comme RDF(S) n'est pas suffisamment expressif, relativement aux exigences du futur Web, un nouveau langage a été adopté nommé DAML-ONT (Darpa Modeling Language of Ontology). Ce dernier représente une extension du RDF(S) avec les capacités d'un langage de représentation des connaissances (Connolly *et al.* (2001)). Dans la même période, un groupe de chercheurs européens ont développé un langage d'ontologies appelé OIL. Ce langage a une syntaxe basée sur RDF. Il a été

construit de façon à ce que sa sémantique puisse être spécifiée à travers une description logique très expressive.

Ainsi, DAML+OIL est la combinaison de ces deux langages. Il doit ses primitives de modélisation intuitives aux *"frames"*. Il utilise la syntaxe des standards XML et RDF. Il adopte la logique de description pour exprimer la sémantique formelle et les mécanismes de raisonnement. D'un point de vue formel, DAML+OIL est basé sur une logique de description très expressive SHIQ (Horrocks *et al.* (2003)). En outre, DAML+OIL permet de définir un ensemble de constructeurs et d'axiomes, qui permettent respectivement de créer une ontologie et de détecter automatiquement les incohérences, et classer automatiquement les concepts d'une ontologie. Les principaux axiomes du langage DAML+OIL sont `daml:Class` (définition de classes), `daml:objectProperty` (création de propriétés), `daml:subClassOf` (définition de sous-classes), `daml:subPropertyOf` (définition de sous-propriétés), `daml:disjointWith` (disjonction entre classes) et `daml:inverseOf` (définition de deux propriétés inverses).

Cependant, DAML+OIL présente aussi les limites suivantes. En effet, DAML+OIL ne permet pas la composition et la fermeture transitive. Il prend en compte uniquement des relations unaires et binaires. Le langage DAML+OIL ne permet pas la déclaration des variables. Le langage DAML+OIL n'est pas un langage normalisé. Néanmoins, DAML+OIL est considéré comme le premier langage à fournir des mécanismes d'inférence sur les concepts d'ontologies.

L'apparition de ces nombreux langages, poursuivant pour la plupart des objectifs communs, a encouragé le World Wide Web Consortium (W3C) à recommander un langage standardisé plus puissant au niveau de l'expressivité. Ce langage est spécialement conçu pour la représentation des ontologies dans le cadre du Web sémantique. Le langage d'ontologies recommandé est le langage OWL. Il est dérivé du langage DAML+OIL. Le langage OWL couvre la plupart des caractéristiques du langage DAML+OIL et renomme la majorité de ses primitives. La sous-section suivante décrit le langage langage OWL de représentation des ontologies pour le Web.

1.5.4 OWL

OWL (Ontology Web Language) est un langage basé sur RDF. Il enrichit le modèle du schéma RDF en définissant un vocabulaire riche pour la description d'ontologies complexes. Le langage OWL est basé sur une sémantique formelle définie par une syntaxe rigoureuse. Le langage OWL est un langage pour la représentation des ontologies dans

le cadre du Web sémantique. Il représente une extension du vocabulaire de RDF(S). Le langage OWL est dérivé du langage d'ontologies DAML+OIL. Le langage OWL permet, en plus des primitives RDF(S), des relations entre les classes (disjonction, intersection, union, etc.), la cardinalité, l'égalité, etc. Le langage OWL permet la définition des types de propriétés (propriété d'objet, annotation, etc.), des caractéristiques des propriétés (par exemple la symétrique et la transitivité) et des classes énumérées.

Le langage d'ontologies OWL est décliné en trois sous-langages possédant une expressivité ascendante à savoir : OWL-Lite, OWL-DL (Ontology Web Language-Description Logic) et OWL-Full. La raison principale de cette division concerne la complexité, la calculabilité et l'implémentation du langage. Le sous-langage OWL-Lite possède la complexité formelle la plus basse et une expressivité minimale. Il est recommandé pour la représentation des thésaurus, des taxonomies ou hiérarchie de classes avec des contraintes simples. Le sous-langage OWL-DL est caractérisé par une expressivité maximale avec une complétude computationnelle (un temps de calcul fini). Le sous-langage OWL-DL est dédié pour la représentation d'ontologies nécessitant une puissance d'expressivité toute en garantissant la calculabilité. Le sous-langage OWL-Full est conçu pour les utilisateurs, qui nécessitent une expressivité maximale. Ce langage possède une syntaxe libre de RDF. Ce sous-langage ne garantit pas la calculabilité. Le OWL-Full est considéré comme une extension du langage RDF, tandis que les deux sous-langages OWL-Lite et OWL-DL sont considérés comme des extensions restreintes de RDF. Cependant, ce sous-langage OWL-Full est une extension de OWL-DL, qui est a son tour une extension de OWL-Lite. Tous les documents OWL sont des documents valides en RDF. Un document RDF est un document OWL-Full, par contre seulement quelques documents décrits avec le langage RDF sont des documents valides en OWL-Lite et OWL-DL.

1.5.5 Comparaison des langages de représentation

La comparaison des langages de représentation propose de les comparer sur leurs capacités d'expressivité. En effet, l'étude comparative permet de mettre en valeur les différentes primitives appartenant aux différents langages. La table 1.1 indique les primitives ainsi que leurs existences dans les langages.

Les langages RDF(S), DAML+OIL et OWL permettent la création de listes. Le langage RDF utilise les balises `rdf:Seq` et `rdf:Bag` pour la définition des listes. Ces balises n'indiquent pas que la liste est complète. Le langage DAML utilise "first/rest" pour la représentation des listes, avec la valeur nulle pour définir la fin de la liste.

	XML & DTD	XML schema	RDF(S)	DAML+OIL	OWL
Listes			x	×	×
Cardinalité	×	×		×	×
Expressions de classe				×	×
Types de données		×		×	×
Définition de classe				×	×
Énumérations	×	×		×	×
Equivalence				×	×
Extensibilité			×	×	×
Sémantiques formelles			×	×	×
Héritage			×	×	×
Inférence				×	×
Restrictions locales				×	×
Contraintes qualifiées				×	
Réification			×	×	×

Tableau 1.1 — Comparaison des langages de représentation des ontologies

Tous les langages à l'exception du langage RDF(S) expriment la cardinalité. La cardinalité représente le nombre de participations au même sujet ou prédicat. Les opérateurs de Kleene " ?" (0 ou 1), "*" (0+) et "+" (1+) dans XML et DTD assurent la définition des contraintes de cardinalité. Le langage DAML supporte les balises suivantes : `cardinality`, `minCardinality` et `maxCardinality`. Les expressions de classes assurent la création de nouvelles classes à partir d'autres existantes. DAML+OIL possède les expressions suivantes de création de classes : `unionOf`, `disjointUnionOf`, `intersectionOf` ou `complementOf`. Le langage RDF possède essentiellement les litéraux de type chaîne de caractères (string). Le langage DAML+OIL ajoute dans sa syntaxe les types de données du schéma XML. Le langage DAML permet la création d'une nouvelle classe de trois manières différentes. La première se base sur les valeurs de propriétés. La seconde est assurée par les restrictions imposées sur une classe déjà existante. La troisième est réalisée par expression de classes.

Les équivalences supportent le raisonnement à travers les ontologies. La balise `equivalentTo`, du langage DAML, est exploitée pour les classes, les propriétés et les instances. Les langages RDF, DAML+OIL et OWL sont extensibles. En effet, les langages

RDF et DAML permettent d'ajouter de nouvelles propriétés à des classes existantes. Le langage RDF est exploité pour la définition du langage DAML+OIL. De même, le langage DAML+OIL peut être utilisé pour la définition d'autres langages, comme le langage DAML-Logic. La sémantique formelle est exprimée dans le langage DAML+OIL à travers les modèles théoriques et axiomatiques.

Le langage XML Schema attribue des groupes à travers la DTD qui utilise des paramètres d'entités dans les définitions d'attributs, mais ce mécanisme ne représente pas la notion d'héritage pur. Les langages RDL, DAL+OIL et OWL supportent l'héritage à travers les balises `subClassOf` et `subPropertyOf`. Uniquement les langages DAML+OIL et OWL offrent des mécanismes d'inférence. Les balises `TransitiveProperty`, `UnambiguousProperty`, `inverseOf` et `disjointWith` génèrent des informations additionnelles pour les moteurs de raisonnement. Les balises `domain` et `range` du langage RDF assurent des contraintes sur les propriétés. Les langages DAML+OIL et OWL associent des restrictions avec les classes et les propriétés. Les contraintes qualifiées sont proposées uniquement dans le langage DAML. Elles permettent d'exprimer des contraintes telles que : *"tous les enfants de X sont de type Personne"*. Les propriétes `hasClassQ`, `cardinalityQ`, `minCardinalityQ` et `maxCardinalityQ` expriment les restrictions qualifiées telles que : *"au maximum trois enfants de X sont de type Docteur"*.

Dans le monde réel, plusieurs ontologies décrivent un domaine particulier de la connaissance. Ces ontologies peuvent être décrites à la fois dans plusieurs langages naturels, dans différents formalismes (ou langages) ontologiques, ou avec des degrés d'expressivité. Ces différences peuvent engendrer des problèmes d'hétérogénéité. La section suivante s'intéressera à la description des différentes formes d'hétérogénéité.

1.6 Problèmes issus de l'hétérogénéité

L'hétérogénéité n'est pas seulement due à la divergence des domaines que peuvent couvrir les ontologies mais aussi aux formalismes requis pour leurs développement. Dans la littérature, plusieurs classifications des types d'hétérogénéité sont recensées. Quelques classifications se basent sur l'étude du décalage sémantique et structurel (Euzenat et Shvaiko (2007)) qui peut exister entre les ontologies. D'autres classifications élucident le degré l'hétérogénéité selon les niveaux d'interopérabilité sémantique (Euzenat (2001)). La littérature recense quatre types d'hétérogénéités, à savoir : l'hétérogénéité syntaxique, l'hétérogénéité terminologique, l'hétérogénéité conceptuelle et l'hétérogénéité sémiotique.

L'hétérogénéité syntaxique se produit quand deux ontologies sont décrites avec deux langages ontologiques différents. Ce type d'hétérogénéité se manifeste lors de la comparaison d'un répertoire avec un modèle conceptuel. Cette hétérogénéité se produit aussi quand deux ontologies sont modélisées en utilisant des formalismes différents, par exemple OWL et F-Logic (Frame Logic). Cette classe d'hétérogénéité survient au niveau théorique, notamment, quand il s'agit d'établir des équivalences entre les primitives de différents langages ontologiques. Il est possible dans certains cas de traduire les ontologies dans différents langages ontologiques à condition de préserver la signification.

L'hétérogénéité terminologique se manifeste dans l'éventualité où deux entités sont référencées par deux noms différents alors qu'elles désignent le même objet. La cause d'une telle hétérogénéité, revient à l'utilisation de différents langages naturels, ou des sous-langages techniques spécifiques à un domaine de connaissances bien déterminé. Elle se manifeste aussi par l'utilisation des synonymies.

L'hétérogénéité conceptuelle est appelée aussi *hétérogénéité sémantique* (Euzenat (2001)) ou la *différence logique* (Klein (2001)). Elle concerne la diversité des modélisations d'un même domaine de connaissances. Elle découle principalement de l'utilisation de différents (ou équivalents) axiomes décrivant les concepts ontologiques. Elle se manifeste aussi lors de l'utilisation de concepts totalement différents. Respectivement, Klein (2001) et Visser *et al.* (1998) évoquent la différence de conceptualisation et la différence de l'explication. La différence de conceptualisation se manifeste à travers la différence entre les concepts inclus dans la modélisation. La différence des explications se base sur la manière avec laquelle les concepts sont exprimés. Visser *et al.* (1998) proposent une classification précise de ces différences.

Par ailleurs, Benerecetti *et al.* (2001) présentent les trois principales raisons de la différence de conceptualisation :

- *Différence de convergence* qui survient lorsque deux ontologies décrivent différentes connaissances avec le même niveau de détail pour une unique perspective ;
- *Différence de granularité* qui se produit quand deux ontologies décrivent le même domaine avec une même perspective mais avec différents degrés d'expression des détails ;
- *Différence de perspectives* qui se manifeste quand deux ontologies décrivent un même domaine, avec un même degré d'expression des détails mais avec des points de vue et des perspectives différents.

L'hétérogénéité sémiotique est appelée aussi *hétérogénéité pragmatique* (Euzenat et

Shvaiko (2007)). L'hétérogénéité sémiotique s'intéresse à la manière dont les entités ontologiques sont interprétées par leurs utilisateurs. Ainsi, les entités ayant les mêmes interprétations sémantiques peuvent être interprétées de différentes manières par l'Homme. Ces différences d'interprétation sont dues principalement à la diversité des contextes et des domaines d'application des ontologies. Par conséquent, la manière de mettre en oeuvre les entités ontologiques influence leurs interprétations. De plus, ce type d'hétérogénéité reste difficile à détecter par la machine.

1.7 Conclusion

Dans ce chapitre, nous avons présenté les concepts de base des ontologies. En outre, nous avons proposé une description détaillée des différents éléments décrivant les ontologies, et une classification d'ontologies selon plusieurs démarches. Il a aussi présenté une étude comparative des principaux langages utilisés dans la représentation des ontologies.

Dans la pratique, lors de l'exploitation de différentes ontologies décrivant un même domaine, il peut se produire à la fois plusieurs types d'hétérogénéité. Dans la littérature, il existe plusieurs travaux qui visent à réduire cette hétérogénéité selon ses aspects terminologiques et/ou conceptuels. Dans ce contexte, plusieurs techniques sont utilisées, à savoir : l'alignement, la fusion, l'intégration, la production de ponts d'axiomes, la traduction (Desprès et Szulman (2007), Euzenat et Shvaiko (2007)). Le chapitre suivant introduit en détail la notion d'alignement, qui est vue comme un processus capital pour garantir l'interopérabilité sémantique entre les ontologies.

CHAPITRE 2

Alignement d'ontologies

2.1 Introduction

L'alignement d'ontologies permet d'identifier un ensemble de correspondances entre les entités ontologiques appartenant aux deux ontologies à aligner. Les techniques d'alignement entre deux structures ont été développées dans de nombreux domaines. Dans le domaine des bases de données, la recherche de correspondances entre les schémas est évoqué pour des objectifs d'interopérabilité et d'intégration (Batini *et al.* (1986)). La recherche d'alignement entre les séquences du domaine biologique vise à identifier des portions communes ou similaires entre les séquences d'ADN, d'ARN ou protéines. La détection des isomorphismes entre les graphes est aussi une autre application des techniques d'alignement.

Plusieurs méthodes d'alignement spécifiques aux schémas (bases de données, objets et XML) et aux ontologies (RDF, OWL, etc.) sont recensées dans la littérature (Euzenat et Shvaiko (2007), Shvaiko et Euzenat (2005), Kalgoflou et Schorlemmer (2003), Rahm et Bernstein (2001), etc.). Les auteurs ont classé les méthodes d'alignement selon plusieurs critères tel que : les techniques et les stratégies employées, les types d'information représentées dans les structures à aligner et l'objectif de l'alignement.

Le présent chapitre est organisé en cinq sections. La section 2.2 propose les définitions de base des méthodes d'alignement ainsi que les mesures de similarités exploitées lors des processus d'alignement. Dans la section 2.3, les caractéristiques des paramètres en entrée et ceux en sortie des méthodes d'alignement d'ontologies sont présentées. La section 2.4 dresse la panoplie des techniques de base exploitées par les méthodes d'alignement lors du processus détermination des correspondances entre les deux ontologies à aligner. Dans la section 2.5, une description de plusieurs stratégies d'alignement. Ces stratégies regroupent des techniques de composition, d'agrégation, etc. La dernière section du chapitre (section 2.6) propose une étude comparative des principales méthodes d'alignement en se basant

sur la comparaison des caractéristiques internes, des techniques de base ainsi que les stratégies d'alignement.

2.2 Définitions

Cette section regroupe les définitions relatives aux méthodes d'alignement. Elle introduit aussi les différentes mesures de similarité (similarité, dissimilarité et distance) ainsi que leurs caractéristiques.

2.2.1 Méthodes d'alignement

Les champs d'application de l'alignement relèvent de plusieurs domaines. Ces domaines regroupent l'intelligence artificielle, les systèmes d'information et les bases de données (Euzenat et Shvaiko (2007)). Rahm et Bernstein (2001) présentent une méthode d'alignement, de schémas de bases de données, comme une fonction considérant deux schémas de bases de données. Cette fonction produit un appariement entre les éléments du premier schéma avec ceux du second. Le résultat représente un ensemble de correspondances qui exprime une relation entre un élément du premier schéma et un élément du second schéma. Shvaiko et Euzenat (2005) proposent une définition complète de l'alignement d'ontologies. Cette définition introduit des éléments complémentaires pour la méthode d'alignement. Ces derniers incluent éventuellement un alignement initial, des ensembles de seuils et de pondérations, et des ressources externes. L'alignement initial (ou d'entrée) est exploité par la méthode d'alignement proposée dans l'objectif de l'enrichir. Les ensembles de seuils et de pondérations sont des paramètres utilisés par la méthode au cours du processus d'alignement. Les ressources externes, telles que les thésaurus et les connaissances du domaine, représentent des appuis sémantiques pour la production de l'alignement.

Définition 1 *(Alignement). Une procédure d'alignement est une fonction notée f. Cette fonction prend en entrée deux ontologies (O et $O^{'}$), un alignement initial (A), un ensemble de paramètres (p) et un ensemble de ressources (r). La fonction f produit un alignement ($A^{'}$) comme résultat. Formellement, la fonction f est définie comme suit : $A^{'} = f(O, O^{'}, A, p, r)$ ou plus simplement $A^{'} = f(O, O^{'})$ lorsque le contexte est fixe, et l'alignement initial, les paramètres et les ressources sont omis (Shvaiko et Euzenat (2005)).*

La figure 2.1 est une représentation schématique du processus d'alignement. L'alignement produit est un jeu de paires d'entités $(e, e^{'})$, appartenant, respectivement, aux deux

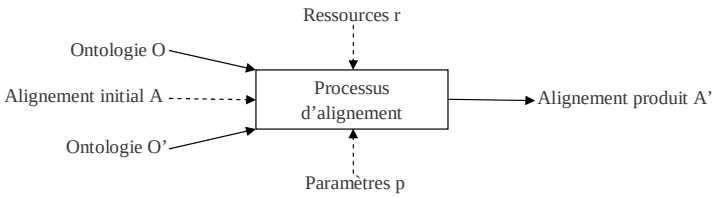


Figure 2.1 — Schéma général d'un processus d'alignement d'ontologies

ontologies O et O' respectivement, possédant une relation re et un seuil de confiance n (Euzenat (2007)). Étant donné deux ontologies O et O', un alignement entre O et O' est un jeu de correspondances, (*i.e.*, un quadruplet) : $< e, e', re, n >$, avec $e \in O$ et $e' \in O'$, re étant une relation liant e et e', alors que n représente le niveau de confiance en cette relation (Euzenat (2007)). Les correspondances sont considérées sous forme d'un triplet $< e, e', re >$. Les relations qui permettent de définir les correspondances sont des mesures de similarité.

2.2.2 Mesures de similarité

La similarité est exploitée dans plusieurs domaines tels que la psychologie et les mathématiques (Euzenat et Shvaiko (2007)). Dans le domaine de la psychologie sociale, la similarité se rapporte à la manière dont les attitudes, les valeurs, les intérêts et la personnalité correspondent entre les personnes. Dans le domaine des mathématiques, plusieurs relations d'équivalence sont appelées *similarités*. Ces relations sont exploitées en géométrie, en algèbre linéaire et en topologie.

La similarité dans le cadre de l'alignement d'ontologies est la similarité sémantique, appelée aussi *proximité sémantique* (Bach (2006)). Cette similarité est calculée entre des documents, ou des termes ou encore des entités, grâce à une métrique basée sur la similitude de leurs significations ou de leur contenu sémantique. La similarité est la quantité qui reflète la force du rapport entre deux objets ou deux caractéristiques. Concrètement, la similarité peut être réalisée en définissant une similarité typologique, ou en employant des ontologies pour définir une distance entre les termes, ou en employant des moyens statistiques pour la corrélation des mots et des contextes textuels (Bach (2006)).

Dans la plupart des approches d'alignement d'ontologies, la similarité sémantique est considérée comme celle de la similarité typologique en mathématiques, où une valeur d'une fonction lui est associée, appelée *fonction de similarité*. La définition de cette fonction de la similarité peut changer selon les approches, selon les propriétés souhaitées. La valeur

de cette fonction est souvent normalisée, c'est à dire comprise entre 0 et 1, ce qui offre des possibilités d'interprétation probabiliste de la similarité. Les propriétés communes de ces fonctions sont la positivité, l'auto-similarité ou la maximalité, la symétrie ou la réflexivité. La similarité possède des caractéristiques supplémentaires telles que la finitude ou la transitivité (Bach (2006)).

2.2.2.1 Mesure de similarité

Une mesure de similarité, notée σ, permet de mesurer le degré de ressemblance entre deux entités (Euzenat *et al.* (2004)). Soit E un ensemble d'entités, la similarité entre les paires d'entités de cet ensemble est définie par la fonction $\sigma : E \times E \longrightarrow \Re$ tel que $\forall x, y, z \in E$

$$\begin{cases} \sigma(x,y) \geq 0 & \textit{(positivité)} \\ \sigma(x,x) \geq \sigma(y,z) & \textit{(maximalité)} \\ \sigma(x,y) = \sigma(y,x) & \textit{(symétrie)} \\ \sigma(x,y) \leq \infty & \textit{(finitude).} \end{cases}$$

Par ailleurs, il existe une autre mesure complémentaire à la mesure de similarité à savoir la dissimilarité.

2.2.2.2 Mesure de dissimilarité

Une mesure de dissimilarité, notée δ, permet de mesurer le degré de différence entre deux entités (Euzenat *et al.* (2004)). Soit E un ensemble d'entités, la similarité entre les pairs d'entités de cet ensemble est définie par la fonction $\delta : E \times E \longrightarrow \Re$ tel que $\forall x, y, z \in E$

$$\begin{cases} \delta(x,y) \geq 0 & \textit{(positivité)} \\ \delta(x,x) = 0 & \textit{(minimalité)} \\ \delta(x,y) = \delta(y,x) & \textit{(symétrie)} \\ \delta(x,y) = \delta(y,z) \Longrightarrow \delta(x,y) = \delta(x,z) & \textit{(transitivité)} \\ \delta(x,y) \leq \infty & \textit{(finitude).} \end{cases}$$

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

2.2.2.3 Mesure de distance

La distance mesure la dissimilarité entre deux entités. Elle est l'inverse de la similarité. En effet, la distance entre deux entités est petite lorsque la valeur de la fonction de la similarité de deux entités est élevée et vice-versa. La distance (ou métrique), $\delta : E \times E \longrightarrow$

$\Re$, est une fonction de dissimilarité respectant la définitivité et l'inégalité triangulaire telles que $\forall x, y, z \in E$ (Euzenat et Shvaiko (2007)) :

$$\begin{cases} \delta(x,y) = 0 \Longleftrightarrow x = y & \textit{(définitivité)} \\ \delta(x,y) + \delta(y,z) \geq \delta(x,z) & \textit{(inégalité triangulaire).} \end{cases}$$

Notons que les valeurs de similarité et de dissimilarité sont souvent des mesures normalisées.

2.2.2.4 Normalisation

La normalisation permet de combiner ces mesures dans des formules et d'obtenir d'autres mesures agrégées qui sont à leurs tours normalisées et peuvent être comparées. La similarité et la dissimilarité normalisée sont notées respectivement $\bar{\sigma}$ et $\bar{\delta}$. Ces mesures vérifient la règle de la complémentarité, $\bar{\sigma} + \bar{\delta} = 1$. Une mesure de similarité est dite *normalisée*, si les valeurs calculées par cette mesure ne peuvent varier que dans un intervalle de 0 à 1.

Les mesures de similarité permettent de calculer des valeurs de similarité entre les différentes entités des ontologies. Ces mesures sont utilisées en fonction de la nature et des caractéristiques des entités ontologiques à aligner. La section suivante va mettre en exergue les principales caractéristiques externes des méthodes d'alignement.

2.3 Les caractéristiques externes

Les méthodes d'alignement peuvent être caractérisées par des caractéristiques internes (de fonctionnement) et des caractéristiques externes. Les caractéristiques internes permettent de distinguer différentes familles de méthodes tandis que les caractéristiques externes concernent les entrées, les sorties et les différents paramètres exploités par les méthodes d'alignement.

2.3.1 Les entrées

Shvaiko et Euzenat (2005) s'intéressent aux données d'entrée des méthodes d'alignement. En effet, les ontologies à aligner peuvent être décrites à travers de nombreux formalismes de représentation de données hiérarchiques. Chaque méthode d'alignement est généralement conçue pour manipuler un certain type de formalismes. Une méthode peut ainsi considérer des schémas de base de données, des ontologies, ou même de simples

structures arborescentes. Les schémas de base de données peuvent être de type relationnel, XML, ou objets. Les ontologies peuvent être décrites en OWL, RDF(S), ou autres langages ontologiques. Les simples structures arborescentes sont les structures des répertoires Web ou encore les structures des répertoires d'un système de fichiers. Les approches d'alignement se distinguent aussi par la prise en charge ou non des hiérarchies peuplées. En effet, certaines méthodes s'appuient sur les données extensionnelles de ces hiérarchies. Ces dernières sont les instances d'une base de données objet ou d'une ontologie, les n-uplets d'une base de données relationnelle, le contenu de documents XML ou encore des fichiers d'un système de fichiers, etc.

2.3.2 Sorties

Les sorties (ou résultats) des méthodes d'alignement peuvent prendre différentes formes. En effet, certaines méthodes peuvent attribuer une mesure de qualité aux couples d'éléments de correspondance trouvées tandis que d'autres produisent uniquement les correspondances. Il existe deux types d'approches d'alignement à savoir les approches symboliques et les approches basées sur des mesures de similarité (Euzenat et Shvaiko (2007)). Le premier type d'approches détermine uniquement l'existence de correspondances, tandis que le second type d'approches attribue une valeur de confiance aux correspondances.

Les méthodes d'alignement se distinguent également par les types de relations qu'elles peuvent déterminer entre les entités alignées. La grande majorité des méthodes s'intéresse seulement à la relation d'équivalence ($\Leftrightarrow$ [1]). Certaines autres méthodes sont plus expressives et distinguent les relations d'implication ($\Rightarrow$ [2], $\Leftarrow$).

Une procédure d'alignement, notée f, peut être symétrique ou asymétrique. Elle est symétrique dans le cas où l'ordre dans lequel les hiérarchies sont considérées n'a pas d'influence sur l'alignement produit. En d'autres termes, si $f(O, O') = f(O', O)$ alors f est dite *symétrique*. Dans le cas contraire, $f(O, O') \neq f(O', O)$ alors f est dite *asymétrique*. Dans le cas où la méthode d'alignement considère la relation d'implication, une méthode asymétrique produira des alignements asymétriques tandis qu'une méthode symétrique produira des alignements symétriques.

Une méthode d'alignement se distingue également par la cardinalité des alignements qu'elle produit. Dans le cadre général, l'alignement entre deux ontologies O et O' possède deux cardinalités : l'une dans le sens O vers O' et l'autre dans le sens inverse de O'

1. notée également $=$.
2. notée également, en fonction du contexte, $\sqsubseteq$ ou $\leq$.

vers O. Cette distinction n'est nécessaire que pour les méthodes prenant en compte des relations directionnelles telles que l'implication. Cependant, comme la grande majorité des méthodes s'intéresse uniquement à l'équivalence, cette distinction n'est pas faite dans la littérature. Principalement, une méthode d'alignement peut produire des alignements :

- fonctionnels (de cardinalité $(0,1)-(0,n)$),
- fonctionnels et injectifs (de cardinalité $(0,1)-(0,1)$),
- de cardinalité $(0,n)-(0,m)$.

La plupart des méthodes d'alignement produisent des alignements au moins fonctionnels. En effet, elles mettent en correspondance chaque élément d'une hiérarchie source à un élément de la hiérarchie cible. Une méthode qui produit des alignements fonctionnels mais pas injectifs sera asymétrique. Cependant, une méthode produisant des alignements fonctionnels et injectifs n'est pas forcément symétrique. La majorité des méthodes d'alignement exploitent des techniques de base pour la production des alignements. Ces techniques de base sont détaillées dans la section suivante.

2.4 Techniques de base d'alignement d'ontologies

Les techniques de base sont aussi appelées *méthodes locales*. Ces techniques regroupent plusieurs méthodes à savoir les méthodes terminologiques, les méthodes structurelles, les méthodes extensionnelles et les méthodes sémantiques (Euzenat et Shvaiko (2007)). Ces différents types de méthodes sont passées en revue dans ce qui suit :

2.4.1 Méthodes terminologiques

Les méthodes terminologiques comparent les chaînes de caractères afin d'en déduire la similarité (ou dissimilarité) en exploitant les relations d'hyperonomie ou d'hyponymie. Certaines méthodes termonologiques se basent sur la comparaison des chaînes de caractères et sont appelées les *méthodes syntaxiques*. Ces méthodes syntaxiques comparent les structures des deux chaînes. Deux chaînes qui partagent des caractères ou des mots en commun seront considérées comme similaires. D'autres méthodes terminologiques qui ont recours à une base de données lexicale, sous forme de réseau sémantique, sont les méthodes linguistiques. Ces méthodes calculent la similarité entre deux chaînes en fonction des relations qu'elles entretiennent.

2.4.1.1 Méthodes syntaxiques

Les méthodes syntaxiques comparent, les termes ou les chaînes de caractères ou bien les textes, des entités à aligner. Ces méthodes permettent de calculer la valeur de la similarité des entités textuelles. Ces entités sont caractérisées par des noms, des étiquettes, des commentaires, des descriptions, etc. Ces méthodes se déclinent en deux sous-catégories. La première sous-catégorie englobe des méthodes qui comparent des termes en se basant sur les caractères contenus dans ces termes. La deuxième sous-catégorie utilise les distances basées sur les tokens (Euzenat et Shvaiko (2007)).

Les méthodes, se basant sur la comparaison de chaînes de caractères, analysent la structure de ces chaînes afin de calculer des mesures de similarité. Ces méthodes exploitent l'ordre des caractères dans la chaîne, le nombre d'apparitions d'une lettre dans une chaîne, etc. Par contre, elles n'exploitent pas la signification des termes.

Dans la littérature, plusieurs mesures assurent le calcul de la valeur de similarité entre deux chaînes de caractères. Ces mesures sont soit des mesures de similarité ou de dissimilarité ou encore des distances. Parmi ces mesures, quelques unes sont très utilisées dans le cadre de l'alignement d'ontologies, telles que la distance de HAMMING, la similarité de JACCARD, la distance d'édition, la distance de LEVENSHTEIN, etc. Dans la suite, une chaîne de caractères, s, est considérée comme un ensemble de caractères, noté S.

La distance de HAMMING permet la comparaison de deux chaînes de caractères. Elle comptabilise le nombre de positions dans lesquelles les deux chaînes de caractères sont différentes (Hamming (1950)).

Définition 2 *(Distance de* HAMMING*). La distance de* HAMMING *est une mesure de dissimilarité* $\overline{\delta}_{\text{HAMMING}} : S \times S \rightarrow [0,1]$ *tel que* $\overline{\delta}_{\text{HAMMING}}(s,s') = \frac{(\sum_{i=1}^{min(|s|,|s'|)} s[i] \neq s'[i]) + ||s| - |s'||}{max(|s|,|s'|)}$.

Définition 3 *(Similarité de* JACCARD*). Étand donné deux chaînes de caractères* s *et* $s^{'}$*. Les deux ensembles* S *et* $S^{'}$ *représentent, respectivement, les caractères de* s *et* $s^{'}$*. La similarité de* JACCARD *est une fonction* $\overline{\sigma}_{\text{JACCARD}} : S \times S \rightarrow [0,1]$ *tel que* $\overline{\sigma}_{\text{JACCARD}}(s,s^{'}) = \frac{|S \cap S^{'}|}{|S \cup S^{'}|}$.

La distance d'édition (*edit distance*) appartient à la classe des fonctions permettant de mesurer la distance entre deux chaînes de caractères. Dans ce groupe, la distance représente le coût de la meilleure séquence des opérations d'édition convertissant la chaîne s en chaîne $s^{'}$. Les opérations d'édition sont l'insertion, la suppression et la substitution de caractères. À chaque opération d'édition est attribuée un poids.

Définition 4 *(Distance d'édition). Étant donné un ensemble Op d'opérations d'édition (op : $S \rightarrow S$) et une fonction de coût $w : Op \rightarrow \mathbb{R}$, tel que pour chaque paire de chaînes de caractères il existe une séquence d'opérations d'édition permettant la transformation de la première chaîne en la seconde chaîne (et vice versa). La distance d'édition est une fonction de dissimilarité $\overline{\delta}_{de} : S \times S \rightarrow [0,1]$, avec $\overline{\delta}_{de}(s,s')$ est le coût minimal de la séquence d'opérations d'édition transformant s en s' telle que $\overline{\delta}_{de}(s,s') = min_{(op_i);op_n(...op_1(s))=s'}(\sum_{i \in I} w_{op_i})$.*

La distance de LEVENSHTEIN est une mesure avec un nombre minimal d'opérations d'insertions, de suppressions et de substitutions de caractères nécessaires pour la transformation d'une chaîne en une autre (Levenshtein (1966)). La distance de LEVENSHTEIN est une distance d'édition où tous les poids sont égaux à 1.

La mesure de JARO est définie pour l'alignement des noms contenant des erreurs d'orthographe (Jaro (1989)). Elle ne présente pas une distance d'édition mais elle se base sur le nombre et l'ordre des caractères communs entre deux chaînes de caractères. Cette mesure n'est pas une mesure de similarité car elle n'est pas symétrique.

Définition 5 *(Mesure de JARO). La mesure de JARO est définie par la fonction $\overline{\sigma}_{\text{JARO}} : S \times S$ tel que $\overline{\sigma}_{\text{JARO}}(s,s') = \frac{1}{3} \times (\frac{|com(s,s')|}{|s|} + \frac{|com(s,s')|}{|s'|} + \frac{|com(s,s')|-|transp(s,s')|}{|com(s,s')|})$ avec $s[i] \in com(s,s')$ si et seulement si $\exists j \in [i - min(|s|,|s'|)/2i + min(|s|,|s'|)/2]$ et $transp(s,s')$ sont les éléments de $com(s,s')$ qui apparaissent dans des ordres différents dans s et s'.*

La distance de JARO-WINKLER est une amélioration de la mesure de JARO. La distance de JARO-WINKLER favorise les correspondances entre les chaînes avec des préfixes communs (Winkler (1999)).

Définition 6 *(Mesure de JARO-WINKLER). La mesure de JARO-WINKLER $\overline{\sigma}_{\text{JARO-WINKLER}} : S \times S$ tel que $\overline{\sigma}_{\text{JARO-WINKLER}}(s,s') = \overline{\sigma}_{\text{JARO}}(s,s') + P \times Q \times \frac{(1-\overline{\sigma}_{\text{JARO}}(s,s'))}{10}$ avec P la longueur du préfixe communs et Q une constante.*

Les mesures de similarité ou de dissimilarité présentées ci-dessus sont plus appropriées lors de la comparaison de deux chaînes de caractères courtes. Dans le cas où les chaînes de caractères à comparer sont représentées par des textes longs ou des documents textuels, ces chaînes sont découpées en plusieurs morceaux, appelés *tokens*. Elles deviennent des ensembles des *tokens*, et la similarité entre les chaînes est mesurée grâce aux mesures de similarité basées sur les tokens.

La littérature comprend plusieurs mesures basées sur les tokens. Ces mesures comprennent par exemple la similarité de DANG, la distance de JENSEN-SHANNON, la distance de FELLEGI-SUNTER, etc. Les mesures à base des tokens sont les plus utilisées par les méthodes d'alignement et sont présentées dans ce qui suit.

La similarité de JACCARD (Définition 3) peut être utilisée pour la comparaison des ensembles de tokens. Elle définit la similarité comme le rapport entre la cardinalité de l'intersection des ensembles sur la cardinalité de leur union.

Le calcul de la similarité entre les commentaires des entités à aligner est réalisé par la mesure Q-GRAM (Ukkonen (1992)). Elle est exploitée pour la comparaison des chaînes de caractères constituées de plusieurs mots. Cette mesure calcule le nombre de *qgrams* en commun, c-à-d., les séquences de n caractères figurant à la fois dans deux chaînes de caractères.

Définition 7 *(Mesure de* Q-GRAM*). La mesure de* Q-GRAM $\sigma_{\text{Q-GRAM}} : S \times S$ *tel que* $\sigma_{\text{Q-GRAM}}(s, s') = \frac{|qgram(s,n) \cap qgram(s',n)|}{min(|s|,|s'|)-n+1}$ *où* n *est la longueur d'un qgram.*

La distance de MONGE-ELKAN est une approche hybride. Elle permet de calculer la similarité entre deux chaînes de manière récursive (Monge et Elkan (1996a)). Elle analyse les sous-chaînes des chaînes initiales.

Définition 8 *(Distance de* MONGE-ELKAN*). La distance de* MONGE-ELKAN *estime la similarité entre deux chaînes* s *et* s'*. Les deux chaînes sont représentées par leurs sous chaînes respectives* $s = a_1 \dots a_k$ *et* $s' = b_1 \dots b_l$*. La mesure de* MONGE-ELKAN $\delta_{\text{MONGE-ELKAN}} : S \times S$ *tel que* $\delta_{\text{MONGE-ELKAN}}(s, s') = \frac{1}{|s|}(\sum_{i=1}^{|s|} max_{j=1}^{|s'|}(sim(a_i, b_j)))$*. La* $sim(a_i, b_j)$ *est la similarité entre les sous chaînes* a_i *et* b_j*. Cette similarité est calculée par l'intermédiaire d'une des mesures de similarité, telles que* JACCARD, JARO, JARO-WINKLER, *etc.*

2.4.1.2 Méthodes linguistiques

La similarité entre deux entités représentées par des termes peut aussi être déduite en les analysant à l'aide des méthodes linguistiques. Les méthodes linguistiques permettent de déterminer la similarité entre deux entités. Ces entités sont représentées par des termes (ou mots). Ces méthodes syntaxiques prennent en charge les propriétés expressives et productives du langage naturel qui peuvent être intrinsèques ou extrinsèques (Euzenat et Shvaiko (2007)).

Les informations intrinsèques sont des propriétés linguistiques internes des termes, telles que des propriétés morphologiques ou syntaxiques. Un même concept (ou entité) peut être décrit par plusieurs termes (synonymie) ou par plusieurs variantes d'un même terme. Les méthodes intrinsèques cherchent la forme canonique ou représentative d'un mot ou d'un terme (lemme) en exploitant ses variantes linguistiques (lexème). La similarité entre deux termes est mesurée en comparant leurs lemmes. La recherche du lemme d'un mot peut être effectuée à l'aide d'un dictionnaire. Une approche automatique utilise les *stemmers*. Le stemmer est un algorithme qui permet de déterminer la forme radicale d'un terme. Cette forme est déduite à partir d'une forme infléchie ou dérivée d'un mot donné. Les radicaux trouvés par les stemmers n'ont pas besoin d'être identiques à la racine morphologique du mot. Il suffit que les mots similaires soient associés à un même radical, même si ce radical n'est pas une racine de mot valide. La *lemmatisation* est une approche qui détermine le radical exact d'un mot. Ce processus extrait la catégorie lexicologique d'un mot et applique des règles de normalisation différentes pour chaque partie du discours. Cette approche nécessite la connaissance de la grammaire ainsi que les différentes règles d'un langage.

Les informations extrinsèques exploitent des ressources externes telles que des dictionnaires ou des vocabulaires. Ces méthodes calculent la valeur de similarité entre deux termes en employant des ressources externes. Ces ressources regroupent les dictionnaires, les lexiques ou les vocabulaires. La similarité entre deux termes est calculée en exploitant les liens sémantiques existants dans ces ressources externes. Ces liens regroupent les synonymes (pour l'équivalence), des liens d'hyponymes ou d'hyperonymes (pour la subsomption). Typiquement, l'API WordNet [3] (Miller (1995)) est un système lexicologique, qui a été employé pour chercher des relations telles que la synonymie entre les termes. WordNet est aussi exploité pour calculer la distance sémantique entre les termes. Les ressources externes utilisées dans les méthodes extrinsèques peuvent aussi être des vocabulaires ou des dictionnaires multi-lingues, ou d'autres systèmes tels que EuroWordNet [4] et Polylex [5].

3. http ://wordnet.princeton.edu/.

4. http ://www.illc.uva.nl/EuroWordNet/ - un système de réseaux sémantiques pour des langues européennes où chaque langue développe son propre WordNet et elles sont reliées entre elles par des liens inter-langues.

5. http ://www.informatics.susx.ac.uk/research/nlp/polylex/ - un lexique multilingue pour le Néerlandais, l'Anglais et l'Allemand, construit à partir de différents lexiques monolingues contenus dans la base de données CELEX (http ://www.ru.nl/celex/).

2.4.2 Méthodes structurelles

Les méthodes structurelles déterminent la similarité entre deux entités en fonction des informations structurelles. En effet, les entités sont reliées entre elles par des liens sémantiques ou syntaxiques. Ces liens forment ainsi une hiérarchie ou un graphe d'entités. Les méthodes structurelles se subdivisent en deux familles : les méthodes structurelles internes et les méthodes structurelles externes. La première famille exploite uniquement les informations décrivant les attributs d'entités, tandis que la seconde famille considère des relations entre ces entités.

2.4.2.1 Méthodes structurelles internes

Les méthodes structurelles internes sont aussi nommées les *approches basées sur les contraintes* (Rahm et Bernstein (2001)). Ces méthodes utilisent les informations contenues dans les structures internes des entités pour le calcul de la similarité. Ces informations regroupent les attributs des entités. Ces attributs sont les informations du co-domaine, de la cardinalité des attributs, des caractéristiques des attributs (la transitivité, la symétrie, etc.), et éventuellement des restrictions sur les attributs. Plusieurs de ces méthodes internes sont proposées pour le calcul de la similarité entre deux schémas de bases de données. Ces méthodes exploitent la similarité entre les contraintes à propos des éléments des deux structures. Valtchev (1999) a introduit la similarité entre deux types de données. Cette mesure est le rapport entre la différence des tailles des types et la taille de leur généralisation commune. La taille d'un type est définie comme la cardinalité de l'ensemble de valeurs qu'il propose. La taille de la généralisation commune dépend des types des éléments (un ensemble pour les types énumérés, un intervalle pour les types ordonnés, etc.).

2.4.2.2 Méthodes structurelles externes

Les méthodes structurelles externes exploitent les relations existantes entre les entités elles-mêmes. Ces relations contiennent des relations de subsomption (spécialisation ou *is a*) ou de méréologie (*part-whole*). En plus de ces relations, les entités sont considérées dans des structures hiérarchiques. La similarité entre les entités est déterminée en fonction de leurs positions dans leurs hiérarchies. Deux entités sont considérées similaires alors leurs voisins pourraient être à leur tour similaires. Si deux entités peuvent être considérées similaires dans les cas où (Euzenat et Shvaiko (2007)) :

- Leurs super-entités directes (ou toutes leurs super-entités) sont similaires ;
- Leurs soeurs (ou toutes leurs soeurs, qui sont les entités ayant la même super-entité directe avec les entités en question) sont déjà similaires ;
- Leurs sous-entités directes (ou toutes leurs sous-entités) sont déjà similaires ;
- Leurs descendants (entités dans le sous-arbre ayant pour racine l'entité en question) sont déjà similaires ;
- Toutes (ou presque toutes) leurs feuilles (les entités de même type, qui n'ont aucune sous-entité, dans le sous-arbre ayant pour racine l'entité en question) sont déjà similaires ;
- Toutes (ou presque toutes) les entités dans les chemins de la racine aux entités en question sont déjà similaires.

Des combinaisons de ces différents cas sont aussi possibles. Les approches externes rencontrent des difficultés dans le cas où les hiérarchies à aligner possèdent des niveaux différents de granularité.

2.4.3 Méthodes extensionnelles

Les méthodes extensionnelles calculent la similarité entre deux entités en fonction de leurs extensions (Euzenat et Shvaiko (2007)). Les extensions représentent les ensembles des instances de deux entités. Dans le cas où l'intersection des ensembles des instances est grande, les mesures employant des opérations sur les ensembles peut être exploitées. Ces mesures comprennent la distance de HAMMING et la distance de JACCARD. Ces mesures peuvent être adaptées pour développer des mesures extensionnelles. Ces mesures produisent la similarité de deux entités en fonction de la similarité entre les deux ensembles de leurs instances. Ces mesures se basent sur la comparaison exacte des éléments existant dans les deux ensembles. Dans le cas où les ensembles des instances ne partagent aucune partie commune, ces mesures ne sont plus applicables (le résultat produit sera toujours égal à 1, ce qui signifie que les entités à comparer sont toujours différentes). Valtchev (1999) a proposé la similarité basée sur des correspondances (*match-based similarity*). La similarité entre deux ensembles est la similarité moyenne des éléments dans l'ensemble des correspondances, noté PARING. Ce dernier possède la somme maximale de toutes les similarités des paires dans l'ensemble. Le calcul de l'ensemble PARING(S, S') est un problème d'optimisation qui maximise le total des similarités des paires des éléments de S et S'.

Définition 9 *(Similarité basée sur des correspondances). Étant donné une fonction de*

similarité $\sigma : E \times E \rightarrow \mathbb{R}$*, la similarité de correspondances entre deux sous-ensembles de* E*, où* E *est un ensemble d'entités, est une fonction de similarité* $MSim : 2^E \times 2^E \rightarrow \mathbb{R}$ *tel que* $\forall x, y \subseteq E$*,* $Msim(x, y) = \frac{max_{p \in \text{PARING}(x,y)}(\sum_{(n,n')\in p} \sigma(n,n'))}{max(|x|,|y|)}$*, avec* PARING$(x, y)$ *est un ensemble de correspondances d'éléments de* x *vers des éléments* y*.*

2.4.4 Méthodes sémantiques

Les méthodes sémantiques se basent principalement sur deux approches. La première approche repose sur les modèles de la logique tandis que la deuxième approche regroupe les méthodes de déduction afin de déduire la similarité entre deux entités. Les approches logiques sont la satisfiabilité propositionnelle (SAT), la SAT modale ou les logiques de descriptions. Giunchiglia *et al.* (2003) et Bouquet *et al.* (2003) respectivement, emploient des techniques issues de la satisfiabilité propositionnelle (SAT). Ces techniques permettent la vérification de la validité d'un ensemble de formules propositionnelles. Ce dernier est construit en traduisant des relations déjà connues et des relations à vérifier entre des entités vers des formules propositionnelles. Giunchiglia *et al.* (2003) étendent les méthodes proposées vers le modèle de la SAT modale, qui peut aussi contenir des prédicats binaires. Le premier modèle n'accepte que des prédicats unaires qui sont des entités comme des concepts ou des classes. Le second modèle permet de calculer en plus des attributs ou des propriétés (*slots*). Il emploie des opérateurs de la logique modale. La validité de l'ensemble de formules en logique modale est aussi vérifiée en utilisant des procédures de recherche de la satisfiabilité (SAT). Dans le cas où la validité est satisfaite, les relations hypothétiques entre des entités, qui sont des traductions de la requête sur la relation entre ces entités en logique modale, sont confirmées. Les techniques des logiques de description (le test de subsomption) peuvent être employées. Elles permettent de vérifier les relations sémantiques entre les entités telles que l'équivalence (la similarité est égale à 1), la subsomption (la similarité est comprise entre 0 et 1) ou l'exclusion (la similarité est égale à 0). Elles assurent aussi la déduction de la similarité de deux entités.

2.5 Stratégies d'alignement

Les techniques de base présentées dans la section 2.4 représentent les fondements des stratégies d'alignement appelées aussi les *méthodes globales*. L'alignement est déduit une fois que la similarité (ou la dissimilarité) entre les entités est calculée. La déduction de l'alignement nécessite des traitements globaux. Euzenat et Shvaiko (2007) évoquent plusieurs

aspects de construction d'alignement à savoir : composition d'alignement, agrégation de similarité, calcul global de similarité, méthodes d'apprentissage et méthodes probabilistiques.

2.5.1 Composition d'alignement

La composition d'alignement permet la combinaison des résultats de deux alignements afin de produire un nouveau alignement. Dans la littérature, deux types de composition sont recensés à savoir la composition séquentielle et la composition parallèle.

La composition séquentielle (ou linéaire) permet d'exécuter successivement les algorithmes d'alignement. Le résultat d'un algorithme sert comme entrée à l'algorithme suivant. Le processus de composition peut être répété plusieurs fois jusqu'à l'obtention du résultat final. La composition linéaire est exploitée dans deux types d'algorithmes : les algorithmes de découverte d'alignement et les algorithmes de filtrage d'alignement. Les algorithmes de découverte sont généralement utilisés quand les algorithmes suivent une stratégie structurelle d'alignement. Les algorithmes de filtrage d'alignement permettent la réduction du nombre de paires d'un alignement en fonction de plusieurs critères.

La composition parallèle permet de combiner plusieurs résultats produits par une panoplie d'algorithmes individuels. Ces algorithmes sont exécutés de manière totalement indépendante (Rahm et Bernstein (2001)). La composition parallèle comporte deux types d'approches : les algorithmes ne permettant pas la sélection d'éléments de correspondance pertinents et les algorithmes le permettant. Le premier type d'algorithme calcule une valeur de qualité pour chaque appariement possible. En effet, une mesure de qualité est proposée pour chaque couple d'entités et pour chaque relation étudiée. Dans ce type d'approches, la combinaison des résultats consiste en l'agrégation des fonctions de qualité produites par chaque algorithme en une fonction globale. Par la suite, l'approche sélectionne un sous-ensemble d'éléments de correspondance pertinentes. Le second type d'approche fournit à son tour une fonction de qualité. Cette fonction est la combinaison des résultats. Elle est réalisée par l'intermédiaire des opérations ensemblistes (union ou intersection) effectuées sur les ensembles de correspondances produites par chaque algorithme d'alignement. De même, ce type d'approches agrège les fonctions de qualité lorsqu'elles sont définies sur des éléments de correspondance produits par plusieurs algorithmes. Ce type d'approches se base sur la combinaison ensembliste des résultats intermédiaires.

2.5.2 Agrégation de similarité

L'agrégation de similarité permet de calculer la similarité composée. Chaque objet structuré (classe, individu, etc.) est évoqué dans plusieurs relations. Il est possible de calculer plusieurs similarités entre les objets des entités de chaque ontologie. Ces similarités doivent être agrégées dans le but de déterminer une similarité entre les entités. Le calcul de la similarité entre deux classes nécessite une agrégation, sous forme d'une seule similarité, des similarités obtenues à partir de leurs noms, des super-classes, des instances et de leurs propriétés respectives.

2.5.3 Calcul global de similarité

Le calcul de la similarité globale se réalise en prenant en considération les nœuds voisins. La similarité peut évoquer les ontologies en entier et la valeur finale de la similarité peut dépendre de toutes les entités appartenant aux ontologies à aligner. En plus, les distances définies par les méthodes locales peuvent être définies d'une manière circulaire quand les ontologies ne sont pas réduites à des graphes acycliques. En effet, la distance entre deux classes dépend des distances entre leurs instances. Les instances peuvent être dépendantes à leur tour des distances entre leurs classes. Les circuits au niveau d'une ontologie peuvent aussi engender des dépendances circulaires (Valtchev (1999)). Le calcul de la similarité dans le cas de la dépendance circulaire ne se fait pas d'une manière locale. La résolution de ce problème se réalise par l'intermédiaire d'un calcul itératif de la distance ou de la similarité en se référant au niveau de chaque étape à la valeur de l'étape précédente.

2.5.4 Méthodes d'apprentissage

Les méthodes d'apprentissage se basent sur le fait d'apprendre à partir des alignements corrects, exemples positifs, et des alignements incorrects, exemples négatifs (Euzenat et Shvaiko (2007)). Les méthodes s'appuyant sur l'apprentissage automatique opèrent généralement en deux phases : une phase d'apprentissage et une phase de classification ou d'alignement. Au cours de la première phase, les données utilisées pour l'apprentissage sont générées par un alignement manuel. Ces données sont exploitées par la méthode d'alignement pour apprendre à réaliser l'alignement. Au cours de la seconde phase, les alignements appris sont utilisés pour l'alignement de nouvelles ontologies. L'alignement produit peut être introduit par la suite dans le jeu des alignements d'apprentissage pour l'enrichir.

2.5.5 Méthodes probabilistes

Les méthodes probabilistes peuvent être utilisées, d'une manière similaire aux méthodes d'apprentissage, pour l'alignement d'ontologies (Euzenat et Shvaiko (2007)). Elles sont exploitées pour la combinaison des aligneurs. Les réseaux bayesiens constituent un exemple typique des méthodes probabilistes.

Les réseaux bayésiens constituent une approche probabilistique pour la modélisation des causes et des effets. Les réseaux bayésiens sont composés d'un graphe orienté acyclique et d'un ensemble de tables des probabilités conditionnelles. Le graphe contient des nœuds (ou des variables) et des arcs. Les arcs reliant les nœuds assurent des dépendances conditionnelles et indiquent la direction d'influence. Par exemple, un arc d'un nœud X_1 (nœud parent) à un nœud X_2 (nœud fils) signifie que X_1 possède une influence directe sur X_2. Les influences d'un nœud à un autre nœud (basée sur les anciennes expériences) sont définies par les tables de probabilités conditionnelles pour les nœuds.

Etant donné certaines valeurs pour quelques nœuds, il est possible d'inférer des valeurs de distributions de probabilités d'autres nœuds. Un réseau bayésien peut être spécifié par un expert et par la suite peut être exploité pour inférer des nouvelles valeurs dans le but de développer des prédictions ou des diagnostics de causes.

Les réseaux bayésiens sont modélisés et utilisés dans différentes méthodes d'alignement d'ontologies. Par exemple, Pan *et al.* (2005) traduisent les deux ontologies sous la forme de deux réseaux bayésiens et l'alignement est réalisé comme des inférences évidentes entre les deux réseaux. Mitra *et al.* (2005) exploitent les réseaux bayésiens pour l'amélioration des alignements existants, comme par exemple pour dériver des alignements oubliés.

2.6 Étude comparative des méthodes d'alignement

La comparaison des méthodes d'alignement peut se faire sur plusieurs volets. Le premier volet permet de les comparer en se basant sur leurs caractéristiques externes. Ces caractéristiques regroupent principalement les entrées et les sorties des méthodes. Le second volet étudie les techniques de base utilisées par les méthodes d'alignement pour le calcul de la similarité. Les méthodes de similarité utilisées renseignent sur les éléments exploités au niveau de l'ontologie afin de produire l'alignement. Le troisième volet indique les stratégies d'alignement exploitées par les méthodes d'alignement. Ces stratégies décrivent éventuellement la démarche exploitée par la méthode pour déterminer les correspondances entre les différents constituants des deux ontologies à aligner.

2.6.1 Comparaison des caractéristiques externes

La table 2.1 regroupe les principales méthodes d'alignement citées dans la littérature.

Méthode	Types d'ontologies	Relation	Cardinalité
ANCHOR-PROMPT (Noy et Musen (2001))	RDF(S), OWL	⇔	(0,n)-(0,n)
ASCO1 (Bach *et al.* (2004))	RDF(S)	⇔	(0,n)-(0,n)
AUTOMS (Kortis *et al.* (2006))	RDF(S), OWL	⇔	
CAIMAN (Lacher et Groh (2001))	HT	⇔	(0,1)-(0,1)
COMA (Do et Rahm (2002))	SR, XML	⇔	(0,1)-(0,1), (0,1)-(0,n), (0,n)-(0,n)
CUPID (Madhavan *et al.* (2001))	XML, SOO	⇔	(0,n)-(0,n)
GLUE (Doan *et al.* (2002))	HT	⇔	
GMO (Hu *et al.* (2005))	RDF(S), OWL	⇔	
V-DOC (Qu *et al.* (2006))	RDF(S), OWL	⇔	
H-MATCH (Castano *et al.* (2006))	RDF(S), OWL	⇔	0,1)-(0,1), (0,1)-(0,n)
OLA (Euzenat et Valtchev (2004))	RDF(S), OWL	⇔	(0,1)-(0,1)
OMAP (Straccia et Troncy (2005))	HT	⇔	(0,1)-(0,n)
OPLMAP (Nottelmann et Straccia (2006))	RDF(S), OWL	⇔	(0,n)-(0,1)
QOM (Ehrig et Staab (2004))	HT, RDF(S), OWL	⇔	(0,1)-(0,1)
RIMOM (Tang *et al.* (2006))	HT	⇔	(0,1)-(0,1), (0,1)-(0,n)
SBI-HICAL (Ichise *et al.* (2002))	HT, RDF(S), OWL	⇔	(0,n)-(0,n)
SCM (Hoshiai *et al.* (2004))	SR, SOO, XML, RDF	⇔	(0,1)-(0,n)
SF (Melnik *et al.* (2002))	HT, RDF(S), OWL	⇔	(0,1)-(0,1)
S-MATCH, CTX-MATCH (Giunchiglia *et al.* (2003))	HT, RDF(S), OWL	⇔, ⇒	

Tableau 2.1: Caractéristiques externes des principales méthodes d'alignement

Elle décrit pour chaque méthode ses caractéristiques générales, c-a-d., le type de l'ontolo-

gie à aligner, le niveau des relations pris en considération et éventuellement les cardinalités des alignements produits.

Dans la littérature, plusieurs méthodes d'alignement ont été proposées. Ces méthodes permettent l'alignement des hiérarchies textuelles (notées HT) et des ontologies. Les ontologies, à aligner, peuvent être décrites avec plusieurs langages de description. Ces langages regroupent le RDF(S) ou les différentes déclinaisons du OWL (OWL-Lite et OWL-DL). D'autres méthodes d'alignement assurent l'alignement des structures. Ces structures décrivent des schémas de bases de données (schémas relationnels, notés SR, et les schémas orientés objets, notés SOO) ou même des documents XML. En effet, les méthodes CUPID et SIMILARITY FLOODING (notée SF dans la table 2.1) sont considérées comme des méthodes d'alignement d'ontologies. Elles sont aussi considérées comme des méthodes représentatives des approches structurelles d'alignement d'ontologies. La méthode COMA a été développée pour obtenir un maximum de flexibilité. En effet, elle est adaptée à de nombreux types de schémas.

Au niveau des relations détectées dans les ontologies, la quasi-totalité des méthodes d'alignement se limitent à la relation d'équivalence ($\Leftrightarrow$). La méthode S-MATCH est l'unique méthode prenant en considération la relation d'équivalence ainsi que la relation d'implication ($\Rightarrow$).

Les méthodes d'alignement d'ontologies proposent généralement un seul type de cardinalités (dans la majorité des cas une cardinalité de type (0,1)-(0,1)). Cependant, la méthode COMA offre encore une grande flexibilité au niveau de la cardinalité en autorisant les quatre types de cardinalités. D'autres méthodes d'alignement telles que ANCHOR-PROMPT, ASCO1, CUPID et SBI-HICAL proposent aussi une cardinalité de type (0,n)-(0,n).

La table 2.2 illustre, pour chaque méthode le type de composition utilisée, la technique de sélection exploitée et éventuellement les post-traitements effectués. La composition permet la combinaison de plusieurs techniques d'alignement ou plusieurs niveaux d'information. La technique de sélection regroupe le seuillage ou l'optimisation (maximisation ou minimisation) locale de l'appariement.

La quasi majorité des méthodes d'alignement combinent les différentes techniques. Cette composition est réalisée de manière parallèle par l'intermédiaire d'une combinaison statistique des résultats. La plupart des méthodes exploitent des moyennes pondérées afin de réaliser la combinaison. Quelques autres méthodes s'appuient sur la fonction de pondération sigmoïde dans le but d'attribuer les poids les plus forts pour les meilleures valeurs

et d'affecter les pondérations faibles pour les petites valeurs. Les méthodes d'alignement d'ontologies reposant seulement sur des approches structurelles (voir table 2.3) utilisent généralement une composition linéaire. La méthode OMAP est basée sur un ordre de préférence entre les différentes techniques utilisées.

Méthode	Composition	Sélection	Post-traitement
ANCHOR-PROMPT	Linéaire (exploite un alignement d'entrée)	Seuil	-
ASCO1	Moyenne pondérée (deux niveaux : terminologique et structurel)	Seuil	-
AUTOMS	-	Maximisation locale	-
CAIMAN	-	Maximisation locale	-
COMA	Statistique (maximum, moyenne, etc.)	Seuil ou maximisation locale	-
CUPID	Moyenne pondérée	Seuil	-
GLUE	Moyenne pondérée	-	Relaxation des labels
H-MATCH	Moyenne pondérée	Seuil, ou maximisation locale	-
OLA	Moyenne pondérée	Seuil, maximisation globale	-
OMAP	Priorités entre méthodes	Maximisation locale	-
OPLMAP	Moyenne pondérée	Seuil	Consistance
QOM	Moyenne pondérée, moyenne sigmoïde	Seuil et maximisation locale	-
RIMOM	Moyenne pondérée, moyenne sigmoïde	Seuil et minimisation locale	Réduction de la cardinalité, consistance
SBI-HICAL	-	Seuil	-
SCM	-	Maximisation locale	Consistance
SF	Linéaire (exploite un alignement d'entrée)	Seuil	Stabilisation

S-MATCH, CTX-MATCH	Union ensembliste	Seuil, résolveur SAT	-

Tableau 2.2: Comparaison des techniques de composition et de sélection des principales méthodes d'alignement

Les méthodes d'alignement reposent, dans la majorité des cas, sur une sélection exploitant un seuil ou sur une optimisation locale de l'appariement. La méthode OLA utilise une approche globale de maximisation de la similarité de l'alignement. La méthode RIMOM fonctionne avec une minimisation locale du risque.

La majorité des méthodes d'alignement n'offrent pas un post-traitement pour leur résultats à l'exception des méthodes RIMOM et GLUE. Le post-traitement le plus utilisé est le filtre de consistance (Nottelmann et Straccia (2005), Hoshiai *et al.* (2004)). La méthode RIMOM est l'unique méthode proposant l'utilisation d'un filtre de réduction de cardinalité. La méthode GLUE exploite une technique de relaxation des labels (*relaxation labelling*). Cette relaxation permet à partir des règles, spécifiques au domaine, d'affiner le résultat de l'alignement produit.

2.6.2 Comparaison des techniques de bases

La table 2.3 présente une étude comparative qui montre les principaux techniques de base utilisées par chaque méthode d'alignement. Ces techniques regroupent les méthodes terminologiques, les méthodes structurelles, les méthodes extensionnelles et les méthodes sémantiques. Parmi les méthodes présentées dans la table 2.3, la majorité sont des méthodes se basant sur les techniques de base. Les méthodes GLUE, V-DOC, OMAP, OPLMAP, QOM et RIMOM utilisent également, en plus des techniques de bases, des stratégies d'alignement.

Méthode	**Techniques terminologiques**		**Techniques structurelles**		
	Syntaxique	*Linguistique*	*Attribut*	*Relation d'ordre*	*Relation transversale*
ANCHOR-PROMPT	-	-	-	-	Similarité des chemins
ASCO1	JARO-WINKLER (identificateurs et labels), TF-IDF (commentaires)	Similarité synatxique (JARO-WINKLER) entre synset WORDNET, tokenisation	Proposition d'entités du voisinage (subsumés, subsumantes, soeurs) qui sont similaires (à partir des similarités terminologiques)	-	-
AUTOMS	-	LSA sur thésaurus WORDNET	-	-	-
COMA	Affixe, distance d'édition, n-gram, soundex	Synonymie (WORDNET), tokenisation	Table de compatibilité	Moyenne sur les subsumés et les feuilles	-
CUPID	-	Basé sur un thésaurus	Table de compatibilité	proportion d'attributs similaire	
GLUE	-	-	-	Prise en compte des subsumants et descendants par la relaxation des labels	

GMO	-	-	Algorithme itératif de propagation de similarités		
V-Doc	-	-	Prise en compte des entités en relation (et des attributs) pour la construction des vecteurs		
H-match	-	Chemin pondéré dans WordNet	Table de compatibilité	-	Similarité contextuelle
OLA	-	Hamming entre synsets WordNet	Système d'équations interdépendantes		
oMAP	Égalité (chaîne entière ou racine)	-	-	Moyenne des similarités des subsumants directs	-
OplMap	Égalité, n-gram, Jaccard	-	-	Similarités binaires sur appariements des subsumants et subsumés directs	-
QOM	Égalité, distance d'édition	-	Égalité des types de données	SimSet sur subsumants, subsumés et soeurs	SimSet sur relations directes
RiMOM	Similarité statistique	Fonction Lin de WordNet, Tagging entités nommées	Table de compatibilité de type	Moyenne des similarités des subsumants et des subsumés directs	Moyennes des similarités du voisinage direct
SBI-Hical	-	Seuil	-	-	-

SCM	-	Maximisation locale	Consistance	-	-
SF	-	-	Propagation des similarités		
S-MATCH, CTX-MATCH	Distance d'édition, n-gram	Prétraitement TAL, relation stricte dans WORDNET (=, ≤)	-	-	-

Tableau 2.3: Comparaison des techniques de combinaison et de sélection des principales méthodes d'alignement

Les méthodes terminologiques regroupent les méthodes syntaxiques et les méthodes linguistiques. Les méthodes syntaxiques incluent souvent les distances d'édition et n-grams. Ces mesures sont parfois appliquées également sur les chemins de noms. Le chemin de noms est la chaîne de caractères issue de la concaténation du nom de l'entité et de ceux de ses subsumantes. Ce principe peut être considéré comme une approche hybride car il prend en compte l'information structurelle sur la relation d'ordre. La méthode RiMOM utilise une mesure de similarité statistique basée sur un corpus textuel.

Les méthodes linguistiques s'appuient généralement sur le thésaurus WordNet. Certaines méthodes tirent profit des relations sémantiques (de manière stricte ou par des mesures sémantiques), d'autres exploitent seulement les synsets (ASCO, OLA). La méthode AUTOMS s'appuie sur une technique d'indexation (LSA) et des relations sémantiques pour déterminer la valeur de la similarité entre deux entités.

Les méthodes structurelles sont exploitées pour déterminer l'alignement. De nombreuses méthodes d'alignement exploitent la relation d'ordre au sein des ontologies. Leur stratégie d'alignement consiste à rechercher de nouvelles relations à partir d'un alignement d'entrée (généralement proposé par une approche syntaxique). Cette stratégie considère que deux entités sont en relation si leurs voisins respectifs sont également en relation. Quelques autres méthodes proposent une approche structurelle générique. Cette dernière confond tous les types de relations, en les traitant séparément les uns des autres. La prise en compte de la structure interne (attribut) est souvent réalisée par le biais d'une table de compatibilité entre les types de données des attributs.

2.6.3 Comparaison des stratégies d'alignement

La table 2.4 donne pour chaque méthode exploitant une stratégie d'alignement recensée, le type de prétraitement qu'elle réalise. La table 2.4 fournit aussi la technique utilisée et la mesure sur laquelle s'appuie la comparaison des extensions. Les méthodes OplMap et OMap n'utilisent pas de mesures de comparaison étant donné que le processus de classification est implicite. En effet, avec ces méthodes, les valeurs des prédictions sont combinées (par une moyenne pondérée).

Au niveau des prétraitements utilisés, les méthodes recensées dans la table 2.4 utilisent des prétraitements, qui assurent la classification des instances par l'intermédiaire d'un classifieur bayésien naïf (ou une amélioration), ou par la réindexation vectorielle. La méthode QOM n'utilise pas de prétraitements étant donné que la comparaison des extensions repose sur l'agrégation des valeurs de similarités individuelles.

Méthode	**Prétraitement**	**Technique**	**Extensions**
CAIMAN	Réindexation vectorielle	TF IDF - Rocchio	COSINUS
GLUE	Classification	Réseau bayésien naïf sur le contenu et chemin de noms	JACCARD
V-DOC	Réindexation vectorielle	TF IDF	COSINUS
OMAP	Classification	Réseau bayésien naïf	
OPLMAP	Classification	Réseau bayésien naïf et k-plus proches voisins	Consistance
QOM			SIMSET sur les instances directes et celles subsumés directs
RIMOM	Classification	Réseau bayésien naïf	
SBI-HICAL	Classification	Adaptation du réseau bayésien naïf aux hiérarchies	KAPPA
SCM	Réindexation vectorielle	KULLBACK-LEIBLER avec espace vectoriel non euclidien	COSINUS

Tableau 2.4: Comparaison des techniques de combinaison et de sélection des principales méthodes d'alignement

2.7 Conclusion

Le chapitre a présenté les différents aspects des méthodes d'alignement. Il a introduit la notion de l'alignement. Il a aussi proposé trois volets pour l'analyse et l'étude des méthodes d'alignement.

La premier volet concerne les caractéristiques externes des méthodes d'alignement (les

entrées et les sorties). Les méthodes d'alignement peuvent prendre en charge, en entrée, des représentations structurés. Ces structures peuvent être décrites dans différents formats : hiérarchies textuelles, schémas objets ou relationnels, et ontologies (RDF(S) et OWL). Les sorties des méthodes se distinguent par la nature des alignements produits. Ces derniers sont caractérisés selon leurs cardinalités ou encore selon les relations qu'ils incorporent (équivalence ou implication).

Le second volet a étudié les techniques de base des méthodes d'alignement. Ces techniques regroupent les méthodes terminologiques, structurelles, extensionnelles et sémantiques. Ces techniques sont à la base de toutes les méthodes d'alignement.

Le troisième volet propose les stratégies d'alignement. Ces stratégies regroupent la composition d'alignement, l'agrégation de similarités, le calcul global de similarité ainsi que que des méthodes d'apprentissage et probabilistes.

Ces trois volets permettent de réaliser une comparaison synthétique de plusieurs méthodes proposées dans la littérature. Cette comparaison montre qu'une grande majorité des méthodes sont basées sur des mesures de similarité ainsi que les techniques de bases. Par conséquent, elles permettent de détecter seulement des relations d'équivalence entre entités. Les seules méthodes considérant la relation d'implication sont uniquement basées sur une reconnaissance stricte de cette relation à partir d'une base de données lexicale. Le chapitre suivant propose une description détaillé des méthodes d'alignement d'ontologies, que nous introduisons, basées sur l'agrégation et la composition de mesures de similarités. Le chapitre suivant propose la description détaillée des trois méthodes d'alignement proposées dans le cadre de cette thèse. Il décrit le processus de représentation des ontologies à aligner sous la forme de graphe ainsi que le processus d'alignement de chaque méthode.

CHAPITRE 3

Alignement d'ontologies par agrégation de similarités

3.1 Introduction

L'alignement d'ontologies s'est vu devenir nécessaire pour la réduction de l'hétérogénéité. Les méthodes d'alignement opèrent généralement sur deux phases. La première phase permet la transformation des ontologies à aligner sous la forme d'une représentation permettant l'exploitation facile des informations contenues dans les ontologies. La deuxième phase est le processus de l'alignement proprement dit. Ce processus permet de prendre en charge les entités appartenant aux ontologies et de chercher les correspondances qui peuvent exister entre les entités. Les méthodes proposées suivent cette démarche et utilisent leur propre mécanisme d'alignement, en exploitant des mesures de similarité et une démarche d'exploration du voisinage ou des structures internes des ontologies à aligner.

Le reste du chapitre est organisé comme suit : la section 3.2 présente le processus de construction des graphes d'ontologies O-Graph. Dans la section 3.3, nous décrivons en détail la méthode d'alignement EDOLA. La section 3.4 propose la description de la méthode SODA. La dernière section (section 3.5) réalise une présentation des apports de la méthode OACAS.

3.2 Graphe d'ontologies

Les deux ontologies à aligner sont décrites dans le langage OWL (OWL-Lite ou OWL-DL) et représentées sous la forme de deux fichiers. La première étape, du processus d'ali-

gnement, consiste à parcourir ces deux fichiers et extraire toutes les informations existantes dans les deux ontologies. Chaque ontologie est transformée sous la forme d'un graphe O-Graph, OWL Graphe, (Zghal *et al.* (2009)). Cette transformation est assurée par l'intermédiaire du module de construction de graphes associé à chaque méthode d'alignement proposée.

Le module de construction de graphes permet l'édification d'un graphe générique, nommé O-Graph, pour chaque ontologie. Ces graphes permettent la représentation des ontologies OWL sous une forme facile à l'exploitation lors du processus d'alignement. Ainsi, les graphes obtenus reflètent toutes les informations des ontologies à savoir les entités, les relations entre ces derniers et les instances (individus). En effet, la transformation cherche principalement à représenter les ontologies OWL sous la forme d'une structure hiérarchique. Cette structure permet la conservation et la mise en évidence de toutes les informations contenues dans les deux ontologies à aligner. Cette modélisation sous la forme de graphes, permet d'avoir comme sortie deux graphes O-Graph représentant chacun une ontologie. Ainsi, les graphes obtenus sont exploités par la suite au cours de l'étape de l'alignement d'ontologies, en exploitant des mesures de similarité. Ces mesures permettent le calcul du degré de ressemblance (similarité) entre les entités à base des informations décrites dans les deux graphes. En effet, la structure des graphes des ontologies à aligner, est exploitée dans l'objectif d'incorporer les informations des ontologies dans les différentes étapes de l'alignement. L'extraction des ressources, depuis le fichier de l'ontologie, est réalisée par l'intermédiaire de l'API Jena [1]. L'API Jena permet l'extraction de toutes les données sous la forme de triplets RDF (Sujet, Prédicat, Objet). Toutes les informations liées aux deux ontologies sont présentes sous la forme de triplets RDF. Ces triplets sont exploités pour la construction du graphe O-Graph. Le développement des méthodes d'alignement proposées repose principalement sur l'exploitation complète des graphes O-Graph et par conséquent sur la grammaire du langage OWL.

3.2.1 Grammaire du langage OWL

Les méthodes d'alignement proposées exploitent deux versions du langage OWL à savoir : le langage OWL-Lite et le langage OWL-DL. Le langage OWL-DL offre un degré d'expressivité plus riche en terme de définitions des entités ontologiques comparativement à la version OWL-Lite du langage. Une entité ontologique représentée en OWL peut représenter une classe, une relation ou encore une instance (*i.e.*, un individu). Toutes les

1. http ://jena.sourceforge.net

entités dans les ontologies OWL sont des ressources décrites selon le formalisme RDF. La description d'une classe ou d'une relation dans une ontologie OWL est par conséquent réalisée par un ensemble de triplets RDF. Le sujet du triplet correspond au nom de la classe ou de la relation. Le prédicat dans le triplet, qui représente la description d'une entité, est à la fois une primitive du langage OWL et du langage RDF (*c.f.*, à la table 3.1). Le langage OWL offre 33 propriétés avec leurs différentes sémantiques prédéfinies dans OWL et RDF(S). Chaque propriété utilisée dans un triplet RDF apporte une connaissance supplémentaire à propos de l'entité à décrire. La combinaison de toutes ces connaissances assure la construction de la définition de l'entité. Ainsi, la table 3.1 récapitule les constructeurs (primitives) du langage OWL-DL. Par exemple, la primitive *cardinality* impose une restriction de cardinalité dans la définition relationnelle des entités. Elle définit ainsi des relations de type $(0, n)$ entre les entités ontologiques. La dernière colonne de la table 3.1 (notée par la lettre N) désigne le nombre de fois qu'une propriété peut apparaître dans la description d'une entité. De ce fait, dans le cas échéant, une entité peut ne pas avoir de relations avec d'autres entités. En revanche, une entité peut entretenir plusieurs relations possibles avec d'autres entités. Par exemple, la primitive *disjointWith* sert à exprimer la disjonction entre deux ou plusieurs classes.

La grammaire expressive du langage OWL-DL pose cependant quelques défis notables, à savoir : les entités anonymes, le traitement de l'information inférée et l'interprétation des constructeurs. Les entités anonymes (les classes et les individus), dans une ontologie, sont des entités ne possédant ni un identifiant et ni un nom. Ces entités sont en particulier ajoutées pour l'enrichissement de la description d'autres entités. La présence de telles entités dans l'ontologie engendre quelques difficultés lors de la représentation sous la forme d'un graphe O-Graph. En effet, chaque nœud dans le graphe doit avoir son propre identifiant et son propre nom. Ces données sont exploitées, par la suite, pour le calcul des valeurs de la similarité. Le traitement de l'information inférée, à travers le mécanisme d'héritage du langage OWL-DL, propage les informations des entités ascendantes vers les entités descendantes (*i.e.*, relation père-fils), notamment pour les classes et les sous-classes ainsi que les propriétés et les sous-propriétés. L'interprétation des constructeurs de OWL pose quelques ambiguïtés. En effet, le langage OWL comporte quelques constructeurs qui sont sémantiquement similaires mais syntaxiquement différents comme les constructeurs *subClassOf*, *intersectionOf* et *equivalentClass*. Ces trois constructeurs permettent la définition d'une nouvelle classe à partir d'autres.

Propriété	Domaine	Co-domaine	N
rdf :ID	-	-	1
rdf :type	rdfs :Resource	rdfs :Class	≥ 1
rdfs :label	rdfs :Resource	rdfs :Literal	≥ 1
rdfs :comment	rdfs :Resource	rdfs :Literal	≥ 1
rdfs :domain	rdf :Property	rdfs :Class	≥ 1
rdfs :range	rdf :Property	rdfs :Class	≥ 1
rdfs :subClassOf	rdfs :Class	rdfs :Class	≥ 1
rdfs :subProperyOf	rdf :Property	rdf :Property	≥ 1
owl :equivalentClass	owl :Class	owl :class	≥ 1
owl :equivalentProperty	rdf :Property	rdf :Property	≥ 1
owl :sameAs	owl :Thing	owl :Thing	≥ 1
owl :complementOf	owl :Class	owl :class	1
owl :inverseOf	owl :ObjectProperty	owl :ObjectProperty	1
owl :differentFrom	owl :Thing	owl :Thing	≥ 1
owl :disjointWith	owl :Class	owl :class	≥ 1
owl :onProperty	owl :Restriction	rdf :Property	1
owl :allValuesFrom	owl :Restriction	owl :class	1
owl :someValuesFrom	owl :Restriction	owl :class	1
owl :hasValue	owl :Restriction	-	1
owl :cardinality	owl :Restriction	xsd :nonNegativeInteger	1
owl :minCardinality	owl :Restriction	xsd :nonNegativeInteger	1
owl :maxCardinality	owl :Restriction	xsd :nonNegativeInteger	1
owl :distinctNumbers	owl :AllDifferent	rdf :List	1
owl :intersectionOf	owl :Class	rdf :List	1
owl :unionOf	owl :Class	rdf :List	1
owl :oneOf	owl :Class	rdf :List	1
rdfs :seeAlso	rdfs :Resource	rdfs :Resource	≥ 1
rdfs :isDefinedBy	rdfs :Resource	rdfs :Resource	≥ 1
owl :versionInfo	-	-	1
owl :priorVersion	owl :Ontology	owl :Ontology	≥ 1
owl :incompatibleWith	owl :Ontology	owl :Ontology	≥ 1
owl :backwardCompatibleWith	owl :Ontology	owl :Ontology	≥ 1
owl :imports	owl :Ontology	owl :Ontology	≥ 1

Tableau 3.1 — Principales propriétés du langage OWL (Smith *et al.* (2004))

3.2.2 Nœuds d'un O-Graph

Les nœuds du O-Graph représentent les différents types d'entités qui appartiennent à une ontologie OWL. La table 3.2 représente les principales catégories des nœuds qui existent dans le graphe O-Graph. Le nœud *OWLClassNode* permet la représentation des concepts, par exemple, *"voiture"*. Le nœud *OWLInstanceNode* représente les instances des concepts, par exemple, la marque *"Peugeot"* est une instance du concept *"voiture"*. Les types de données sont représentées par le nœud *OWLDataTypeNode*, *e.g.*, le type *"string"* ou le type *"nonNegativeInteger"*. Le nœud *OWLDataValue* décrit les valeurs des types de données. Les deux dernières lignes de la table 3.2 représentent les nœuds qui illustrent les propriétés associées aux classes.

Entité OWL	**Nœud du graphe**
Classe	*OWLClassNode*
Individu	*OWLInstanceNode*
Type de donnée	*OWLDatatypeNode*
Valeur de donnée	*OWLDatavalueNode*
Propriété de nature objet	*OWLObjectPropertyNode*
Propriété de nature type de donnée	*OWLDataTypePropertyNode*

Tableau 3.2 — Entités OWL et leurs équivalents dans un graphe O-Graph

3.2.3 Relations du graphe

Les relations représentées au sein du graphe O-Graph décrivent les relations sémantiques qui existent entre les entités d'une ontologie. Le graphe O-Graph permet de représenter quatre catégories de liens qui sont : le lien de spécialisation, le lien d'attribution, le lien d'instantiation et le lien d'équivalence. Le lien de spécialisation peut exister entre deux classes ou entre deux propriétés. Le lien d'attribution assure un lien entre les classes et leurs propriétés. Le lien d'instantiation permet de réunir entre les classes et les instances de classes, les propriétés et les instances de propriétés, et les nœuds type de données et les valeurs de données. Le lien d'équivalence relie deux classes entre elles par l'intermédiaire de la relation *Equivalent*. La table 3.3 montre la correspondance entre les relations et axiomes du langage OWL et les types de relations au sein du graphe O-Graph.

Axiome OWL-DL	Relation dans O-Graph
Domaine d'une ObjectProperty ou d'une DatatypeProperty	Domain
Image d'une ObjectProperty ou d'une DatatypeProperty	Range
Classe équivalente	EquivalentClass
Sous classe	Subclass
Restriction de type "All Values From"	Allvaluesfrom
Restriction de type "Cardinality"	MaxCardinality ou MinCardinality
Restriction de type "Has property"	Hasproperty
Sous propriété	Subproperty
Propriété inverse	Inverseproperty
Instance d'une classe	InstanceOf
Instance d'un type de donnée	value

Tableau 3.3 — Axiomes et types de liens correspondants dans le graphe O-Graph

3.2.4 Exemple du graphe

La figure 3.1 illustre un exemple restreint d'un O-Graph. Ce graphe comprend trois classes, à savoir : *"Humain"*, *"Appartement"* et *"Chambre"*. La table 3.4 correspond à la description de l'ontologie dans le langage OWL-DL du graphe proposé.

Ces trois classes entretiennent entre elles des liens régis par les axiomes décrits dans la table 3.3. La classe *"Humain"* représente un individu qui peut être spécifié grâce à son nom, son âge et éventuellement son conjoint. Ces deux dernières propriétés sont soumises à des contraintes bien déterminées. En effet, le nom ne peut être qu'une chaîne de caractères, alors que l'âge est un entier strictement positif. Un individu peut posséder un domicile. À son tour, un domicile peut être un *"Appartement"* ayant une adresse et un certain nombre de chambres. La *"Chambre"* peut être caractérisée par sa surface représentée sous forme d'un entier strictement positif. En effet, les types de liens présents dans le graphe O-Graph, ne sont qu'une traduction et une concrétisation des liens sémantiques entretenus entre les entités ontologiques. Le contrôle de la logique est assuré par l'intermédiaire des primitives de restriction (*MaxCardinality*, *MinCardinality*, *Allvaluesfrom*, etc.).

Les graphes des ontologies ainsi représentés sont exploités par la suite dans le processus d'alignement des méthodes proposées. La section suivante décrit d'une manière détaillée la méthode d'alignement EDOLA.

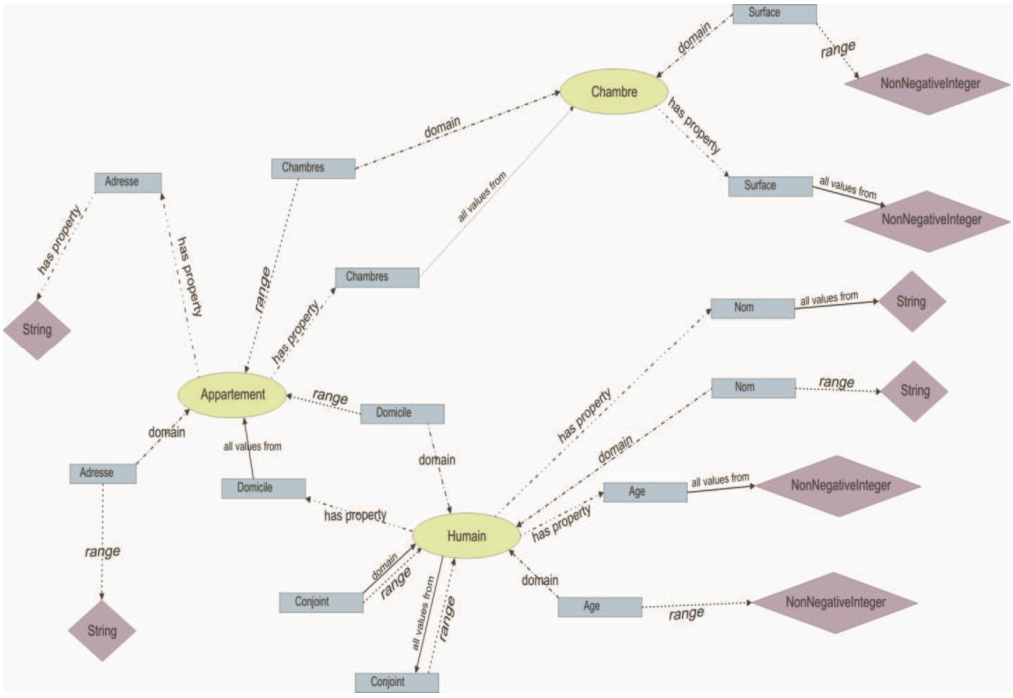


Figure 3.1 — Exemple d'un O-GRAPH

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE rdf :RDF (View Source for full doctype...)>
<rdf :RDF
xmlns="http ://co4.inrialpes.fr/align/Contest/104/onto.rdf#"
xmlns :units="http ://visus.mit.edu/fontomri/0.01/units.owl#"
xmlns :foaf="http ://xmlns.com/foaf/0.1/"
xmlns :ical="http ://www.w3.org/2002/12/cal/#"
xmlns :rdf="http ://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns :xsd="http ://www.w3.org/2001/XMLSchema#"
xmlns :rdfs="http ://www.w3.org/2000/01/rdf-schema#"
xmlns :owl="http ://www.w3.org/2002/07/owl#"
xmlns :wot="http ://xmlns.com/wot/0.1/"
xmlns :dc="http ://purl.org/dc/elements/1.1/"
xmlns :dcterms="http ://purl.org/dc/terms/"
xmlns :dctype="http ://purl.org/dc/dcmitype/">
<owl :Class rdf :ID="Humain"> <rdfs :subClassOf>
</owl :Class> <owl :DatatypeProperty rdf :ID="Nom">
<rdfs :domain rdf :resource="#Humain" />
<rdfs :range rdf :resource="http ://www.w3.org/2001/
XMLSchema#string" /> <rdfs :label xml :lang="fr">key
</rdfs :label> </owl :DatatypeProperty>
<owl :DatatypeProperty rdf :ID="Age">
<rdfs :domain rdf :resource="#Humain" />
<rdfs :range rdf :resource="http ://www.w3.org/2001/
XMLSchema#nonNegativeInteger" />
</owl :DatatypeProperty>
<owl :DatatypeProperty rdf :ID="Adresse">
<rdfs :range rdf :resource="http ://www.w3.org/2001/
<rdfs :range rdf :resource="http ://www.w3.org/2001/
XMLSchema#string" />
<rdfs :label xml :lang="fr">key</rdfs :label>
</owl :DatatypeProperty>
```

```
<owl :Restriction><owl :onProperty rdf :resource="#Domicile"/>
<owl :allValuesFrom rdf :resource="#Appartement" />
</owl :Restriction> </rdfs :subClassOf><rdfs :subClassOf>
<owl :Restriction><owl :onProperty rdf :resource="#Conjoint"/>
<owl :allValuesFrom rdf :resource="#Humain" />
</owl :Restriction></rdfs :subClassOf><rdfs :subClassOf>
<owl :Restriction><owl :onProperty rdf :resource="#Conjoint"/>
<owl :maxCardinality rdf :datatype="http ://www.w3.org/2001/
XMLSchema#nonNegativeInteger">1</owl :maxCardinality>
</owl :Restriction> </rdfs :subClassOf> </owl :Class>
<owl :Class rdf :ID="Appartement">
<rdfs :label xml :lang="fr">Appartement</rdfs :label>
<rdfs :subClassOf> <owl :Restriction>
<owl :onProperty rdf :resource="#Chambres" />
<owl :allValuesFrom rdf :resource="#Chambre" />
</owl :Restriction> </rdfs :subClassOf>
</owl :Class> <owl :Class rdf :ID="Chambre">
<rdfs :label xml :lang="fr">Chambre</rdfs :label>
<owl :DatatypeProperty rdf :ID="Surface">
<rdfs :domain rdf :resource="#Chambre" />
<rdfs :range rdf :resource="http ://www.w3.org/2001/
XMLSchema#nonNegativeInteger" />
</owl :DatatypeProperty>
<owl :ObjectProperty rdf :ID="Domicile">
<rdfs :domain rdf :resource="#Humain" />
</owl :ObjectProperty>
</owl :ObjectProperty>
<owl :ObjectProperty rdf :ID="Conjoint">
<rdfs :domain rdf :resource="#Humain" />
<rdfs :range rdf :resource="#Humain" />
</owl :ObjectProperty>
<owl :ObjectProperty rdf :ID="Chambres">
<rdfs :domain rdf :resource="#Appartement" />
<rdfs :range rdf :resource="#Chambre" />
</owl :ObjectProperty> </rdf :RDF>
```

Tableau 3.4: Ontologie OWL associée au graphe de la figure 3.1

Les ontologies à aligner sont représentées sous la forme de graphes, O-Graph, et sont exploitées par le processus d'alignement. Le processus d'alignement au niveau de chaque méthode permet de prendre en charge toutes les informations contenues dans les deux ontologies à aligner. La section suivante développe une description de la méthode d'alignement EDOLA.

3.3 Description de la méthode EDOLA

La première méthode d'alignement d'ontologies, EDOLA (Zghal *et al.* (2007e), Zghal *et al.* (2007d)), proposée permet d'aligner deux ontologies décrites en format OWL-Lite. Elle produit un alignement représenté sous la forme d'un appariement. L'appariement produit est un fichier RDF sérialisé sous le format XML. Ce fichier comporte les pairs d'entités ontologiques appariées ainsi qu'une mesure de similarité. Chaque pair d'entités

- O_1 et O_2 : deux ontologies à aligner
- VS_L : vecteur de similarité locale
- VS_S : vecteur de similarité sémantique
- VD : vecteur des diamètres de nœuds
Chaque nœud de l'ontologie est caratérisé par :
- Type : type du nœud
- Diamètre : diamètre du nœud
Chaque élément des vecteurs VS_L et VS_S possède les caractéristiques suivantes :
- Nœud$_1$ de l'ontologie O_1
- Nœud$_2$ de l'ontologie O_2
- Valeur de similarité

Tableau 3.5 — Notations utilisées dans la description de la méthode EDOLA

est composée d'une entité appartenant à la première ontologie et d'une autre entité de la seconde ontologie. La mesure de similarité indique le degré de ressemblance entre les deux entités appariées. La méthode, EDOLA (Zghal *et al.* (2007a)), transforme initialement les deux ontologies à aligner sous la forme de deux graphes O-Graph (voir section 3.2). Par la suite, la méthode EDOLA exploite ces deux graphes pour déduire l'appariement entre les différentes entités ontologiques. La méthode opère en deux étapes successives. La première étape permet de calculer la similarité locale, tandis que la seconde étape permet de calculer la similarité globale. La similarité locale est composée de la similarité syntaxique (similarité lexicale). La similarité globale est une mesure exploitant la structure des entités ontologiques à aligner à travers les graphes O-Graph. La table 3.5 présente les différentes notations utilisées dans la description de la méthode EDOLA.

3.3.1 Calcul de similarité locale

Le calcul de la similarité locale des couples d'entités est effectué par l'intermédiaire de l'algorithme 1 (voir la fonction Similarité_Locale). La fonction Similarité_Locale prend en entrée les deux ontologies à aligner, O_1 et O_2, représentées sous la forme de deux graphes O-Graph ainsi que la mesure de similarité à exploiter (FSL : Fonction de Similarité Locale). Elle produit un résultat représenté sous la forme d'un vecteur de similarité locale, VS_L.

Le calcul de la similarité locale s'effectue une seule fois pour chaque couple de nœuds. Les nœuds des deux ontologies sont classés par catégorie (ou type). Au niveau de chaque catégorie, la mesure de la similarité locale est calculée par l'intermédiaire de la similarité syntaxique ou la similarité lexicale. La similarité syntaxique exploite la distance de LEVENSHTEIN (Levenshtein (1966)) ou la distance d'édition. La similarité lexicale est déterminée par l'intermédiaire de l'API WORDNET (Miller (1995)). Le calcul de la similarité locale s'effectue entre les entités en exploitant les différents descripteurs ontologiques de ces dernières, à savoir : les noms, les commentaires et éventuellement les labels.

La fonction CALCULSIMILARITÉLOCALE (voir Algorithme 1, ligne 8) prend en entrée les deux entités ontologiques, représentée par les deux nœuds N_1 et N_2, et la fonction de la similarité locale, FSL. Elle retourne en sortie la valeur de la similarité locale. Cette fonction exploite les mesures syntaxiques ou lexicales. Les valeurs de la similarité locale ainsi que les différents couples de nœuds sont stockées par la suite dans le vecteur de similarité locale, VS_L, (voir Algorithme 1, ligne 10). Le vecteur de similarité locale contient un ensemble de paires d'entités de même catégorie ainsi que la valeur de la similarité locale qui leur est associée. Cette dernière est exploitée par la suite pour le calcul de la similarité globale. La sous-section suivante décrit en détail le processus de calcul de la similarité globale.

3.3.2 Calcul de la similarité globale

La similarité globale est une mesure regroupant la similarité locale et la similarité du voisinage par catégorie. La similarité de voisinage exploite la structure des nœuds (entités) à aligner. La méthode EDOLA explore un voisinage étendu à travers les diamètres de nœuds à aligner (Zghal *et al.* (2007a)).

Le calcul de la similarité globale se fait entre les ensembles de nœuds voisins par catégorie. La fonction SIMILARITÉ_GLOBALE de l'Algorithme 2 organise, par catégorie, les nœuds adjacents au couple d'entités à apparier. Ensuite, elle calcule le diamètre minimal de chaque paire de nœuds à aligner. Le diamètre des nœuds ainsi calculé est exploité pour le calcul de la similarité globale entre chaque paire d'entités. La similarité entre deux groupes de nœuds de même catégorie est déterminée par l'intermédiaire de la similarité basée sur les correspondances, "*Match-Based similarity*", (Valtchev (1999)) :

$$MSim(E_1, E_2) = \frac{\sum_{(e_1,e_2)\in Paires(E_1,E_2)} Sim(e_1, e_2)}{Max(|E_1|, |E_2|)}, \tag{3.1}$$

où E_1 et E_2 représentent deux ensembles de nœuds de même catégorie, appartenant

```
1  Fonction : SIMILARITÉ_LOCALE
   Données :
   – O1 et O2 : deux ontologies à aligner
   – FSL : fonction de similarité locale
   Résultats :
   – VSL : vecteur de la similarité locale
2  début
3      /* parcours des nœuds de l'ontologie O1 */
4      pour chaque (N1 ∈ O1) faire
5          /* parcours des nœuds de l'ontologie O2 */
6          pour chaque (N2 ∈ O2) faire
7              si N1.type=N2.type alors
8                  Sim_L = CALCULSIMILARITÉLOCALE(N1,N2,FSL)
9                  /* Ajouter : 2 nœuds et la valeur de la similarité locale*/
10                 AJOUTERSIMILARITÉ((N1,N2,Sim_L),VS_L)
11     retourner(VS_L)
12 fin
```

Algorithme 1 : La fonction SIMILARITÉ_LOCALE

respectivement à O_1 et O_2. Cette fonction, requiert que les similarités locales des couples d'entités (e_1, e_2) soient déjà calculées, et donne comme résultat les couples de l'ensemble $P = E_1 \times E_2$. Les couples (e_1, e_2), intervenant dans le calcul, doivent présenter les meilleures valeurs de la similarité. Le choix de la meilleure valeur de la similarité se fait par l'intermédiaire d'une technique d'optimisation. Deux approches d'optimisation peuvent être exploitées : l'algorithme glouton (Cormen *et al.* (2001)) et la programmation dynamique (Angel et Bellman (1972)).

L'algorithme glouton suit le principe d'effectuer, étape par étape, un choix optimum local, dans l'espoir d'obtenir un résultat optimum global. L'algorithme glouton effectue des choix locaux au niveau de chaque étape. Ainsi, l'algorithme glouton choisit ce qui lui semble le meilleur, au niveau de chaque étape, pour avancer et espère ensuite que la succession des choix locaux contribue à obtenir une solution globale optimale.

Par contre, la programmation dynamique s'appuie sur une relation entre la solution optimale du problème et celles d'un nombre fini de sous-problèmes. Concrètement, cela signifie que la solution optimale d'un problème est déduite à partir de la solution optimale

du sous problème. Généralement, cette relation est utilisée pour évaluer les solutions des problèmes, c'est-à-dire que le calcul des solutions des sous-problèmes les plus petits pour ensuite déduire étape par étape les solutions de tous les sous-problèmes. La programmation dynamique essaie d'aboutir à une approche d'optimisation globale.

Dans la méthode d'alignement, EDOLA, l'algorithme glouton a été retenu. En effet, l'algorithme glouton choisit un couple d'entités ayant la plus grande valeur de similarité. Cette valeur doit être supérieure ou égale à un seuil fixé. Ensuite, l'algorithme élimine le couple d'entités retenu des couples d'entités restant dans le processus d'optimisation. Il continue par la suite la vérification pour chaque couple jusqu'à ce que il n'existe plus de couples ayant une mesure de similarité supérieure au seuil.

La similarité du voisinage entre les groupes d'entités de même catégorie exploite la similarité basée sur les correspondances déjà calculées. Étant donné, une catégorie de nœuds C et l'ensemble des relations impliqués $R(C)$, la mesure de similarité du voisinage $Sim_V : C \rightarrow [0, 1]$ est définie par :

$$Sim_V(e_1, e_2) = \sum_{x \in R(C)} \Pi_C MSim(R(e_1), R(e_2)). \tag{3.2}$$

La valeur de la similarité du voisinage est normalisée puisque la somme des poids attribués à chaque catégorie est égal à 1 ($\sum(\Pi_C) = 1$). Dans l'approche d'alignement EDOLA, les poids attribués à chaque catégorie sont fixés par défaut et répartis équitablement pour chaque catégorie de nœuds. En utilisant l'équation 3.2, pour calculer la similarité du voisinage des différentes catégories, un système d'équations linéaires est obtenu. Les variables de ce système sont les similarités des couples de nœuds déduites de l'équation 3.1. La résolution du système de l'équation 3.2, est effectuée itérativement. L'itération `0` (voir Algorithme 2, ligne 9) exploite les valeurs de la similarité locale, déjà calculées par l'intermédiaire de l'algorithme 1. Ensuite, l'itération `1` (voir Algorithme 2) utilise l'Equation 3.2 pour calculer les similarités globales entre couples d'entités d'une même catégorie. Les mesures de similarité des catégories intervenant dans le calcul de la similarité d'un couple sont issues de l'itération précédente. Ainsi, l'itération `j` fonctionne de la même manière que l'itération précédente. Le calcul de la similarité globale de chaque couple est basé sur les mesures de similarité calculées à l'itération `(j-1)`. Dans chaque itération, le nombre de candidats à aligner diminue en fonction du diamètre minimum du couple de nœuds à apparier. L'exploration du diamètre de chaque nœud permet la propagation de la similarité à travers le voisinage. La sous-section suivante décrit le processus de calcul du diamètre des nœuds.

Fonction : SIMILARITÉ_GLOBALE
Données :
- O_1 et O_2 : deux ontologies à aligner
- VS_L : vecteur de similarité locale
- Π_L : poids de la similarité locale
- Π_V : poids de la similarité du voisinage

Résultats :
- VS_G : vecteur de la similarité globale

début
  /*calcul du diamètre minimal pour chaque couple de nœuds*/
  pour chaque *($(N_1$,N_2) $\in VS_L$)* **faire**
    d_1=DIAMÈTRENŒUD(N_1)
    d_2=DIAMÈTRENŒUD(N_2)
    $VD(N_1, N_2)$=MINIMUM(d_1,d_2)
  /*itérer jusqu'à atteindre le maximum des diamètres appartenant à VD*/
  pour *(it=1 ; it* $\leq Max_{j\in[1,VD.taille]}VD_j$ *;it++)* **faire**
    /*parcourir le vecteur des similarités de l'itération précédente, le vecteur de similarité de la première itération est VS_L*/
    pour *(j=0 ; j* $< VS_L.taille$ *; j++)* **faire**
      /* vérifier numéro itération et de la valeur du diamètre minimum de nœuds à aligner*/
      si $it < VD_j$ **alors**
        $Voisinage$ = CALCULSIMILARITÉVOISINAGE(N_1, N_2)
        $Sim_G = \Pi_L \times VS_L(N_1, N_2) + \Pi_V \times Voisinage$
        AJOUTERSIMILARITÉ((N_1,N_2,Sim_G),VS_G)
  retourner(VS_G)
fin

Algorithme 2 : La fonction SIMILARITÉ_GLOBALE

3.3.3 Calcul du diamètre des nœuds

Les diamètres des nœuds des couples de nœuds à aligner peuvent être vus comme le point innovant de la méthode EDOLA. En effet, ces diamètres permettent l'exploration étendue du voisinage des deux nœuds dans l'objectif d'explorer toutes les informations qu'elles incorporent. Les diamètres sont prises en compte pour le calcul de la similarité globale, ainsi que la propagation de la similarité à travers le voisinage.

À noter que le diamètre d'un nœud dans le graphe O-Graph désigne la profondeur de ce nœud dans ce graphe, c-à-d., *le nombre de nœuds non encore visités dans le graphe à partir du nœud considéré*. Dans le processus de calcul du diamètre, il faut prendre en charge deux aspects : la nature du graphe (orienté ou non) et les relations circulaires. Dans ce cas, le graphe O-Graph est considéré comme un graphe non orienté. La méthode d'alignement permet de prendre en charge la circularité dans la définition des entités. La fonction Calcul_Diamètre de l'algorithme 3 permet de déterminer le diamètre d'un nœud du graphe de l'ontologie à aligner.

La fonction Calcul_Diamètre prend en entrée le graphe OWL-Graph de l'ontologie, G, le nœud à aligner, N, ainsi que le vecteur de nœuds déjà visités, $Visite$. La fonction retourne en sortie le diamètre maximal du nœud, $MaxDiamètre$. La fonction AjouterNœud permet d'ajouter le nœud N au vecteur de nœuds déjà visités $Visite$ (voir ligne 5 de l'algorithme 3). Par la suite, les nœuds non encore visités sont recherchés et stockés dans le vecteur V (voir ligne 7 de l'algorithme 3) avec la fonction Chercher-Nœud. Cette dernière parcourt le graphe OWL-Graph, G, pour rechercher les nœuds non encore visités. La valeur maximale du diamètre, $MaxDiamètre$, est initialisée à zéro (voir ligne 3 de l'algorithme 3). Les nœuds du vecteur V sont traités et la valeur du diamètre est actualisée (voir ligne 10 de l'algorithme 3). Après le calcul du diamètre, le nœud N ainsi que les valeurs du diamètre sont ajoutés au vecteur $Vdiam$ par l'intermédiaire de la fonction Diamètre (voir ligne 12 de l'algorithme 3). La valeur $MaxDiamètre$ est la valeur maximale du vecteur $Vdiam$ (voir ligne 13 de l'algorithme 3). Les diamètres des nœuds à apparier sont exploités dans le processus de calcul de la similarité globale. Le principe de cette propagation est expliqué dans ce qui suit.

3.3.4 Propagation de la similarité à travers le voisinage

La méthode, EDOLA, effectue une propagation de similarité trés étendue à travers le voisinage. En effet, dans son processus d'alignement, tout le voisinage du couple d'entités

```
1  Fonction : DIAMÈTRE
   Données :
   – G : graphe OWL-GRAPH de l'ontologie
   – N : nœud à aligner
   – Visite : vecteur des nœuds visités
   Résultats :
   – G : MaxDiamètre : Valeur du diamètre
2  début
3      MaxDiameter = 0
4      /* Ajouter le nœud N au vecteur des nœuds visités*/
5      AJOUTERNŒUD(N,Visite)
6      /* Récupérer le vecteur des nœuds non encore visités*/
7      V=CHERCHERNŒUD(G,N,Visite)
8      si V ≠ ∅ alors
9          pour chaque (N1 ∈ V) faire
10             Diamètre=DIAMÈTRE(G,N,Visite)+1
11             /*Ajouter diamètre dans le vecteur Vdiam qui présente tous les diamètres
               du nœud N*/
12             Vdiam=AJOUTERDIAMÈTRE(N,diamètre)
13         MaxDiamètre=MAXIMUM(Vdiam)
14     retourner(MaxDiamètre
15 fin
16 )
```

Algorithme 3 : La fonction CALCUL_DIAMÈTRE.

à aligner est intégré dans le calcul de similarités. Par exemple, considérons la figure 3.2 qui présente deux ontologies O_1 et O_2. Étant donné le couple d'entités (Étudiant(O_1), Étudiant(O_2)), le calcul de la similarité inclut les entités voisines qui entrent en jeu.

Le calcul de similarité du couple en question évoque dans cet exemple le type `objectProperty`. Ainsi, la table 3.6 présente les entités voisines du couple (Étudiant(O_1), Étudiant(O_2)) pour la méthode EDOLA. Ainsi, la méthode EDOLA intègre les mesures de similarité des couples d'entités (encadre(O_1), est_encadré(O_2)) et (réalise(O_1), est_réalisé(O_2)) dans le calcul de similarités du couple (Étudiant(O_1), Étudiant(O_2)). Par conséquent, la mesure de similarité pour ce couple est mieux cernée avec la méthode EDOLA.

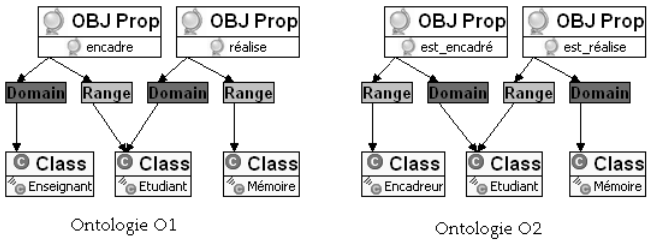


Figure 3.2 — Exemple de deux graphes O-GRAPH des deux ontologies

Étudiant(O_1)	**Étudiant**(O_2)
encadre, réalise	est_encadré, est_réalisé

Tableau 3.6 — Entités voisines du couple (Étudiant(O_1), Étudiant(O_2))

En outre, la méthode EDOLA ne se base pas sur le principe de la stabilité de la mesure de similarité, en utilisant un seuil ϵ défini par l'utilisateur. La méthode EDOLA opère en exploitant la notion du diamètre, c-à-d., la profondeur de l'entité dans le graphe O-GRAPH. Ainsi, la méthode EDOLA n'arrête pas d'itérer sur un couple d'entités qu'après avoir exploité toute sa structure avoisinante.

Le nombre d'itérations, lors de la propagation de la similarité à travers le voisinage, dans la méthode EDOLA est égal au minimum des maximums des diamètres des candidats à apparier (voir Algorithme 2, ligne 7). Dans chaque itération, l'algorithme 2, (voir ligne 13), vérifie les candidats à aligner. Les couples de nœuds dont le diamètre minimum est inférieur au numéro de l'itération courante, ne seront pas traités. Cependant, le diamètre de chaque nœud dans le graphe doit être déterminé (voir Algorithme 2, lignes 5 - 6). Pour déterminer le diamètre d'un nœud, il faut considérer deux aspects. Le premier consiste à vérifier si le graphe est orienté ou non. Le deuxième consiste à tenir compte des relations circulaires. L'algorithme du calcul du diamètre utilise la représentation du graphe O-GRAPH de l'ontologie, et permet de déterminer les diamètres des nœuds existants. En outre, le graphe O-GRAPH considéré est un graphe non orienté. Cependant, il existe des catégories de nœuds pour lesquels un diamètre égal à zéro est donné. En effet, ces nœuds doivent être traités seulement dans l'itération 0, c-à-d., dans l'itération de calcul

de la similarité terminologique. Ces nœuds sont soit de nature type de données (string, non negative integer, etc.), ou valeur de données (une valeur numérique, une chaîne de caractères, etc.). La mesure de similarité de chaque couple de nœuds varie d'une itération à une autre pour prendre en charge les informations incorporées dans le voisinage. Le nombre d'itérations dans EDOLA est égal au maximum des minimums des diamètres des candidats à apparier. La méthode EDOLA est dotée d'un prototype de visualisation (Zghal *et al.* (2007d)) permettant de visualiser graphiquement les ontologies à aligner ainsi que le résultat de l'alignement obtenu. L'annexe A1 donne une brève description du prototype.

La méthode EDOLA permet d'aligner des ontologies OWL-Lite. Cette méthode est basée sur une techniques de propagation de la similarité à travers les voisinages des entités à aligner en fonction de leurs diamètres respectifs. Cette propagation étendue de la similarité permet de prendre en charge toutes les informations du voisinage. La propagation de la similarité à travers les diamètres des nœuds risque d'augmenter le l'espace à travers lequel la propagation s'effectue. Pour cette raison, nous proposons une seconde méthode d'alignement qui permet d'aligner des ontologies décrites en OWL-DL. Le langage OWL-DL est un langage plus expressive que le langage OWL-Lite. La méthode SODA opère dans son processus d'alignement sur un voisinage moins étendu mais elle prend en charge la richesse du langage OWL-DL et exploite plusieurs mesures de similarité pour le calcul de la similarité linguistique.

3.4 Description de la méthode SODA

La deuxième méthode proposée d'alignement d'ontologies, SODA [2], aligne deux ontologies représentées dans le format OWL-DL (Zghal *et al.* (2007c), Zghal *et al.* (2007b)). Elle fournit un résultat décrit sous la forme d'un ensemble de correspondances. L'appariement produit est un fichier RDF sérialisé sous le format XML. Ce fichier comporte les paires d'entités ontologiques appariées ainsi qu'une mesure de similarité. Chaque paire d'entités est composée d'une entité appartenant à la première ontologie et d'une autre entité de la seconde ontologie. La méthode, SODA comme la méthode EDOLA, transforme dans une première phase les deux ontologies à aligner sous la forme de deux graphes O-Graph (voir section 3.2) décrivant les constituants des deux ontologies. Par la suite, la méthode SODA exploite ces deux graphes pour déduire l'appariement lors de la phase d'aligne-

2. La méthode est hébergée dans le lien : www.cril.univ-artois.fr/ m ephu/OAEI2007/systemSODA.rar

- O_1,O_2 : deux ontologies à aligner en format OWL-DL
- V_{SL} : vecteur de la similarité linguistique
- V_{SS} : vecteur de la similarité structurelle
- V_{SSem} : vecteur de la similarité sémantique
Les caractéristiques de chaque nœud sont :
- Type : catégorie du nœud
- Nom : nom du nœud
Les vecteurs V_{SL}, V_{SS} et V_{SSem} se caractérisent par :
- Nœud$_1$: nœud de l'ontologie O_1
- Nœud$_2$: nœud de l'ontologie O_2
- Sim : valeur de similarité

Tableau 3.7 — Notations utilisées dans la description de la méthode SODA

ment. La méthode SODA est une extension de la méthode EDOLA. En effet, la méthode SODA opère sur des ontologies OWL-DL, pour les aligner, à la place des ontologies décrites en OWL-Lite. La méthode SODA est une approche structurelle pour l'alignement d'ontologies OWL-DL. Elle opère en trois étapes successives et exploite la structure des ontologies à travers leurs graphes O-Graph. Le première étape permet de calculer la similarité linguistique à travers des mesures de similarité plus adaptés aux descripteurs des constituants des ontologies à apparier. La seconde étape détermine la similarité structurelle en exploitant la structure des deux graphes O-Graph. La troisième étape déduit la similarité sémantique, en prenant en considération les deux types de similarité déjà calculées, afin de fournir l'alignement résultat. La table 3.7 présente les différentes notations utilisées lors de la description de la méthode SODA.

3.4.1 Calcul de la similarité linguistique

La similarité linguistique est calculée par l'intermédiaire de l'Algorithme 4 (voir la fonction Similarité_Linguistique). La fonction Similarité_Linguistique prend en entrée les deux ontologies à aligner, O_1 et O_2, ainsi que la fonction de la similarité linguistique à utiliser, $Fonct_{SL}$. Les deux ontologies sont prises en charge sous la forme de deux graphes O-Graph. Elle produit en sortie un vecteur de similarité linguistique, V_{SL}. La similarité linguistique est calculée une seule fois. La fonction Similarité_Linguistique organise les entités à apparier sous la forme de groupes selon le type d'entités ontologiques (voir

Algorithme 4, ligne 7). Les entités ontologiques appartenant aux deux ontologies peuvent être des classes, des relations, des propriétés ou des instances. Chaque entité ontologique est décrite par un nom et éventuellement un commentaire et une étiquette. Le calcul de la similarité linguistique s'effectue entre les noms des entités pour les propriétés et les instances. Pour les classes et les relations, le calcul de la similarité linguistique intègre aussi les commentaires et les étiquettes. La similarité linguistique est calculée par l'intermédiaire des mesures de JARO-WINKLER ou MONGE-ELKAN (Monge et Elkan (1996b)). La mesure de JARO-WINKLER est adaptée pour les chaînes de caractères courtes, telles que celles représentant les noms et les étiquettes. Par contre, la mesure MONGE-ELKAN est plus appropriée pour les chaînes longues, telles que les commentaires. La similarité linguistique, Sim_L, est déterminée en prenant en compte les deux entités ontologiques (nœuds des graphes O-GRAPH) à aligner et la mesure de similarité appropriée (voir Algorithme 4, ligne 8). La valeur de la similarité linguistique ainsi que le couple de nœuds sont stockés dans le vecteur de la similarité linguistique, V_{SL}, (voir Algorithme 4, ligne 10). La similarité linguistique pour les différents couples d'entités est exploitée par la suite pour le calcul de la similarité structurelle. La sous-section suivante décrit en détail le processus de calcul de la similarité structurelle de la méthode SODA.

3.4.2 Calcul de la similarité structurelle

La similarité structurelle est la mesure permettant l'exploitation des structures des deux ontologies à aligner, représentées sous la forme de deux graphes O-GRAPH. La prise en charge des structures des deux ontologies permet le traitement de toutes les informations incorporées au sein des ontologies. En effet, l'exploration de chaque nœud appartenant aux couples à apparier permet de prendre en considération les voisins. Ces voisins entretiennent des liens directs et indirectes. Dans la méthode SODA, les liens retenus sont les relations directes à travers les relations d'héritage (relation père-fils). Le calcul de la similarité structurelle se fait par l'intermédiaire de l'Algorithme 5 (voir la fonction SIMILARITÉ_STRUCTURELLE). La fonction SIMILARITÉ_STRUCTURELLE prend en entrée deux ontologies, O_1 et O_2, le vecteur de la similarité linguistique, V_{SL}, ainsi que les poids attribués à chaque catégorie, Π_C. La fonction retourne comme résultat le vecteur de la similarité structurelle, V_{SS}.

La similarité structurelle est calculée en exploitant la similarité linguistique du couple de nœuds à apparier ainsi que la structure avoisinante. Les nœuds voisins adjacents aux entités à aligner sont classés par catégorie, (voir Algorithme 5, ligne 7). Par la suite,

```
1  Fonction : SIMILARITÉ_LINGUISTIQUE
   Données :
   – O1 et O2 : deux ontologies à aligner (sous la forme de deux graphes O-GRAPH)
   – Fonct_SL : mesure de la similarité linguistique (JARO-WINKLER ou MONGE-ELKAN)
   Résultats :
   – V_SL : vecteur de la similarité linguistique
2  début
3      /* parcours des nœuds de l'ontologie O1 */
4      pour chaque (Noeud1 ∈ O1) faire
5          /* parcours des nœuds de l'ontologie O2 */
6          pour chaque (Noeud2 ∈ O2) faire
7              si Noeud1.type==Noeud2.type alors
8                  Sim_L = CALCUL_LINGUISTIQUE(Noeud1,Noeud2,Fonct_SL)
9                  /* Ajouter Noeud1, Noeud2 et Sim_L à V_SL*/
10                 AJOUTERSIMILARITÉ((Noeud1,Noeud2,Sim_L),V_SL)
11     retourner(V_SL)
12 fin
```

Algorithme 4 : La fonction SIMILARITÉ_LINGUISTIQUE

la fonction SIMILARITÉ_STRUCTURELLE calcule la mesure de similarité pour deux ensembles de même catégorie. La fonction EXTRAIRENŒUDS, (voir Algorithme 5, lignes 9 - 11), permet d'extraire pour chaque nœud à aligner les nœuds du voisinage et de les stocker dans les vecteurs V_{Noeud_i}, où $Noeud_i$ désigne un nœud de O_1 ou de O_2. Les deux vecteurs V_{Noeud_1} et V_{Noeud_2} ainsi que les poids attribués à chaque catégorie, Π_C, sont exploités par la fonction CALCUL_SIMILARITÉ_STRUCTURELLE (voir Algorithme 5, ligne 13) pour le calcul de la similarité structurelle, Sim_S. Pour effectuer ce calcul, la mesure de similarité basée sur les correspondances, "*Match-Based similarity*", est exploitée (Valtchev (1999)). Cette mesure est décrite par l'équation 3.1. La similarité basée sur les correspondances permet de calculer la similarité entre les groupes de voisins d'une même catégorie. La similarité structurelle est déterminée en prenant en considération les similarités entre les différents groupes. La similarité structurelle est une somme pondérée des différentes valeurs de similarité des groupes. Elle est calculée comme suite :

$$Sim_S = \sum_{C \in (V_{Noeud_1}, V_{Noeud_2})} \Pi_C MSim(E, E_1). \tag{3.3}$$

La similarité structurelle, Sim_S, est une valeur normalisée. En effet, la somme des poids assignés aux différentes catégories, des nœuds du voisinage, est unitaire ($\sum \Pi_C = 1$). Les poids sont attribués d'une manière équitable entre les différents groupes. Les valeurs de similarité des deux vecteurs de similarité linguistique, c-à-d. V_{SL}, et structurelle, c-à-d. V_{SS}, ainsi obtenues sont combinées pour le calcul de la similarité globale. La section suivante va décrire en détail le processus de calcul de la similarité sémantique.

```
1  Fonction : SIMILARITÉ_STRUCTURELLE
   Données :
   – O1 et O2 : deux ontologies à aligner (sous la forme de deux graphes O-GRAPH)
   – VSL : vecteur de similarité linguistique
   – ΠC : poids de la similarité de chaque catégorie de nœuds
   Résultats :
   – VSS : vecteur de la similarité structurelle
2  début
3      /* parcours des nœuds de l'ontologie O1 */
4      pour chaque (Noeud1 ∈ O1) faire
5          /* parcours des nœuds de l'ontologie O2 */
6          pour chaque (Noeud2 ∈ O2) faire
7              si Noeud1.type==Noeud2.type alors
8                  /*Extraire, dans le vecteur VNoeud1, les nœuds voisins de Noeud1*/
9                  VNoeud1=EXTRAIRENŒUDS(Noeud1)
10                 /*Extraire, dans le vecteur VNoeud2, les nœuds voisins de Noeud2*/
11                 VNoeud2=EXTRAIRENŒUDS(Noeud2)
12                 /*Calculer la similarité structurelle*/
13                 SimS = CALCUL_SIMILARITÉ_STRUCTURELLE(VNoeud1,VNoeud2,ΠC)
14                 /* Ajouter Noeud1, Noeud2 et SimS à VSS*/
15                 AJOUTER((Noeud1,Noeud2,SimS),VSS)
16     retourner(VSS)
17 fin
```

Algorithme 5 : La fonction SIMILARITÉ_STRUCTURELLE

3.4.3 Calcul de la similarité sémantique

La similarité sémantique, de la méthode SODA, est une mesure de similarité agrégée. Elle combine la mesure de la similarité linguistique et la mesure de la similarité structurelle. L'Algorithme 6 prend en entrée les deux ontologies à aligner, O_1 et O_2, les deux vecteurs des similarités linguistique, V_{SL}, et structurelle, V_{SS}, ainsi que les poids attribués pour chaque type de similarité, Π_L et Π_S. L'Algorithme 6 produit un résultat sous la forme d'un vecteur de similarité sémantique, V_{SSem}. La fonction ExtraireSimilarité (voir Algorithme 6, lignes 9 - 11) permet d'extraire, du vecteur de similarité (V_{SL} ou V_{SS}), la valeur de la similarité correspondante aux deux nœuds ($Noeud_1$ et $Noeud_2$). Pour chaque couple d'entités, $Noeud_1$ et $Noeud_2$, appartenant à une même catégorie des deux ontologies à aligner, O_1 et O_2, la similarité sémantique est calculée comme suit (voir Algorithme 6, ligne 13) :

$$Sim_{Sem}(Noeud_1, Noeud_2) = \Pi_L Sim_L(Noeud_1, Noeud_2) + \Pi_S Sim_S(Noeud_1, Noeud_2). \tag{3.4}$$

La similarité sémantique produite est unitaire (comprise entre 0 et 1). En effet, les valeurs des similarité linguistique et structurelle sont à leur tour normalisées ainsi que les poids attribués à chaque type de similarité ($\Pi_L + \Pi_S = 1$). Ces valeurs normalisées de similarité permettent la comparaison des résultats obtenus avec les résultats d'autres méthodes d'alignement de la littérature. Dans la section suivante, la méthode d'alignement OACAS est présentée en détail.

3.5 Description de la méthode OACAS

La méthode d'alignement OACAS est une méthode d'alignement reposant sur un modèle de composition et d'agrégation de similarités (Zghal *et al.* (2009)). Elle prend en entrée deux ontologies OWL-DL. Elle les représente sous la forme de deux graphes O-Graph (voir la section 3.2). Les graphes décrivent les différents constituants des deux ontologies à aligner. En se basant sur le graphe O-Graph, la méthode d'alignement, OACAS, opère en trois étapes successives pour produire l'alignement. La première étape permet de calculer la similarité linguistique composée. La similarité linguistique composée prend en considération tous les descripteurs des entités ontologiques à aligner. La seconde étape détermine la similarité du voisinage par niveau. Le processus de calcul, de la similarité de voisinage, opère sur deux niveaux successifs. La troisième étape agrège les

```
1  Fonction : SIMILARITÉ_SÉMANTIQUE
   Données :
     1. O1 et O2 : deux ontologies à aligner
     2. VSL : vecteur de similarité linguistique
     3. VSS : vecteur de similarité structurelle
     4. ΠL et ΠS : les poids respectifs des similarités linguistique et structurelle
   Résultats : VSG : vecteur de la similarité globale
2  début
3      /* parcours des nœuds de l'ontologie O1 */
4      pour chaque (Noeud1 ∈ O1) faire
5          /* parcours des nœuds de l'ontologie O2 */
6          pour chaque (Noeud2 ∈ O2) faire
7              si Noeud1.type==Noeud2.type alors
8                  /*Extraire la similarité linguistique de Noeud1 et Noeud2 de VSL*/
9                  SimL=EXTRAIRESIMILARITÉ(VSL,Noeud1,Noeud2)
10                 /*Extraire la similarité structurelle de Noeud1 et Noeud2 de VSS*/
11                 SimS=EXTRAIRESIMILARITÉ(VSS,Noeud1,Noeud2)
12                 /*calculer la similarité globale*/
13                 SimSem = ΠL SimL + ΠS SimS
14                 /* Ajouter Noeud1, Noeud2 et SimS à VSG*/
15                 AJOUTER((Noeud1,Noeud2,SimSem),VSSem)
16     retourner(VSG)
17 fin
```

Algorithme 6 : La fonction SIMILARITÉ_SÉMANTIQUE

- O_1, O_2 : les deux ontologies à aligner en format OWL-DL
- V_{SLC} : le vecteur de la similarité linguistique composée
- V_{SV} : le vecteur de la similarité du voisinage
- V_{SA} : le vecteur de la similarité agrégée
Les caractéristiques de chaque nœud sont :
- Type : la catégorie du nœud (classes, relations et instances)
- Descripteur : le descripteur du nœud (nom, commentaire et etiquette)
Les vecteurs V_{SLC}, V_{SV} et V_{SA} se caractérisent par :
- Nœud 1 : le nœud de l'ontologie O_1
- Nœud 2 : le nœud de l'ontologie O_2
- Sim : la valeur de similarité
- MSim : la matrice de la similarité linguistique
Les poids assignés aux différents descripteurs sont :
- Π_C : le poids de la similarité de chaque catégorie de nœuds
- Π_D : le poids des descripteurs des nœuds
- Π_V : la pondération des niveaux de voisinage des nœuds
Les différents types de similarité sont :
- Sim_{LC} : la similarité linguistique composée
- $Sim_{L.D}$: la similarité linguistique par descripteur
- Sim_V : la similarité du voisinage
- Sim_A : la similarité agrégée

Tableau 3.8 — Notations utilisées dans la description de la méthode OACAS

composants de la similarité linguistique composée et la similarité de voisinage par niveau pour déterminer la similarité agrégée. La similarité agrégée représente les résultats de la méthode d'alignement OACAS. La table 3.8 résume les notations algorithmiques utilisées dans les algorithmes de la méthode OACAS.

3.5.1 Similarité linguistique composée

Le calcul des similarités linguistiques s'effectue une seule fois pour chaque nœud d'une même catégorie. Les mesures de similarité linguistiques des couples d'entités de même type (classe, propriété et instance) sont calculées par l'intermédiaire de l'Algorithme 7 (voir la fonction LINGUISTIQUE). Le calcul de la similarité linguistique est effectué entre

les noms des entités pour les propriétés et les instances. Pour les classes, le calcul de la similarité linguistique intègre aussi les commentaires et les étiquettes. Le calcul des similarités linguistiques fait appel à l'utilisation de plusieurs mesures. Ces dernières sont adaptées aux descripteurs (nom, commentaire et étiquette) des entités à aligner. Les différentes valeurs de similarité produites en fonction des descripteurs des entités sont composées. Cette composition est réalisée en pondérant les différentes valeurs de similarité. La somme des poids attribués à chaque mesure est égale à 1. Cette somme unitaire garantit que la valeur de la similarité composée soit normalisée.

La distance de LEVENSHTEIN (Levenshtein (1966)) est exploitée pour calculer la valeur de la similarité entre les noms des entités ontologiques. La mesure LEVENSHTEIN est une distance représentant le nombre minimal d'opérations (insertions, suppressions et substitutions) de caractères nécessaires pour transformer une chaîne de caractères à une autre. Le coût de chaque opération est égal à 1. Par exemple, si nous considérons deux chaînes de caractères égales, s=t="ontologie", on aura la mesure de dissimilarité $\delta(s,t) = 0$ et la mesure de similarité :

$$\delta_{\text{LEVENSHTEIN}}(s,t) = Max(0, \frac{Min(|9|,|9|-0)}{Min(|9|,|9|)}) = Max(0,\frac{9}{9}) = Max(0,1) = 1. \quad (3.5)$$

Le calcul de la similarité entre les commentaires des entités à aligner est réalisé par la mesure Q-GRAM (Ukkonen (1992)). Cette dernière calcule le nombre de *qgrams* en commun, c-à-d., les séquences de q caractères figurant à la fois dans deux chaînes de caractères. La mesure de Q-GRAM (Ukkonen (1992)) permet de calculer le nombre des *qgrams* en commun entre deux chaînes de caractères considérées. En effet, cette mesure calcule le nombre des séquences, de n caractères, qui figurent à la fois dans deux chaînes de caractères distinctes. Par exemple, les *"trigrams"* pour la chaîne "miel" est : *mie* et *iel*. Si nous considérons les deux chaînes, s = "ciel" et t = "miel", avec $q = 3$, les "trigrams" sont respectivement *mie*, *iel*, *cie* et *iel*. Dans ce cas, la mesure de Q-GRAM se calcule comme suit :

$$\sigma_{\text{Q-GRAM}}(s,t) = \frac{1}{Min(|4|,|4|) - 3 + 1} = \frac{1}{2} = 0.5. \quad (3.6)$$

La mesure de JARO-WINKLER (Euzenat et Shvaiko (2007)) est exploitée pour calculer la valeur de la similarité entre les étiquettes. Cette mesure est une amélioration de la mesure de JARO (Jaro (1989)). Elle favorise la correspondance entre les chaînes de caractères en se basant sur le plus long préfixe en commun. Elle est définie comme suit :

$$\sigma_{\text{Jaro-Winkler}}(s,t) = \sigma_{\text{J}}(s,t) + P \times \frac{(1 - \sigma_{\text{Jaro}}(s,t))}{10}. \quad (3.7)$$

avec P le plus long commun préfixe et Q une constante inférieure ou égale à 4. Ainsi, la distance de Jaro (Jaro (1989)) est une mesure asymétrique et elle est exprimée comme suit :

$$\sigma_{\text{Jaro}}(s,t) = \frac{1}{3}(\frac{|com(s,t)|}{|s|} + \frac{|com(s,t)|}{|t|} + \frac{|com(s,t)| - |transp(s,t)|}{|com(s,t)|}). \quad (3.8)$$

La fonction $com(s,t)$ désigne le nombre de caractères *correspondants*, *i.e.*, la différence entre leurs positions dans leurs chaînes respectives. Cette valeur ne doit pas dépasser un certain seuil $l = (\frac{max(|s1|,|s2|}{2})$. Alors que la fonction $transp(s,t)$ représente le nombre de transpositions, qui est obtenu en comparant le ième caractère correspondant de s avec le ième caractère correspondant de t. Le nombre de fois où ces caractères sont différents, divisé par deux, donne le nombre de transpositions. Par exemple, si nous considérons les deux chaînes s = "sabre" et t = "arbre", les détails de calcul de la mesure de Jaro (Jaro (1989)) seront comme suit : $com(s,t) = 6$, $|s| = 5$, $|t| = 5$, $transp(s,t) = \frac{2}{2} = 1$; dans ce cas $\sigma_{\text{Jaro}}(s,t) = \frac{1}{3} \times (\frac{5}{5} + \frac{5}{5} + \frac{5-1}{6}) = 0.88$. La mesure de Jaro-Winkler (Euzenat et Shvaiko (2007)) aura pour valeur : $\sigma_{\text{Jaro-Winkler}} = (0.88 + 3 \times 3 \times \frac{1-0.88}{10}) = 0.98$ dans le cas où P=3 et Q=3. La fonction Linguistique dont le pseudo-code est donné par l'Algorithme 7 permet de calculer les similarités linguistiques composées des couples de nœuds des deux ontologies à aligner O_1 et O_2. Elle prend en entrée les deux ontologies représentées sous la forme de deux graphes O-Graph, ainsi que les fonctions de similarité linguistiques à utiliser, c-à-d., $Fonct$. Elle donne en retour un vecteur similarité linguistique composé, V_{SLC}, de chaque couple de nœuds par catégorie.

La fonction Similarité (voir ligne 9 de l'Algorithme 7) prend en entrée deux nœuds, $Noeud_1$ et $Noeud_1$, ainsi que la fonction de la similarité linguistique dépendante du descripteur du nœud. La fonction Similarité retourne une valeur de similarité linguistique du descripteur Sim_{LD}. Le calcul de cette fonction est assuré par l'intermédiaire d'une des mesures de similarité suivantes : la mesure de Levenshtein ou Q-Gram ou Jaro-Winkler. La similarité linguistique composée, Sim_{LC}, est déterminée en fonction des descripteurs des deux nœuds à aligner (voir lignes 11-13 de l'Algorithme 7). La fonction AjouterNœud permet d'ajouter au vecteur de similarité linguistique composée, V_{SLC}, les deux nœuds ainsi que la valeur de la similarité linguistique composée, Sim_{LC}, (voir ligne 15 de l'Algorithme 7). La similarité linguistique composée pour les différents couples d'entités est exploitée par la suite pour le calcul de la similarité du voisinage.

```
1  Fonction : LINGUISTIQUE
   Données :
   – O1 et O2 : deux ontologies à aligner
   – Fonct : fonctions de similarité linguistique
   – ΠD : poids des descripteurs des Nœuds
   Résultats :
   – VSLC : vecteur de la similarité linguistique composée
2  début
3      pour chaque (Noeud1 ∈ O1) faire
4          /*Parcourir les nœuds de l'ontologie O1*/
5          pour chaque (Noeud2 ∈ O2) faire
6              /*Parcourir les nœuds de l'ontologie O2*/
7              si Noeud1.type == Noeud2.type alors
8                  pour chaque (D ∈ Noeud1.descripteur) faire
9                      SimL.D = SIMILARITÉ(Noeud1.D ,Noeud2.D ,Fonct.D )
10                     /*Calculer la similarité linguistique pour chaque descripteur*/
11                 SimLC = 0
12                 pour chaque (D ∈ (Noeud1.descripteur et Noeud2.descripteur))
                   faire
13                     SimLC = SimLC + ΠD × SimL.D
14                     /*Cumuler les similarités linguistiques des descripteurs*/
15                 AJOUTERNŒUD(Noeud1, Noeud2, SimLC,VSLC)
16                 /*Ajouter Noeud1, Noeud2 et SimLC à VSLC*/
17     retourner(VSLC)
18 fin
```

Algorithme 7 : La fonction LINGUISTIQUE

3.5.2 Similarité du voisinage

La fonction VOISINAGE prend en entrée les deux ontologies à aligner O_1 et O_2, le vecteur de similarité linguistique composée V_{SLC}, les poids assignés à chaque catégorie Π_C et les pondérations attribuées à chaque niveau de voisinage Π_V. Cette fonction produit comme résultat un vecteur de similarité du voisinage V_{SV}. La similarité du voisinage est calculée en exploitant la similarité linguistique composée du couple de nœuds à apparier ainsi que leurs structures. Les nœuds voisins par niveau sont classés par catégorie. Le calcul de la similarité du voisinage s'effectue sur deux niveaux de voisins successifs. Le niveau 1 représente les voisins directs des nœuds à aligner, tandis que le niveau 2 contient les voisins indirects. Les voisins du premier niveau représentent les entités reliées par une relation directe (notion *père-fils*). Les voisins du second niveau sont les entités ontologiques entretenant des liens avec les fils des nœuds à aligner (notion *fils du voisin du niveau 1*).

La fonction EXTRAIRENŒUDS, (voir lignes 9-11 de l'Algorithme 8), permet d'extraire, pour chaque nœud à aligner, les nœuds du voisinage par niveau (1 et 2). Elle est stockée dans les vecteurs $VNoeud_i$, où $Noeud_i$ désigne un nœud de O_1 ou O_2. Les nœuds des vecteurs $VNoeud_1$ et $VNoeud_2$ regroupés par catégorie (ou type) (voir la figure 3.3) sont exploités pour calculer la similarité du voisinage par groupe $MSim$, (voir lignes 16-17 de l'Algorithme 8).

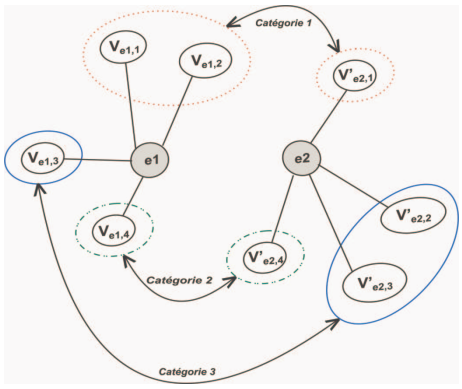


Figure 3.3 — Schéma illustratif du calcul de la similarité du voisinage par catégorie

Elle est calculée par la mesure de similarité basée sur les correspondances "*Match-Based similarity*" (Valtchev (1999)) (voir équation 3.1). La similarité du voisinage, Sim_V, est calculée comme suit :

$$\begin{aligned} Sim_V = \Pi_{V1}(& \sum_{(E_1,E_2)\in(VN\text{œ}ud_1,VN\text{œ}ud_2)} \Pi_{(E_1,E_2)} MSim(E_1,E_2)) \\ +\Pi_{V2}(& \sum_{(E_1,E_2)\in(VN\text{œ}ud_1,VN\text{œ}ud_2)} \Pi_{(E_1,E_2)} MSim(E_1,E_2)). \end{aligned} \tag{3.9}$$

La similarité du voisinage, Sim_V, est normalisée. En effet, la somme des pondérations, assignées aux différents voisins par niveau, est égale à 1 ($\Pi_{V1} + \Pi_{V2} = 1$). Les voisins directs (niveau 1) entretiennent des liens plus étroits que ceux des voisins indirects (niveau 2). Ces liens influencent directement l'alignement à produire. Pour cette raison, la pondération associée au premier niveau, $\Pi_{V1} = 0.8$, est plus importante que celle attribuée au second niveau, $\Pi_{V2} = 0.2$.

De même, la somme des poids attribués aux différentes catégories de nœuds est égale à 1 ($\sum(\Pi_C) = 1$). Ces poids sont répartis d'une manière équitable entre les différentes catégories. La similarité du voisinage est calculée par un processus itératif, niveau par niveau, (voir lignes 14-19 de l'Algorithme 8).

Les valeurs de similarité des deux vecteurs de la similarité linguistique composée, V_{SLC}, et du voisinage, V_{SV}, ainsi obtenues sont combinées pour le calcul de la similarité agrégée. La section suivante décrit en détail le processus de calcul de la similarité agrégée.

3.5.3 Similarité agrégée

La similarité agrégée est une mesure de similarité combinée entre la similarité linguistique composée et la similarité du voisinage. L'Algorithme 9 prend en entrée les deux ontologies à aligner (O_1 et O_2), les deux vecteurs des similarités linguistiques composées (V_{SLC}) et du voisinage (V_{SV}) ainsi que les poids attribués pour la similarité linguistique composée et du voisinage (Π_{LC} et Π_V). Il produit en résultat un vecteur de similarité agrégée (V_{SA}).

La fonction ExtraireSimilarité (voir lignes 8-10 de l'Algorithme 9) permet d'extraire, du vecteur de similarité (V_{SLC} ou V_{SV}), la valeur de la similarité correspondante aux deux nœuds ($Noeud_1$ et $Noeud_2$). Pour chaque couple d'entités, $Noeud_1$ et $Noeud_2$, appartenant à une même catégorie des deux ontologies à aligner, la similarité agrégée est calculée comme suit (voir ligne 12 de l'Algorithme 9) :

Fonction : VOISINAGE

Données :

- O_1 et O_2 : deux ontologies à aligner
- V_{SLC} : vecteur de similarité linguistique
- Π_C : poids de la similarité de chaque catégorie de nœuds
- Π_V : pondération des niveaux de voisinage des nœuds

Résultats :

- V_{SV} : vecteur de la similarité de voisinage

début

pour chaque *($Noeud_1 \in O_1$)* **faire**

/*Parcourir les nœuds de l'ontologie O_1*/

pour chaque *($Noeud_2 \in O_2$)* **faire**

/*Parcourir les nœuds de l'ontologie O_2*/

si $Noeud_1.type == Noeud_2.type$ **alors**

$VNoeud_1$=EXTRAIRENŒUDS($Noeud_1$)

/*Extraire les nœuds voisins de $Noeud_1$ dans les niveaux 1 et 2*/

$VNoeud_2$=EXTRAIRENŒUDS($Noeud_2$)

/*Extraire les nœuds voisins de $Noeud_2$ dans les niveaux 1 et 2*/

$Sim_V = 0$

/*Initialiser la similarité de voisinage à 0*/

pour chaque ($i \in [niveau1, niveau2]$) **faire**

/Parcourir les deux niveaux/

pour chaque *($(E \in (Noeud_1.type))$ et $(E' \in (Noeud_2.type))$)* **faire**

$MSim(E, E') = \frac{\sum_{(i,i') \in Paires(E,E')} Sim_{SLC}(i,i')}{Max(|E|,|E'|)}$

/*Calculer la similarité par type de nœuds et par niveaux*/

$Sim_V = Sim_V + \Pi_{Vi}(\sum_{(E,E')} \Pi_C MSim(E, E'))$

/*Calculer la similarité du voisinage par niveau*/

AJOUTER($Noeud_1$,$Noeud_2$,Sim_V,V_{SV})

/*Ajouter $Noeud_1$, $Noeud_2$ et Sim_V à V_{SV}*/

retourner(V_{SV})

fin

Algorithme 8 : La fonction VOISINAGE

```
1  Fonction : AGRÉGATION
   Données :
   – O1 et O2 : deux ontologies à aligner
   – VSLC : vecteur de similarité linguistique composée
   – VSV : vecteur de similarité du voisinage
   – ΠLC et ΠV : poids des similarités linguistiques composées et du voisinage
   Résultats :
   – VSA : vecteur de la similarité agrégée
2  début
3      pour chaque (Noeud1 ∈ O1) faire
4          /*Parcourir les nœuds de l'ontologie O1*/
5          pour chaque (Noeud2 ∈ O2) faire
6              /*Parcourir les nœuds de l'ontologie O2*/
7              si (Noeud1.type == Noeud2.type) alors
8                  SimLC=EXTRAIRESIMILARITÉ(VSLC,Noeud1,Noeud2)
9                  /*Extraire la similarité linguistique composée de VSLC*/
10                 SimV=EXTRAIRESIMILARITÉ(VSV,Noeud1,Noeud2)
11                 /*Extraire la similarité du voisinage de VSV*/
12                 SimA = ΠLC SimLC + ΠV SimV
13                 /*Calculer la similarité agrégée*/
14                 AJOUTER(Noeud1,Noeud2,SimA,VSA)
15                 /*Ajouter Noeud1, Noeud2 et SimA à VSA*/
16     retourner(VSA)
17 fin
```

Algorithme 9 : La fonction AGRÉGATION

$$Sim_A(e_1, e_2) = \Pi_{LC} Sim_{LC}(e_1, e_2) + \Pi_V Sim_V(e_1, e_2). \quad (3.10)$$

La somme des pondérations est égale à 1 afin de garantir que la similarité agrégée soit comprise entre 0 et 1 ($\Pi_{LC} + \Pi_V = 1$). L'annexe A2 développe un exemple pour illustrer la méthode OACAS.

3.6 Conclusion

Le chapitre a présenté une description détaillée des trois méthodes introduites dans le cadre de cette thèse. Les méthodes proposées exploitent différentes mesures de similarité pour déterminer le degré de ressemblance entre les entités ontologiques. De même, elles exploitent la structure des ontologies pour développer un mécanisme pour la prise en charge des informations sémantiques contenues dans les deux ontologies à aligner.

Le chapitre suivant réalise l'évaluation et l'expérimentation des méthodes (EDOLA, SODA et OACAS). Cette évaluation est réalisée en exploitant un ensemble de métriques d'évaluation qui permettent d'estimer la qualité de l'alignement produit. L'expérimentation est menée sur les bases de test *Benchmark* utilisées par la communauté focalisant sur l'alignement d'ontologies.

CHAPITRE 4

Évaluation et expérimentation

4.1 Introduction

Dans ce chapitre, l'accent est mis sur l'évaluation et l'expérimentation des trois méthodes d'alignement introduites dans ce manuscrit. Cette évaluation permet d'apprécier les résultats obtenus. Cette appréciation est réalisée en exploitant un ensemble de métriques d'évaluation. Ces métriques permettent, d'une part, d'estimer la qualité de l'alignement obtenu ; et d'autre part, de proposer un mécanisme de comparaison avec d'autres méthodes. L'expérimentation est réalisée sur des bases de test *Benchmark*. Ces bases sont mises à la disposition de la communauté, qui travaille dans le cadre de l'alignement d'ontologies pour développer un processus de comparaison des méthodes d'alignement.

Le reste du chapitre est organisé comme suit : la section 4.2 décrit les différentes métriques d'évaluation permettant d'estimer la qualité des résultats obtenus par chaque méthode. La section 4.3 décrit les différents tests appartenant aux bases *benchmark*. Dans la section 4.4, l'expérimentation de la méthode EDOLA est réalisée sur deux aspects : intra-méthode et inter-méthodes. La section 4.5 expérimente la méthode SODA en réalisant une étude comparative des résultats. Dans la section 4.6, la méthode OACAS est évaluée en proposant une démarche de paramétrisation de la méthode. La section 4.6 propose aussi une étude comparative avec d'autres méthodes, de la littérature.

4.2 Métriques d'évaluation

Dans l'objectif de mesurer les performances des méthodes d'alignement, un ensemble de métriques est nécessaire. Ces mesures permettent l'évaluation de la qualité des alignements produits. Elles sont souvent utilisées dans plusieurs domaines tel que la recherche d'infor-

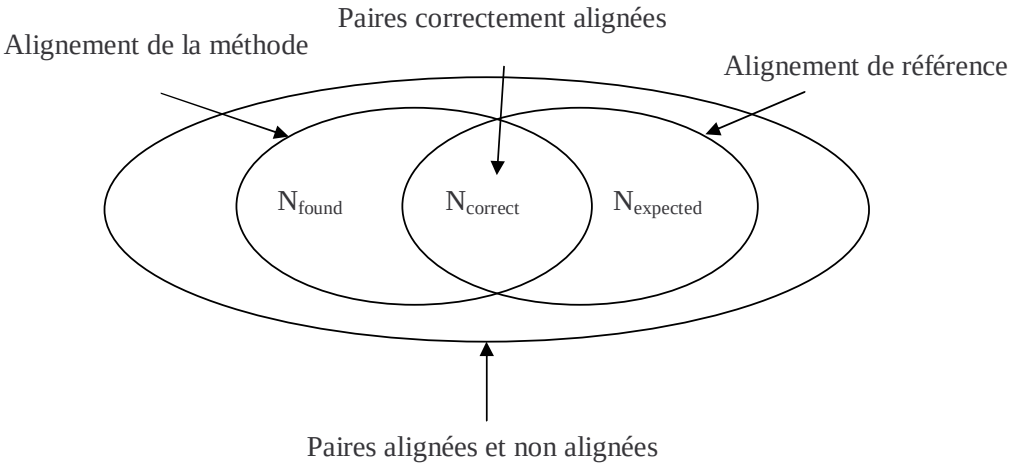


Figure 4.1 — Correspondance entre l'alignement de référence et l'alignement produit

mations, le traitement des requêtes, etc. Ces mesures incluent les métriques de *précision*, *rappel*, $Fallout$ et $Fmesure$ (Do *et al.* (2002)). L'OAEI [1] (Ontology Alignment Evaluation Initiative) retient ces mesures pour apprécier et évaluer la qualité des alignements proposés par les différentes méthodes participant aux différentes campagnes. Les métriques d'évaluation assurent aussi le processus de comparaison des méthodes d'alignement sur la base des résultats obtenus.

La première phase dans le processus d'évaluation de la qualité d'alignement consiste à résoudre le problème manuellement ou à recourir à un alignement fourni par un expert du domaine. Le résultat obtenu manuellement (ou l'alignement de l'expert) est considéré comme l'alignement de référence. La comparaison de l'alignement de référence avec celui obtenu par la méthode d'alignement proposée produit trois ensembles : N_{found}, $N_{expected}$ et $N_{correct}$. L'ensemble N_{found} représente les paires alignées avec la méthode d'alignement. L'ensemble $N_{expected}$ désigne l'ensemble des couples appariés dans l'alignement de référence. L'ensemble $N_{correct}$ désigne l'intersection des deux ensembles N_{found} et $N_{expected}$. Il représente l'ensemble des paires appartenant à la fois à l'alignement obtenu et à l'alignement de référence. La figure 4.1 montre les trois ensembles N_{found}, $N_{expected}$ et $N_{correct}$. L'ensemble qui englobe ces derniers, représente l'ensemble de toutes les paires d'entités (*i.e.*, toutes les paires alignées et non alignées).

La *précision* représente le rapport du nombre de paires pertinentes trouvées, c-à-d.,

1. Ontology Alignment Evaluation Initiative - OAEI - Campaign, http ://oaei.ontologymatching.org/.

$N_{correct}$, rapporté au nombre total de paires obtenues par la méthode d'alignement, c-à-d., N_{found}. Elle renvoie ainsi, la partie des vraies correspondances parmi celles trouvées. Ainsi, la métrique *précision* est définie par :

$$précision = \frac{|N_{correct}|}{|N_{found}|}.$$

Le *rappel* est le rapport du nombre de paires pertinentes trouvées, $N_{correct}$, rapporté au nombre total de paires pertinentes, $N_{expected}$. Il spécifie ainsi, la part des vraies correspondances trouvées. La métrique *rappel* est définie par :

$$rappel = \frac{|N_{correct}|}{|N_{expected}|}.$$

La métrique *Fallout* permet d'estimer le pourcentage d'erreurs obtenues au cours du processus d'alignement. Elle est définie par le rapport des paires erronées (ou non identifiée), $N_{found} - N_{correct}$, rapporté au nombre total des paires trouvées, N_{found}. Cette métrique est définie par :

$$Fallout = \frac{|N_{found}|-|N_{correct}|}{|N_{found}|} = 1 - précision.$$

La métrique *Fmesure* est une mesure harmonique. Elle combine les deux mesures de *précision* et de *rappel*. Elle est définie comme suit :

$$Fmesure = \frac{2 \times précision \times rappel}{précision + rappel}.$$

Les métriques d'évaluation *précision*, *rappel*, *Fallout* et *Fmesure* sont exploitées lors de l'expérimentation des méthodes proposées. La section suivante présentera l'ensemble de tests appartenant à la base *Benchmark* utilisée pour l'expérimentation et l'évaluation des différentes méthodes d'alignement proposées.

4.3 Base *Benchmark*

Dans le cadre de l'étude expérimentale menée, les tests fournis dans la base *Benchmark* mise à la disposition de la communauté par la compétition OAEI sont utilisés. Les expérimentations sont réalisées sur les bases depuis 2007 et jusqu'à 2009. La base *Benchmark* fournit une description d'un nombre de références bibliographiques. Cette base est d'une perception subjective d'une ontologie bibliographique. De ce fait, il peut y avoir différentes classifications de publications.

Chaque test de la base *Benchmark* permet de comparer une ontologie à l'ontologie de référence (l'ontologie du Test 101). L'ontologie de base (ou de référence) est composée comme suit de :

- 33 classes nommées ;
- 24 propriétés de nature objet ;
- 40 propriétés de nature type de données ;
- 56 individus ou instances ;
- 20 instances anonymes.

L'ontologie de référence est considérée comme une *ontologie complète*. Les autres tests sont dérivés et générés d'une façon systématique à partir de l'ontologie de référence (voir tables 4.1 et 4.2). En effet, la suppression d'un certain nombre d'informations permet d'étudier le comportement de la méthode d'alignement lorsque ces informations sont inexistantes. Ainsi, l'objectif principal de cette base de tests est de prendre en charge tous les aspects qui existent dans une ontologie OWL et qui pourraient avoir un impact considérable sur le résultat de l'alignement obtenu ainsi que sur les valeurs des métriques d'évaluation.

La base *Benchmark* 2007 contient 50 tests. Ces derniers sont répartis en trois grandes familles à savoir :

- **La famille 1xx** : contient trois tests. Dans cette famille, l'ontologie de référence est comparée à elle même (Test 101), avec une ontologie de généralisation en format OWL-Lite (Test 103) et avec une ontologie de restriction (Test 104) ;
- **La famille 2xx** : comporte 42 tests. Cette famille compare l'ontologie de référence avec des ontologies modifiées obtenues à partir de l'ontologie de référence ;
- **La famille 3xx** : inclut quatre tests. Ces tests effectuent la comparaison de l'ontologie de référence avec quatre ontologies réelles de références bibliographiques (Tests 301-304, voir table 4.2).

La famille 2xx comporte des tests exploitant des ontologies modifiées. Ces modifications sont réalisées sur l'ontologie de référence et portent principalement sur :

- Les labels (suppression, synonymie, traduction, chaînes aléatoires) ;
- Les commentaires (suppression ou traduction) ;
- L'ordre partiel (suppression, extension, restriction) ;
- Les instances (suppression) ;
- Les propriétés (suppression ou restrictions) ;
- Les classes (extension ou restriction de leur définition).

Test	Description
101	Ontologie de référence
103	Généralisation de langue
104	Restriction de langue
201	Aucun nom
202	Aucun nom, aucun commentaire
203	Aucun commentaire
204	Nommage de conventions
205	Synonymes
206	Traduction (nom)
207	Traduction (nom et commentaire)
208	Nommage de conventions, aucun commentaire
209	Synonymes, aucun commentaire
210	Traduction, aucun commentaire
221	Aucune spécialisation
222	Hiérarchie aplatie
223	Hiérarchie étendue
224	Aucune instance
225	Aucune restriction
228	Aucune propriété
230	Classes aplaties
231	Classes étendues
232	Aucune spécialisation, aucune instance
233	Aucune spécialisation, aucune propriété
236	Aucune instance, aucune propriété
237	Hiérarchie aplatie, aucune instance
238	Hiérarchie étendue, aucune instance
239	Hiérarchie aplatie, aucune propriété
240	Hiérarchie étendue, aucune propriété
241	Aucune spécialisation, aucune instance, aucune propriété
246	Hiérarchie aplatie, aucune instance, aucune propriété
247	Hiérarchie étendue, aucune instance, aucune propriété
248	Aucun nom, aucun commentaire, aucune spécialisation

Tableau 4.1 — Description de la base de tests *Benchmark* - OAEI 2007

Les ontologies sont décrites en langage OWL et sérialisées dans le format RDF/XML. Pour chaque méthode, le résultat de l'alignement doit être évalué par rapport à un alignement de référence faisant partie aussi de la base *Benchmark*. En effet, chaque test de la base est composé d'une ontologie à comparer avec l'ontologie de référence ainsi qu'un alignement de référence qui lui correspond.

Test	Description
249	Aucun nom, aucun commentaire, aucune instance
250	Aucun nom, aucun commentaire, aucune propriété
251	Aucun nom, aucun commentaire, hiérarchie aplatie
252	Aucun nom, aucun commentaire, hiérarchie étendue
253	Aucun nom, aucun commentaire, aucune spécialisation, aucune instance
254	Aucun nom, aucun commentaire, aucune spécialisation, aucune propriété
257	Aucun nom, aucun commentaire, aucune instance, aucune propriété
258	Aucun nom, aucun commentaire, hiérarchie aplatie, aucune instance
259	Aucun nom, aucun commentaire, hiérarchie élargie, aucune instance
260	Aucun nom, aucun commentaire, hiérarchie aplatie, aucune propriété
261	Aucun nom, aucun commentaire, hiérarchie élargie, pas de propriétés
262	Aucun nom, aucun commentaire, aucune spécialisation, aucune instance, aucune propriété
265	Aucun nom, aucun commentaire, hiérarchie aplatie, aucune instance, aucune propriété
266	Aucun nom, aucun commentaire, hiérarchie élargie, aucune instance, aucune propriétés
301	Ontologie réelle semblable à celle de BibTeX/MIT
302	Ontologie réelle et très semblable à sa précédente
303	Ontologie réelle, proposée par l'université de Karlsruhe
304	Ontologie réelle qui contient une hiérarchie étendue de classes, (INRIA)

Tableau 4.2 — Description de la base de tests *Benchmark* - OAEI 2007 (suite)

Pareillement à la base *Benchmark* 2007, les bases de 2008 et 2009, sont constituées d'un ensemble d'ontologies de références bibliographiques. Les bases *Benchmark* de 2008 et 2009 sont semblables à celle de 2007, sauf qu'elles sont constituées d'un ensemble de 111 paires d'ontologies à aligner. De plus, il y a un certain nombre de tests qui s'ajoutent pour

la première fois, et des tests qui sont une duplication d'autres anciens tests. Les nouveaux tests proposés comportent des modifications au niveau de la description terminologique et morphologique d'une ontologie (voir table 4.3). Les tests 101 à 104 restent identiques à ceux du *Benchmark* de 2007. La section suivante s'intéressera à l'évaluation expérimentale de la méthode EDOLA.

4.4 Expérimentations de la méthode EDOLA

L'évaluation expérimentale de la méthode d'alignement EDOLA est menée sur deux aspects complémentaires. L'aspect *"intra-méthode"* vise à l'évaluation de ses performances, c-à-d., le temps d'exécution, de EDOLA par rapport à la variation des constituants des ontologies à aligner et de la mesure de similarité utilisée. Le deuxième aspect, dit *"inter-méthodes"*, permet de comparer les résultats qualitatifs obtenus par EDOLA par rapport à d'autres méthodes, par exemple la méthode OLA.

4.4.1 Aspect "intra-méthode"

L'évaluation "intra-méthode" permet de mesurer les performances de la méthode EDOLA par rapport à la composition structurelle de l'ontologie (modification de la taille de l'ontologie). La table 4.4, présente les temps d'exécution de la méthode EDOLA ainsi que ceux de la construction du graphe O-Graph relatifs à trois séries de tests réalisées. En effet, la même ontologie est utilisée, c-à-d., l'ontologie 101 décrite dans la table 4.1. Chaque test apporte un aspect incrémental de la composition structurelle de l'ontologie. Dans le premier test, l'ontologie de référence est composée uniquement de classes. Ainsi, elle est constituée de 33 entités à aligner. Dans le second test, les 24 propriétés de nature objet sont ajoutées aux classes. Ainsi, le nombre d'entités à aligner devient 57. Dans le troisième test, l'ontologie de référence complète est exploitée. D'après les résultats obtenus et présentés dans la table 4.4, les performances du processus d'alignement dépendent de deux critères :

- *La taille des ontologies à aligner* influence le temps d'exécution[2] de la méthode. Le temps d'exécution augmente d'une manière considérable quand le nombre d'entités à aligner accroît et inversement. Cette augmentation est plus considérable au niveau du module d'alignement qu'au niveau du module de construction du graphe O-Graph.

2. Les expérimentations ont été menées sur une machine tournant sous le système d'exploitation Windows XP familial et dotée d'un processeur Pentium4 3,20 GHz et 512 Mo de mémoire centrale.

Test	Description
201 (-2-4-6-8)	Aucun nom
202 (-2-4-6-8)	Aucun nom, ni commentaire
203	Orthographe modifié
221	Aucune hiérarchie
226	Aucune donnée
227	Différences d'unités
229	Classes vs instances
248 (-2-4-6-8)	Etiquettes brouillées, aucun commentaire, aucune hiérarchie
249 (-2-4-6-8)	Etiquettes brouillées, aucun commentaire, aucune instance
250 (-2-4-6-8)	Etiquettes brouillées, aucun commentaire, aucune propriété
251 (-2-4-6-8)	Etiquettes brouillées, aucun commentaire, hiérarchie aplatie
252 (-2-4-6-8)	Etiquettes brouillées, aucun commentaire, hiérarchie étendue
253 (-2-4-6-8)	Etiquettes brouillées, aucun commentaire, aucune hiérarchie, aucune instance
254 (-2-4-6-8)	Etiquettes brouillées, aucun commentaire, aucune hiérarchie, aucune propriété
257 (-2-4-6-8)	Etiquettes brouillées, aucun commentaire, aucune instance, aucune propriété
258 (-2-4-6-8)	Etiquettes brouillées, aucun commentaire, hiérarchie aplatie, aucun instance
259 (-2-4-6-8)	Etiquettes brouillées, aucun commentaire, hiérarchie étendue, aucun instance
260 (-2-4-6-8)	Etiquettes brouillées, aucun commentaire, aucune aucune propriété
261 (-2-4-6-8)	Etiquettes brouillées, aucun commentaire, hiérarchie étendue, aucune propriété
262 (-2-4-6-8)	Etiquettes brouillées, aucun commentaire, aucune hiérarchie, aucune propriété
265 (-2-4-6-8)	Etiquettes brouillées, aucun commentaire, hiérarchie, aucune propriété
266 (-2-4-6-8)	Etiquettes brouillées, aucun commentaire, hiérarchie étendu, aucune instance, aucune propriété

Tableau 4.3 — Description de la base de tests *Benchmark* - OAEI 2008 & 2009

	Test 1	Test 2	Test 3
Construction O-Graph	3.45	4.70	6.78
EDOLA : Levenshtein	110.46	225.68	357.56
EDOLA : WordNet	148.54	301.98	455.84

Tableau 4.4 — Performances de la méthode EDOLA exprimées en secondes.

Le choix de la fonction de similarité terminologique influence à son tour le temps d'exécution du module d'alignement.

– *Le choix de la fonction de similarité terminologique* agit à son tour sur le temps d'exécution. L'utilisation d'une mesure de similarité simple, comme la mesure de Levenshtein, pour le calcul de la similarité terminologique réduit considérablement le temps d'exécution. En outre, l'utilisation d'une fonction plus complexe comme l'API WordNet augmente considérablement le temps d'exécution du processus d'alignement. Cette augmentation est due au temps consommé par la méthode d'alignement, EDOLA, pour le calcul de la valeur de similarité syntaxique ou lexicale. En effet, l'utilisation de l'API WordNet nécessite des accès disque coûteux pour la recherche des valeurs de similarités au niveau du dictionnaire.

La sous-section suivante sera dédiée à une étude "inter-méthodes" permettant de comparer la méthode EDOLA par rapport à la méthode OLA.

4.4.2 Aspect "inter-méthodes"

L'évaluation "inter-méthodes" effetue une étude comparative de la méthode EDOLA avec la méthode OLA. Les résultats obtenus mettent en évidence les apports (les points forts) de la méthode proposée. Étant donné que la méthode OLA est utilisée pour la comparaison avec la méthode EDOLA, une brève description de cette dernière est proposée. La méthode d'alignement OLA permet d'aligner des ontologies OWL-Lite. Elle exploite un processus d'alignement sur deux étapes : le calcul de la similarité locale et le calcul de la similarité globale. Dans la première étape, la méthode OLA détermine pour chaque couple d'entités la valeur de mesure de la similarité en exploitant soit la mesure de Levenshtein, soit l'API WordNet. Dans la seconde étape, la méthode OLA calcule la similarité globale en exploitant le voisinage direct. Elle classifie les nœuds voisins par catégorie pour calculer la similarité. Le processus de calcul de la similarité globale exploite un processus de propagation et de stabilisation de la similarité globale. En effet, la méthode OLA pro-

page la similarité à travers le voisinage des nœuds à aligner. Le processus de propagation est arrêté quand la stabilisation de la similarité globale est atteinte. La propagation de la similarité globale est arrêtée quand le gain entre deux itérations successives est inférieur à un certain seuil. Ce seuil est pré-indiqué par l'utilisateur. La méthode d'alignement EDOLA est comparée à la méthode OLA. Ce choix de comparaison est argumenté par le fait que les deux méthodes exploitent le même type d'ontologies (OWL-Lite) et possèdent une certaine ressemblance dans leur processus de fonctionnement.

Il est important de rappeler que la méthode EDOLA effectue une propagation de similarité sur tout le voisinage des entités, par contre la méthode OLA opère avec un processus de stabilisation. Ce processus de stabilisation élimine la visite d'un certain nombre de voisins dans le cas où une amélioration n'est pas constatée. La méthode EDOLA exploite la notion de diamètre des entités ontologiques afin d'explorer la totalité de la structure avoisinante du nœud à apparier au niveau des deux ontologies. L'alignement, produit par l'algorithme EDOLA à chaque test, est comparé à l'alignement de référence. Ainsi, les résultats des mesures de qualité sont calculées. La table 4.5 récapitule les résultats obtenus par les deux méthodes d'alignement EDOLA et OLA. Les meilleurs résultats des valeurs de *précision* de EDOLA sont obtenus lorsque les structures d'ontologies sont semblables ou identiques, c-à-d., les tests 101, 104, 222, 224 et 225. Ainsi, EDOLA obtient des valeurs de *précision* pour ces tests qui sont égales à 1. Ceci s'explique par le fait que l'approche EDOLA exploite plus efficacement les structures des entités à aligner. Les ontologies qui possèdent presque la même structure sont correctement alignées. La méthode EDOLA dépasse la méthode OLA dans tous les tests où l'ontologie de référence est comparée à des ontologies réelles (Tests 301 - 304). Ceci s'explique par le fait que la méthode EDOLA exploite, dans son processus de calcul, un voisinage plus étendu que celui de OLA. Ce voisinage permet de prendre en charge la sémantique incorporée au niveau des structures des nœuds à apparier. Les résultats des tests où la valeur de *précision* est moins bonne s'explique par les deux arguments suivants :

- La méthode EDOLA calcule les mesures de similarité des entités de même catégorie. Ceci induit que certains couples d'entités ne sont pas pris en considération par le processus d'alignement. De ce fait, l'ensemble des paires appartenant à la fois à l'alignement obtenu et l'alignement de référence, $N_{Correct}$, serait faible. Par conséquent, la valeur de *précision* est affaiblie. En outre, les couples qui ont été exclus du processus d'alignement peuvent aider à l'augmentation des valeurs de similarité des couples d'entités voisines et par conséquent, augmenter le nombre de couples correctement

	Précision		*Rappel*		*Fallout*		*Fmeasure*	
Test	**EDOLA**	**OLA**	**EDOLA**	**OLA**	**EDOLA**	**OLA**	**EDOLA**	**OLA**
101	**1.00**	**1.00**	1.00	1.00	0.00	0.00	1.00	1.000
103	0.99	**1.00**	0.99	1.00	0.01	0.00	0.99	1.00
104	**1.00**	**1.00**	0.93	1.00	0.00	0.00	0.96	1.00
201	**0.93**	0.71	0.93	0.62	0.07	0.29	0.93	0.66
202	**0.78**	0.66	0.78	0.56	0.22	0.34	0.78	0.61
204	0.72	**0.94**	0.75	0.94	0.28	0.06	0.73	0.94
205	**0.76**	0.43	0.76	0.42	0.24	0.57	0.76	0.42
206	0.82	**0.94**	0.82	0.93	0.17	0.06	0.82	0.93
221	0.76	**1.00**	0.73	1.00	0.24	0.00	0.75	1.00
222	**1.00**	**1.00**	0.97	1.00	0.00	0.00	0.98	1.00
223	0.71	**1.00**	0.72	1.00	0.29	0.00	0.72	1.00
224	**1.00**	**1.00**	0.99	1.00	0.00	0.00	0.99	1.00
225	**1.00**	**1.00**	0.97	1.00	0.00	0.00	0.98	1.00
228	0.80	**1.00**	0.80	1.00	0.20	0.00	0.80	1.00
230	0.91	**0.95**	0.97	0.97	0.09	0.05	0.94	0.96
301	**0.75**	0.42	0.86	0.38	0.25	0.58	0.80	0.40
302	**0.61**	0.37	0.73	0.33	0.39	0.63	0.67	0.35
303	**0.55**	0.41	0.71	0.49	0.45	0.59	0.62	0.45
304	**0.75**	0.74	0.75	0.66	0.25	0.26	0.75	0.70

Tableau 4.5 — Comparaison entre EDOLA et OLA - OAEI 2007.

alignés, c-à-d., $N_{Correct}$;

- La méthode EDOLA n'utilise pas dans son processus d'alignement une comparaison entre les libellés ou les commentaires des entités.

Afin d'évaluer les résultats de l'approche d'alignement proposée, la table 4.5 donne les valeurs des métriques des deux méthodes EDOLA et OLA. Ainsi, la méthode d'alignement, EDOLA, est meilleure par rapport à la méthode OLA dans la majorité des tests (63% des tests). En effet, la méthode d'alignement EDOLA fournit des mesures de qualité plus performantes sur presque la totalité des tests. Ces meilleurs résultats s'expliquent par les apports de la méthode à savoir :

- La méthode EDOLA effectue une propagation de similarité nettement meilleure que

celle de OLA ;
- La méthode EDOLA, contrairement à OLA, ne se base pas sur la stabilité de la mesure de similarité en utilisant un seuil ϵ défini par l'utilisateur. La valeur par défaut de ce seuil est fixé à 0.01 pour la méthode OLA.

La section suivante procédera à l'évaluation expérimentale de la méthode SODA.

4.5 Expérimentations de la méthode SODA

Dans cette section, une évaluation est présentée permettant de comparer la méthode SODA par rapport à la méthode ASCO2 (Bach (2006)), qui permet d'aligner des ontologies OWL-DL. Elle permet de comparer les résultats obtenus par les deux méthodes.

4.5.1 Évaluation

La table 4.6 récapitule les résultats obtenus par les deux méthodes d'alignement SODA et ASCO2. La méthode ASCO2 est retenue car elle aligne des ontologies en format OWL-DL. En se basant sur les valeurs des métriques d'évaluation de la qualité de l'alignement (mesures de *précision*, *rappel*, *Fallout* et *Fmesure*), la méthode SODA présente de meilleurs résultats (voir table 4.6) dans presque la majorité des tests. Dans ce cadre, il est important de rappeler que la méthode SODA effectue un calcul de la similarité sémantique en exploitant la similarité linguistique et la similarité structurelle. La méthode SODA exploite la notion de structure du voisinage pour le calcul de la similarité structurelle. En effet, dans une première étape, la méthode SODA classe les nœuds adjacents par catégorie. Par la suite, elle calcule la similarité maximale par catégorie pour déterminer la similarité structurelle. L'alignement, produit par la méthode SODA au niveau de chaque test, est comparé à l'alignement de référence. Ainsi, les valeurs des mesures de qualité d'alignement (*précision*, *rappel*, *Fallout* et *Fmesure*) sont déterminées.

Les meilleurs résultats des valeurs de *précision* de SODA sont obtenus lorsque les structures des deux ontologies à aligner sont semblables (ou identiques), c-à-d. les familles de tests 10x, 22x et 23x (voir table 4.7). Ainsi, la méthode SODA obtient des valeurs de *précision* pour ces tests qui sont égales à 1.00 (voir table 4.7). Ceci s'explique par le fait que l'approche SODA explore plus efficacement les structures avoisinantes des entités à aligner à travers la similarité structurelle. Les entités qui possèdent des structures semblables (ou similaires) sont correctement alignées. Les résultats des tests où la valeur de *précision* est moins performante s'explique par le fait que SODA calcule les mesures de similarité

des entités de même catégorie. Ceci implique que certains couples d'entités ne sont pas pris en considération par le processus d'alignement. Par conséquent, l'ensemble des paires appartenant à la fois à l'alignement obtenu et l'alignement de référence, c-à-d. $N_{Correct}$, est faible. Par conséquent, la valeur de *précision* est affaiblie. En outre, les couples qui ont été exclus du processus d'alignement peuvent augmenter les valeurs des mesures de similarité des couples d'entités voisines et par la suite, augmenter le nombre de couples correctement alignés, c-à-d. $N_{Correct}$.

	Précision		*Rappel*		*Fallout*		*Fmesure*	
Test	**SODA**	**ASCO2**	**SODA**	**ASCO2**	**SODA**	**ASCO2**	**SODA**	**ASCO2**
101	**1.00**	**1.00**	1.00	1.00	0.00	0.00	1.00	1.00
103	**1.00**	**1.00**	1.00	1.00	0.00	0.00	1.00	1.00
104	**1.00**	**1.00**	1.00	1.00	0.00	0.00	1.00	1.00
201	0.33	**0.87**	0.01	0.87	0.67	0.13	0.02	0.87
202	**0.33**	0.31	0.01	0.30	0.67	0.69	0.02	0.30
203	**1.00**	**1.00**	1.00	1.00	0.000	0.00	1.00	1.00
204	**1.00**	**1.00**	0.93	1.00	0.000	0.00	0.96	1.00
205	0.82	**1.00**	0.33	0.85	0.18	0.00	0.47	0.92
221	**1.00**	**1.00**	1.00	1.00	0.00	0.00	1.00	1.00
222	**1.00**	**1.00**	1.00	1.00	0.00	0.00	1.00	1.00
223	**1.00**	0.98	1.00	0.00	0.02	1.00	1.00	0.98
224	**1.00**	**1.00**	1.00	1.00	0.00	0.00	1.00	1.00
225	**1.00**	**1.00**	1.00	1.00	0.00	0.00	1.00	1.00
228	**1.00**	0.97	1.00	1.00	0.00	0.03	1.00	0.98
230	0.94	**1.00**	1.00	1.00	0.06	0.00	0.97	1.00
231	**1.00**	**1.00**	1.00	1.00	0.00	0.00	1.00	1.00
232	**1.00**	**1.00**	1.00	1.00	0.00	0.03	1.00	0.98
233	**1.00**	0.97	1.00	1.00	0.00	0.03	1.00	0.98
236	**1.00**	0.97	1.00	1.00	0.00	0.03	1.00	0.98
237	**1.00**	**1.00**	1.00	1.00	0.00	0.00	1.00	1.00
238	**1.00**	0.98	1.00	0.98	0.00	0.02	1.00	0.98
239	0.88	**0.97**	1.00	1.00	0.12	0.03	0.94	0.98
240	0.89	**0.97**	1.00	0.94	0.11	0.03	0.94	0.95
241	**1.00**	0.97	1.00	1.00	0.00	0.03	1.00	0.98

246	0.88	**0.97**	1.00	1.00	0.12	0.03	0.94	0.98
247	0.89	**0.97**	1.00	0.94	0.11	0.03	0.94	0.95
248	0.25	**0.30**	0.01	0.29	0.75	0.70	0.02	0.29
249	**0.33**	0.32	0.01	0.31	0.67	0.68	0.02	0.31
252	0.16	**0.26**	0.01	0.22	0.84	0.74	0.02	0.24
253	0.33	**0.42**	0.01	0.30	0.67	0.58	0.02	0.35
258	**0.60**	0.36	0.10	0.37	0.40	0.64	0.17	0.36
259	0.20	**0.38**	0.01	0.32	0.80	0.62	0.02	0.35
301	0.86	**0.92**	0.39	0.57	0.150	0.08	0.54	0.70
302	**0.96**	0.74	0.58	0.35	0.04	0.26	0.72	0.48
303	**0.70**	0.51	0.81	0.76	0.30	0.49	0.75	0.61
304	**0.96**	0.90	0.95	0.87	0.04	0.10	0.95	0.88

Tableau 4.6: Résultats de la comparaison entre SODA et ASCO2 - OAEI 2007

4.5.2 Étude comparative des résultats obtenus

L'évaluation expérimentale de la méthode d'alignement SODA permet de comparer les résultats qualitatifs obtenus par la méthode SODA à la méthode ASCO2 (Bach (2006)). Dans le cadre des expérimentations menées, les tests fournis dans la base *Benchmark* de l'AOEI 2007 sont exploitées (voir tables 4.1 et 4.2).

À partir des données présentées dans la table 4.6, la méthode d'alignement SODA se montre meilleure sur plusieurs tests par rapport à la méthode ASCO2. En effet, la méthode d'alignement, SODA, fournit des valeurs de métrique d'évaluation plus performantes sur la quasi-totalité des tests réels, la famille de tests 3xx, voir aux Tables 4.6 et 4.7. Ces meilleurs résultats s'expliquent par les deux aspects suivants :

– La méthode SODA exploite le voisinage des nœuds à aligner pour le calcul de la similarité sémantique, tandis que la méthode ASCO2 exploite la notion de similarité finale (Bach (2006)). La similarité sémantique est calculée en se basant sur les similarités linguistiques et structurelles des entités à aligner. La similarité linguistique se calcule par l'intermédiaire des mesures de similarité Jaro-Winkler ou Monge-Elkan. La similarité structurelle exploite la similarité linguistique ainsi que la similarité du voisinage (structurelle) par catégorie. Par contre, la méthode ASCO2 calcule la similarité

	Précision		*Rappel*	
Test	**SODA**	**ASCO2**	**SODA**	**ASCO2**
10x	**1.00**	**1.00**	1.00	1.00
20x	0.70	**0.84**	0.46	0.80
22x	**1.00**	0.99	1.00	1.00
23x	**0.98**	**0.98**	1.00	1.00
24x	0.71	**0.75**	0.67	0.75
25x	0.32	**0.36**	0.03	0.30
30x	**0.87**	0.77	0.68	0.64
Moyenne	0.80	0.81	0.70	0.78

Tableau 4.7 — Résultats de la comparaison entre SODA et ASCO2 par catégorie de tests - OAEI 2007.

partielle des entités à aligner. Cette similarité se limite au calcul de la similarité entre les descripteurs des entités à aligner ;

– La méthode SODA, calcule la similarité sémantique en agrégeant les éléments des similarités linguistiques et structurelles. La similarité sémantique représente une mesure intégrée des différentes mesures de similarité des entités à aligner. Par contre, la méthode ASCO2 calcule la similarité finale uniquement en exploitant la similarité partielle. La similarité finale est calculée par un processus itératif (un certain nombre d'itérations) ou par une stabilisation de la similarité finale (Bach (2006)).

La section suivante s'intéressera à l'étude expérimentale de la méthode OACAS.

4.6 Expérimentation de la méthode OACAS

L'expérimentation de la méthode OACAS s'effectue dans le but de paramétrer la méthode et de chercher la meilleure combinaison de mesures de similarité. Les résultats obtenus par la méthode mettent en valeur les aspects forts de la méthode. Une étude comparative permet de situer la méthode par rapport à quelques méthodes de la littérature.

4.6.1 Paramétrage

L'objectif des expérimentations effectuées avec la méthode OACAS est la détermination de la meilleure combinaison de composition des mesures de similarité linguistiques. En effet, lors de l'expérimentation, plusieurs mesures ont été testées. L'objectif consiste à prendre en charge les mesures les plus appropriées selon la catégorie du nœud à aligner. Afin d'atteindre l'objectif fixé, 27 combinaisons de tests sont réalisées. Chaque combinaison de test permet l'utilisation d'un arrangement particulier de mesures de similarités pour le calcul de la similarité linguistique. La table 4.8 illustre les différentes combinaisons de mesures linguistiques utilisées dans ces tests. Les combinaisons (1-3) exploitent la même mesure de similarité pour les différents descripteurs (nom, commentaire et étiquette) des entités à aligner. La ligne 1 de la table 4.8 indique que la mesure de Levenshtein est appliquée aux trois descripteurs des entités à apparier. Dans les combinaisons (4-21), les expérimentations exploitent seulement deux mesures de similarité pour les trois descripteurs des entités. En effet, une même mesure de similarité est appliquée à deux descripteurs.

La ligne 4 de la table 4.8 indique que la mesure de Levenshtein est utilisée pour calculer la similarité entre les noms des entités, tandis que la mesure Jaro-Winkler est appliquée pour le calcul de la similarité entre les commentaires et les étiquettes des entités. La ligne 10 de la table 4.8 stipule que la combinaison des mesures de similarité (Jaro-Winkler, Jaro-Winkler et Levenshtein) est exploitée pour calculer les valeurs de similarité respectivement entre les noms, les commentaires et les étiquettes. À partir de la ligne 22 et jusqu'à la ligne 27 de la table 4.8, les combinaisons utilisent trois mesures de similarité différentes pour les trois descripteurs de l'entité. La ligne 22 de la table 4.8 indique que la combinaison des mesures de similarité (Levenshtein, Jaro-Winkler et Q-Gram) est exploitée pour le calcul des valeurs de similarité respectivement entre les noms, les commentaires et les étiquettes.

Au cours des tests effectués, plusieurs pondérations ont été attribuées aux différents descripteurs (noms, commentaires et étiquettes). En effet, les nœuds à aligner peuvent avoir des descripteurs différents. En fonction de l'existence de chaque descripteur, un poids approprié lui a été attribué. Dans le cas où les nœuds sont décrits par leurs trois descripteurs, les poids sont de 0.8, 0.1 et 0.1 respectivement pour les noms, les commentaires et les étiquettes. Dans la situation où les nœuds comportent seulement des descripteurs de noms et de commentaires, les pondérations sont respectivement de 0.85 et 0.15. Les poids de 0.85 et 0.15 sont assignés respectivement aux noms et aux étiquettes quand ces deux

	Descripteurs des entités		
Combinaison	**Nom**	**Commentaire**	**Etiquette**
1	Levenshtein	Levenshtein	Levenshtein
2	Jaro-Winkler	Jaro-Winkler	Jaro-Winkler
3	Q-Gram	Q-Gram	Q-Gram
4	Levenshtein	Jaro-Winkler	Jaro-Winkler
5	Levenshtein	Q-Gram	Q-Gram
6	Q-Gram	Levenshtein	Levenshtein
7	Jaro-Winkler	Q-Gram	Q-Gram
8	Jaro-Winkler	Levenshtein	Levenshtein
9	Q-Gram	Jaro-Winkler	Jaro-Winkler
10	Jaro-Winkler	Jaro-Winkler	Levenshtein
11	Jaro-Winkler	Jaro-Winkler	Q-Gram
12	Q-Gram	Q-Gram	Jaro-Winkler
13	Q-Gram	Jaro-Winkler	Q-Gram
14	Q-Gram	Q-Gram	Levenshtein
15	Q-Gram	Levenshtein	Q-Gram
16	Levenshtein	Q-Gram	Levenshtein
17	Levenshtein	Levenshtein	Jaro-Winkler
18	Levenshtein	Jaro-Winkler	Levenshtein
19	Jaro-Winkler	Q-Gram	Jaro-Winkler
20	Jaro-Winkler	Levenshtein	Jaro-Winkler
21	Levenshtein	Levenshtein	Q-Gram
22	Levenshtein	Jaro-Winkler	Q-Gram
23	Jaro-Winkler	Levenshtein	Q-Gram
24	Q-Gram	Levenshtein	Jaro-Winkler
25	Q-Gram	Jaro-Winkler	Levenshtein
26	**Levenshtein**	**Q-Gram**	**Jaro-Winkler**
27	Jaro-Winkler	Q-Gram	Levenshtein

Tableau 4.8 — Combinaisons des mesures de similarité linguistiques composées dans la méthode OACAS

descripteurs décrivent les nœuds. Ces poids sont attribués après avoir effectué des séries de tests qui nous ont permis de les choisir. En effet, le nom de chaque nœud représente l'information la plus pertinente le décrivant. À cet effet, la pondération la plus élevée est assignée au descripteur nom. Les résultats obtenus, lors de cette expérimentation, sont décrits dans les tables 4.9, 4.10 et 4.11.

Les tables 4.9, 4.10 et 4.11 récapitulent les valeurs moyennes obtenues pour les métriques de *précision* et de *rappel* pour la base *Benchmark* 2007. Ces valeurs sont issues de 27 combinaisons de tests effectuées avec la base *Benchmark*. Chaque combinaison, comme déjà mentionné, exploite un arrangement particulier des mesures de similarité linguistiques. Les résultats obtenus lors de l'expérimentaion sont présentés dans les Tables 4.9, 4.10 et 4.11. Ces résultats sont regroupés en huit familles (10x, 20x, 22x, 23x, 24x, 25x, 26x et 30x) pour chaque combinaison. Au niveau de chaque famille, les valeurs moyennes de la *précision* et du *rappel* sont indiquées. Ces valeurs sont déterminées après avoir effectué tous les tests de la famille. Les tables 4.9, 4.10 et 4.11 montrent que la combinaison 26 (la meilleure combinaison) présente les meilleurs résultats sur les différentes combinaisons de tests. En effet, la combinaison 26 exploite trois mesures différentes de similarités linguistiques (Levenshtein, Q-Gram et Jaro-Winkler). La mesure de Levenshtein se montre la plus adaptée pour le calcul de la similarité linguistique entre les noms des entités à aligner. La mesure de Q-Gram est plus appropriée pour le calcul de la similarité linguistique entre les commentaires des entités à apparier. La mesure de Jaro-Winkler s'avère comme la meilleure mesure pour la détermination de la valeur de la similarité linguistique entre les étiquettes des entités ontologiques.

En effet, les noms et les étiquettes des entités ontologiques sont représentés par des chaînes de caractères courtes. Pour ces deux types de chaînes, les mesures de Levenshtein et Jaro-Winkler sont mieux indiquées pour le calcul de la similarité linguistique. Les commentaires sont des descripteurs constitués des chaînes de caractères. Ces chaînes sont composées de plusieurs mots. Pour ce type de descripteurs, la mesure de Q-Gram permet de fournir les meilleures valeurs de la similarité linguistique.

La table 4.12 récapitule les valeurs des métriques de *précision* et de *rappel* pour la meilleure combinaison obtenue lors des tests déjà effectués sur la base *Benchmark* dans sa version 2007. En effet, la combinaison 26 utilise trois différentes mesures de similarités linguistiques, qui sont : Levenshtein, Q-Gram et Jaro-Winkler, respectivement pour les noms, les étiquettes et les commentaires.

Les tables 4.13 et 4.14 récapitulent les valeurs des métriques de *précision* et de *rappel*

	Précision	*Rappel*	*Précision*	*Rappel*	*Précision*	*Rappel*
Test	***Combinaison 1***		***Combinaison 2***		***Combinaison 3***	
10x	0.60	0.99	0.75	0.75	0.00	0.00
20x	0.38	0.16	0.38	0.38	0.00	0.00
22x	0.83	0.82	1.00	1.00	0.00	0.00
23x	0.85	0.87	0.85	0.87	0.00	0.00
24x	0.76	0.67	0.70	0.67	0.00	0.00
25x	0.27	0.01	0.17	0.01	0.00	0.00
26x	0.00	0.00	0.00	0.00	0.00	0.00
30x	0.86	0.64	0.86	0.67	0.00	0.00
	Combinaison 4		***Combinaison 5***		***Combinaison 6***	
10x	1.00	1.00	0.72	0.92	0.00	0.00
20x	0.66	0.43	0.64	0.40	0.00	0.00
22x	1.00	1.00	0.97	0.90	0.00	0.00
23x	0.98	1.00	0.94	0.85	0.00	0.00
24x	0.76	0.67	0.71	0.50	0.00	0.00
25x	0.27	0.01	0.27	0.01	0.00	0.00
26x	0.00	0.00	0.00	0.00	0.00	0.00
30x	0.89	0.64	0.81	0.39	0.00	0.00
	Combinaison 7		***Combinaison 8***		***Combinaison 9***	
10x	1.00	1.00	1.00	1.00	0.00	0.00
20x	0.64	0.47	0.64	0.47	0.00	0.00
22x	1.00	0.98	1.00	1.00	0.00	0.00
23x	0.97	0.95	0.97	1.00	0.00	0.00
24x	0.69	0.56	0.70	0.67	0.00	0.00
25x	0.18	0.01	0.17	0.01	0.00	0.00
26x	0.00	0.00	0.00	0.00	0.00	0.00
30x	0.83	0.56	0.86	0.68	0.00	0.00

Tableau 4.9 — Valeurs de *précision* et de *rappel* par famille de tests de la méthode OACAS - AOEI 2007 (Partie 1)

	Précision	*Rappel*	*Précision*	*Rappel*	*Précision*	*Rappel*
Test	***Combinaison 10***		***Combinaison 11***		***Combinaison 12***	
10x	1.00	1.00	1.00	1.00	0.00	0.00
20x	0.89	0.48	0.89	0.49	0.00	0.00
22x	1.00	1.00	1.00	1.00	0.00	0.00
23x	1.00	1.00	1.00	1.00	0.00	0.00
24x	1.00	0.68	1.00	0.68	0.00	0.00
25x	0.50	0.02	0.50	0.01	0.00	0.00
26x	1.00	0.00	0.00	0.00	0.00	0.00
30x	1.00	0.81	1.00	0.64	0.00	0.00
	Combinaison 13		***Combinaison 14***		***Combinaison 15***	
10x	0.00	0.00	0.00	0.00	0.00	0.00
20x	0.00	0.00	0.00	0.00	0.00	0.00
22x	0.00	0.00	0.00	0.00	0.00	0.00
23x	0.00	0.00	0.00	0.00	0.00	0.00
24x	0.00	0.00	0.00	0.00	0.00	0.00
25x	0.00	0.00	0.00	0.00	0.00	0.00
26x	0.00	0.00	0.00	0.00	0.00	0.00
30x	0.00	0.00	0.00	0.00	0.00	0.00
Test	***Combinaison 16***		***Combinaison 17***		***Combinaison 18***	
10x	1.00	0.96	1.00	1.00	1.00	0.99
20x	0.89	0.42	0.89	0.43	0.89	0.43
22x	1.00	0.96	1.00	1.00	1.00	0.99
23x	1.00	0.97	1.00	1.00	1.00	0.99
24x	1.00	0.67	1.00	0.67	1.00	0.67
25x	0.50	0.01	0.50	0.01	0.50	0.01
26x	0.00	0.00	0.00	0.00	0.00	0.00
30x	0.75	0.71	1.00	0.71	1.00	0.00

Tableau 4.10 — Valeurs de *précision* et de *rappel* par famille de tests de la méthode OACAS - AOEI 2007 (Partie 2)

	Précision	*Rappel*	*Précision*	*Rappel*	*Précision*	*Rappel*
Test	***Combinaison 19***		***Combinaison 20***		***Combinaison 21***	
10x	1.00	1.00	1.00	1.00	1.00	0.99
20x	0.89	0.50	0.89	0.49	0.89	0.43
22x	1.00	1.00	1.00	1.00	1.00	0.99
23x	1.00	1.00	1.00	1.00	1.00	0.99
24x	1.00	0.68	1.00	0.68	1.00	0.67
25x	0.50	0.02	0.02	0.02	0.50	0.01
26x	0.00	0.00	0.00	0.00	0.00	0.00
30x	1.00	0.64	0.80	0.80	1.00	0.62
Test	***Combinaison 22***		***Combinaison 23***		***Combinaison 24***	
10x	1.00	1.00	1.00	1.00	0.00	0.00
20x	1.00	1.00	0.64	0.47	0.00	0.00
22x	1.00	1.00	1.00	1.00	0.00	0.00
23x	0.50	0.02	0.97	1.00	0.00	0.00
24x	0.00	0.00	0.70	0.67	0.00	0.00
25x	1.00	0.81	0.18	0.01	0.00	0.00
26x	0.00	0.00	0.00	0.00	0.00	0.00
30x	0.88	0.64	0.88	0.68	0.00	0.00
	Combinaison 25		***Combinaison 26***		***Combinaison 27***	
10x	0.00	0.00	**1.00**	**1.00**	1.00	1.00
20x	0.00	0.00	**0.89**	**0.60**	0.65	0.47
22x	0.00	0.00	**1.00**	**1.00**	1.00	1.00
23x	0.00	0.00	**1.00**	**1.00**	0.97	1.00
24x	0.00	0.00	**1.00**	**0.70**	0.70	0.67
25x	0.00	0.00	**0.50**	**0.05**	0.16	0.01
26x	0.00	0.00	**0.00**	**0.00**	0.00	0.00
30x	0.00	0.00	**0.95**	**0.87**	0.86	0.68

Tableau 4.11 — Valeurs de *précision* et de *rappel* par famille de tests de la méthode OACAS - AOEI 2007 (Partie 3)

Tests	*Précision*	*Rappel*	**Tests**	*Précision*	*Rappel*
101	1.00	1.00	**238**	1.00	1.00
103	1.00	1.00	**239**	0.91	1.00
104	1.00	1.00	**240**	0.91	1.00
201	0.66	0.15	**241**	1.00	1.00
202	0.78	0.15	**246**	1.00	1.00
203	1.00	1.00	**247**	1.00	1.00
204	1.00	0.95	**248**	1.00	0.10
205	0.91	0.40	**249**	1.00	0.10
206	0.00	0.00	**250**	0.00	0.00
207	0.00	0.00	**251**	0.00	0.00
208	1.00	0.95	**252**	1.00	0.16
209	0.00	0.00	**253**	1.00	0.08
210	0.00	0.00	**254**	0.00	0.00
221	1.00	1.00	**257**	0.00	0.00
222	1.00	1.00	**258**	1.00	0.08
223	1.00	1.00	**259**	1.00	0.08
224	1.00	1.00	**260**	0.00	0.00
225	1.00	1.00	**261**	0.00	0.00
228	1.00	1.00	**262**	0.00	0.00
230	1.00	1.00	**265**	0.00	0.00
231	1.00	1.00	**266**	0.00	0.00
232	1.00	1.00	**301**	0.95	0.83
233	1.00	1.00	**302**	0.96	0.88
236	1.00	1.00	**303**	0.96	0.85
237	1.00	1.00	**304**	0.96	0.95

Tableau 4.12 — Mesures de *précision* et de *rappel* pour la meilleure combinaison (combinaison 26) de mesures de similarité - OAEI 2007

pour la meilleure combinaison obtenue lors des tests déjà effectués sur la base *Benchmark* dans sa version 2008. Pareillement, la combinaison 26 utilise trois différentes mesures de similarités linguistiques, qui sont : LEVENSHTEIN, Q-GRAM et JARO-WINKLER, respectivement pour les noms, les étiquettes et les commentaires.

4.6.2 Étude comparative

Afin d'évaluer les résultats fournis par la méthode OACAS, les Tables 4.15 et 4.16 récapitulent les valeurs des métriques de *précision* et de *rappel*. Elles présentent aussi les valeurs obtenues par les méthodes FALCON-AO (Hu *et al.* (2007)), ONTODNA (Kiu et Lee (2007)) et RIMOM (Li *et al.* (2007)). La méthode OACAS donne de meilleurs résultats *vs.* ces trois méthodes, notamment sur la famille de tests 30x, c-à-d., celle des ontologies réelles. La méthode OACAS fournit de mauvais résultats sur la famille de tests 26x. Cette famille de tests se caractérise par l'absence des noms, des commentaires et des propriétés des entités ontologiques. Ces composants ontologiques, sont des facteurs capitaux pour la détermination de l'alignement par la méthode OACAS. Les résultats expérimentaux, montrent que les performances de la méthode OACAS sont étroitement liées aux caractéristiques des différents composants ontologiques (noms, commentaires et étiquettes). En effet, l'absence de ces descripteurs dégrade considérablement la qualité de l'alignement fourni. Par exemple, les tests 257 et 260 marquent l'absence des noms et des commentaires ainsi que les relations et les propriétés. L'absence des descripteurs noms et commentaires diminue la valeur de la similarité linguistique composée. De même, l'absence des relations et des propriétés dégrade la valeur de la similarité de voisinage. La valeur de la similarité locale (la similarité linguistique composée et celle du voisinage) affecte la valeur de la similarité agrégée. Par conséquent, les valeurs des métriques de *précision* et de *rappel* s'affaiblissent. Pour évaluer les résultats de la méthode OACAS sur la base de *Benchmark 2008* la table 4.17 présente les métriques de *précision* et de *rappel* par famille de tests. Elle présente aussi les valeurs obtenues par la méthode MAPPSO (Bock et Hettenhausen (2008)). Les figures 4.2 et 4.3 illustrent les valeurs de *précision* et de *rappel* de la méthode OACAS en comparaison avec les méthodes FALCON-AO (Hu *et al.* (2007)), ONTODNA (Kiu et Lee (2007)) et RIMOM (Li *et al.* (2007)). Les tests pris en considération dans les deux figures 4.2 et 4.3 sont les tests qui représentent les ontologies réelles. Sur cette famille (*i.e.*, les tests 30x) la méthode OACAS se montre plus compétitive. Ces résultats s'expliquent par le fait que la méthode OACAS s'appuie sur la composition et l'agrégation des similarités. D'une part, ceci explique le choix adéquat

Tests	*Précision*	*Rappel*	**Tests**	*Précision*	*Rappel*
101	1.00	1.00	**228**	1.00	1.00
103	1.00	1.00	**230**	1.00	1.00
104	1.00	1.00	**231**	1.00	0.98
201	1.00	0.02	**232**	1.00	1.00
201-2	1.00	0.78	**233**	1.00	1.00
201-4	1.00	0.60	**236**	1.00	1.00
201-6	1.00	0.42	**237**	1.00	1.00
201-8	1.00	0.22	**238**	1.00	1.00
202	1.00	0.02	**239**	1.00	1.00
202-2	1.00	0.80	**240**	1.00	1.00
202-4	1.00	0.60	**241**	1.00	1.00
202-6	1.00	0.43	**246**	1.00	1.00
202-8	1.00	0.22	**247**	1.00	1.00
203	1.00	1.00	**248**	1.00	0.02
204	1.00	0.90	**248-2**	1.00	0.02
205	1.00	0.28	**248-4**	1.00	0.59
206	1.00	0.34	**248-6**	1.00	0.41
207	1.00	0.32	**248-8**	1.00	0.22
208	1.00	0.91	**249**	1.00	0.02
209	1.00	0.28	**249-2**	1.00	0.79
210	1.00	0.32	**249-4**	1.00	0.59
221	1.00	0.98	**249-6**	1.00	0.44
222	1.00	1.00	**249-8**	1.00	0.22
223	1.00	0.98	**250**	0.00	0.00
224	1.00	1.00	**250-2**	1.00	0.78
225	1.00	0.98	**250-4**	1.00	0.60

Tableau 4.13 — Mesures de *précision* et de *rappel* pour la meilleure combinaison de mesures de similarité - OAEI 2008

Tests	*Précision*	*Rappel*	**Tests**	*Précision*	*Rappel*
250-6	1.00	0.42	**258-4**	1.00	0.61
250-8	1.00	0.24	**258-6**	1.00	0.41
251	1.00	0.02	**258-8**	1.00	0.23
251-2	1.00	0.78	**259**	1.00	0.02
251-4	1.00	0.61	**259-2**	1.00	0.81
251-6	1.00	0.41	**259-4**	1.00	0.79
251-8	1.00	0.23	**259-6**	1.00	0.79
252	1.00	0.02	**259-8**	1.00	0.80
252-2	1.00	0.80	**260**	0.00	0.00
252-4	1.00	0.79	**260-2**	1.00	0.79
252-6	1.00	0.79	**260-4**	1.00	0.62
252-8	1.00	0.79	**260-6**	1.00	0.41
253	1.00	0.02	**260-8**	1.00	0.24
253-2	1.00	0.81	**261**	0.00	0.00
253-4	1.00	0.59	**261-2**	1.00	0.78
253-6	1.00	0.45	**261-4**	1.00	0.78
253-8	1.00	0.22	**261-6**	1.00	0.78
254	0.00	0.00	**261-8**	1.00	0.78
254-2	1.00	0.78	**262**	0.00	0.00
254-4	1.00	0.60	**262-2**	1.00	0.78
254-6	1.00	0.42	**262-4**	1.00	0.60
254-8	1.00	0.24	**262-6**	1.00	0.42
257	0.00	0.00	**262-8**	1.00	0.24
257-2	1.00	0.78	**265**	0.00	0.00
257-4	1.00	0.60	**266**	0.00	0.00
257-6	1.00	0.42	**301**	1.00	0.34
257-8	1.00	0.21	**302**	0.00	0.00
258	1.00	0.02	**303**	0.00	0.00
258-2	1.00	0.80	**304**	1.00	0.98

Tableau 4.14 — Mesures de *précision* et de *rappel* pour la meilleure combinaison de mesures de similarité - OAEI 2008 (Suite)

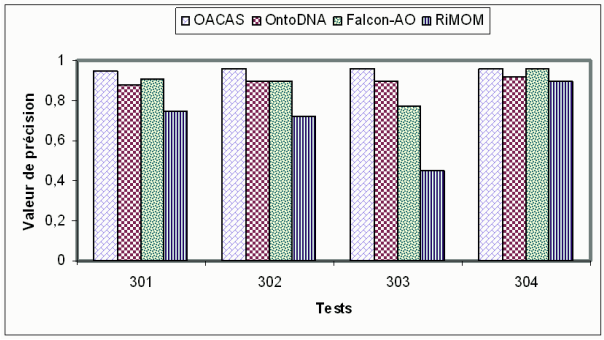


Figure 4.2 — Valeurs de précision de la méthode OACAS par rapport aux autres méthodes sur les ontologies réelles

de la mesure de similarité qui s'adapte le mieux à un descripteur donné. D'autre part, ça s'explique par une bonne exploitation des aspects textuels et structurels de la méthode OACAS. Sur la base de *Benchmark 2008*, la méthode OACAS a donné de bons résultats, notamment, sur la famille des tests (26x-x) qui ont porté comme nouveauté la restitution de quelques informations textuelles (*i.e.*, les étiquettes des entités ontologiques).

La table 4.17 récapitule les valeurs moyennes de *précision* et de *rappel* de la méthode OACAS en comparaison avec la méthode MapPSO. Les performances de la méthode OACAS dépassent celles de MapPSO, notamment sur les familles de tests 25x, 26x et 30x comme le montrent les figures 4.4 et 4.5.

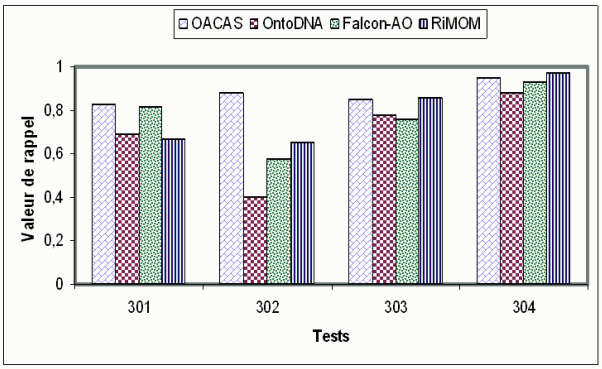


Figure 4.3 — Valeurs de rappel de la méthode OACAS par rapport aux autres méthodes sur les ontologies réelles

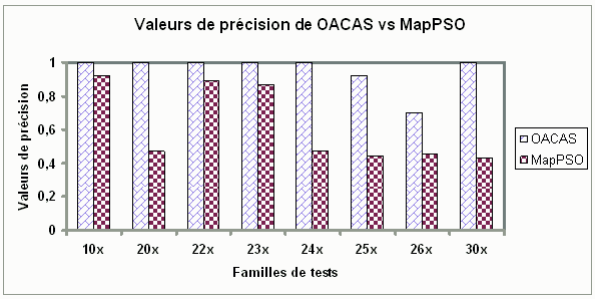


Figure 4.4 — Valeurs de précision de la méthode OACAS par rapport à MapPSO par famille de tests

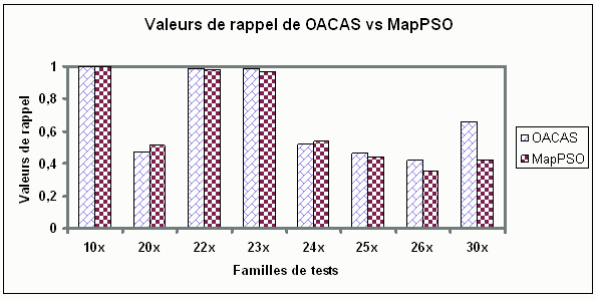


Figure 4.5 — Valeurs de rappel de la méthode OACAS par rapport à MapPSO par famille de tests

4.7 Conclusion

Le présent chapitre a décrit une évaluation et une expérimentation des méthodes EDOLA, SODA et OACAS sur les bases *Benchmark*. L'évaluation a consisté en une étude comparative des méthodes proposées avec d'autres méthodes appartenant à la littérature. Les résultats obtenus sont encourageants et satisfaisants. En effet, les méthodes EDOLA, SODA et OACAS ont obtenu de meilleurs résultats en particulier sur les tests réels et dans les tests où les deux ontologies possèdaient des structures semblables ou identiques. Les résultats ont été moins bons sur les jeux de test qui comportaient des ontologies dégradées.

	OACAS		OntoDNA		Falcon-AO		RiMOM	
Tests	*Précision*	*Rappel*	*Précision*	*Rappel*	*Précision*	*Rappel*	*Précision*	*Rappel*
101	1.00	1.00	0.94	1.00	1.00	1.00	1.00	1.00
103	1.00	1.00	0.94	1.00	1.00	1.00	1.00	1.00
104	1.00	1.00	0.94	1.00	1.00	1.00	1.00	1.00
201	0.66	0.15	0.11	0.01	1.00	0.95	1.00	1.00
202	0.78	0.15	0.11	0.11	0.87	0.87	1.00	0.80
203	1.00	1.00	0.94	1.00	1.00	1.00	1.00	0.88
204	1.00	0.95	0.93	0.84	0.98	0.98	1.00	1.00
205	0.91	0.40	0.57	0.12	1.00	0.98	1.00	0.99
206	0.00	0.00	0.69	0.23	1.00	0.93	1.00	0.99
207	0.00	0.00	0.69	0.23	0.98	0.91	1.00	0.99
208	1.00	0.95	0.93	0.84	1.00	1.00	0.98	0.86
209	0.00	0.00	0.57	0.12	0.79	0.78	1.00	0.84
210	0.00	0.00	0.69	0.23	0.81	0.80	0.99	0.85
221	1.00	1.00	0.93	0.76	1.00	1.00	1.00	1.00
222	1.00	1.00	0.94	1.00	1.00	1.00	1.00	1.00
223	1.00	1.00	0.94	1.00	1.00	1.00	1.00	1.00
224	1.00	1.00	0.94	1.00	1.00	0.99	1.00	0.99
225	1.00	1.00	0.94	1.00	1.00	1.00	1.00	1.00
228	1.00	1.00	0.53	0.27	1.00	1.00	1.00	1.00
230	1.00	1.00	0.91	1.00	0.94	1.00	0.94	1.00
231	1.00	1.00	0.94	1.00	1.00	1.00	1.00	1.00
232	1.00	1.00	0.93	0.76	1.00	0.99	1.00	0.99
233	1.00	1.00	0.53	0.27	1.00	1.00	1.00	1.00
236	1.00	1.00	0.53	0.27	1.00	1.00	1.00	1.00
237	1.00	1.00	0.94	1.00	1.00	0.99	1.00	0.99
238	1.00	1.00	0.94	1.00	1.00	0.99	1.00	0.99
239	0.91	1.00	0.50	0.31	1.00	1.00	1.00	1.00
240	1.00	1.00	0.50	0.27	1.00	1.00	1.00	1.00
241	1.00	1.00	0.53	0.27	1.00	1.00	1.00	1.00
246	1.00	1.00	0.50	0.31	1.00	1.00	1.00	1.00

Tableau 4.15 — Comparatif des mesures de *précision* et de *rappel* pour la meilleure combinaison de mesures de similarité - OAEI 2007 (Partie 1)

	OACAS		OntoDNA		Falcon-AO		RiMOM	
Tests	*Précision*	*Rappel*	*Précision*	*Rappel*	*Précision*	*Rappel*	*Précision*	*Rappel*
247	1.00	1.00	0.50	0.27	1.00	1.00	1.00	1.00
248	1.00	0.10	0.11	0.01	0.85	0.84	0.99	0.78
249	1.00	0.10	0.11	0.01	0.87	0.87	1.00	0.79
250	0.00	0.00	0.00	0.00	1.00	0.27	1.00	0.55
251	0.00	0.00	0.11	0.01	0.56	0.56	0.76	0.58
252	1.00	0.16	0.11	0.01	0.71	0.71	0.85	0.70
253	1.00	0.08	0.11	0.01	0.85	0.84	0.99	0.77
254	0.00	0.00	0.00	0.00	1.00	0.27	1.00	0.27
257	0.00	0.00	0.00	0.00	1.00	0.27	1.00	0.55
258	1.00	0.08	0.11	0.01	0.54	0.54	0.76	0.57
259	1.00	0.08	0.11	0.01	0.70	0.70	0.85	0.69
260	0.00	0.00	0.00	0.00	1.00	0.31	0.93	0.45
261	0.00	0.00	0.00	0.00	0.89	0.24	1.00	0.27
262	0.00	0.00	0.00	0.00	1.00	0.27	1.00	0.27
265	0.00	0.00	0.00	0.00	1.00	0.31	0.93	0.45
266	0.00	0.00	0.00	0.00	0.89	0.24	1.00	0.27
301	0.95	0.83	0.88	0.69	0.91	0.82	0.75	0.67
302	0.96	0.88	0.90	0.40	0.90	0.58	0.72	0.65
303	0.96	0.85	0.90	0.78	0.77	0.76	0.45	0.86
304	0.96	0.95	0.92	0.88	0.96	0.93	0.90	0.97

Tableau 4.16 — Comparatif des mesures de *précision* et de rappel pour la meilleure combinaison de mesures de similarité - OAEI 2007 (Partie 2)

	OACAS		**MapPSO**	
Tests	*Précision*	*Rappel*	*Précision*	*Rappel*
10x	1.00	1.00	0.92	1.00
20x	1.00	0.47	0.47	0.51
22x	1.00	0.99	0.89	0.98
23x	1.00	0.99	0.87	0.97
24x	1.00	0.52	0.47	0.54
25x	0.92	0.46	0.44	0.44
26x	0.70	0.42	0.45	0.35
30x	1.00	0.66	0.43	0.42

Tableau 4.17 — Valeurs moyennes de *précision* et de *rappel*, de OACAS et MapPSO - OAEI 2008

Conclusion générale

Les travaux qui ont été menés dans cadre de cette thèse se situent dans l'intersection de domaines à savoir : l'ingénierie des connaissances et du Web sémantique. L'objectif est de proposer des méthodes d'alignement d'ontologies OWL. Ces méthodes permettent s'assurer une interopérabilité sémantique des connaissances contenues dans les ontologies à aligner.

Le résultat des travaux menés est la proposition de trois méthodes d'alignement des ontologies décrites en OWL. Ces méthodes exploitent des mesures de similarité et les structures internes des ontologies à aligner. Ces méthodes reposent sur une technique d'agrégation et de composition de différentes mesures de similarité.

La première méthode EDOLA aligne des ontologies OWL-Lite. Elle opère en deux étapes successives (Zghal *et al.* (2007e), Zghal *et al.* (2007d)). La première étape permet de calculer la similarité locale par l'intermédiaire des mesures de similarité linguistique ou terminologique. La deuxième étape determine la similarité globale en exploitant les diamètres des entités à aligner et le processus de propagation de la similarité à travers le voisinage.

La seconde méthode SODA assure l'alignement des ontologies OWL-DL (Zghal *et al.* (2007c)). Cette méthode suit un processus sur trois phases. La première phase calcule la similarité liguistique par l'intermédiaire de la mesure de Jaro-Winkler et Monge-Elkan. Dans la seconde phase, les structures de premier niveau des entités ontologiques sont exploitées pour déterminer la similarité structurelle en calculant des mesures de similarité par groupe. La troisième phase est une étape de composition des similarités précédemment déterminées pour fournir la similarité sémantique et résultat de l'alignement.

La troisième méthode OACAS (Zghal *et al.* (2009)) est une extension des deux méthodes EDOLA et SODA. La méthode propose de choisir une composition de similarité pour le calcul de la similarité linguistique composée. Cette composition dépend de la nature des

informations contenues dans les entités à aligner. Les valeurs des mesures de similarité linguistique composée sont exploitées pour la détermination de la similarité du voisinage. En effet, la similarité du voisinage exploite la structure des entités sur deux niveaux. Cette exploration étendue assure une meilleure prise en charge de la sémantique incorporée dans les voisins. La similarité agrégée est une mesure permettant d'apprécier les degrés de ressemblance des entités à aligner.

Les trois méthodes ont été évaluées en exploitant les bases de test *Benchmark* de OAEI à travers les métriques d'évaluation. Ces derniers permettent de mesurer la qualité des alignements obtenus. Elles assurent aussi une comparaison des méthodes proposées par rapport à d'autres appartenant à la littérature dédiée. Les résultats obtenus sont satisfaisants et encourageants. Ils sont dans certains cas plus performants, en particulier sur les tests où les deux ontologies sont semblables et dans le contexte des ontologies réelles. Les valeurs des métriques d'évaluation restent perfectibles dans le cas où les deux ontologies à aligner ne possèdent pas la même structure. Dans ce cas, les voisinages des entités à aligner ne possèdent pas les mêmes structures.

Perspectives

Les perspectives de cette thèse portent sur deux volets. Le premier volet inclut des travaux futurs visant l'amélioration des méthodes proposées dans le cadre de cette thèse. Le second volet s'intéresse au développement des méthodes d'alignement multilingues. Les travaux futurs pour l'amélioration des processus d'alignement dans les méthodes proposées comportent :

- **L'exploration des connaissances textuelles extraites** : le processus de calcul des valeurs de similarité repose sur l'exploitation des chaînes de caractère décrivant les différentes entités contenues dans les ontologies à aligner. L'analyse du contenu des ontologies à aligner montre qu'elles contiennent des informations très pertinentes et en particulier ceux qui appartiennent aux commentaires. Ces données sont d'une grande utilité pour les travaux futurs. La réalisation de cette extraction de connaissances se fait sur deux phases :

  1. *Extraction de relations taxonomiques par l'intermédiaire des caractéristiques lexico-syntaxique* : cette extraction a pour objectif principal l'identification des relations taxonomiques, par exemple *hyponymie*, *hyperonyumie*, *synonymie*, etc. Ces caractéristiques permettent la définition de liens entre les concepts d'une on-

tologie source et une ontologie destination. Elles enrichissent l'ontologie source par la découverte de nouveaux concepts. Ces dernières peuvent être alignées avec d'autres concepts de l'ontologie destination.

2. *Extraction des relations non taxonomiques par l'intermédiaire de la découverte des relations associatives* (Ben Yahia *et al.* (2009)) : deux types de règles d'association sont retenues, *Règle 1* : SN1 ⇒ SN2 et *Règle 2* : (SN2, SV) ⇒ SN3, où SN est un syntagme nominal et SV est syntagme verbal. La *Règle 1* permet de déterminer un syntagme verbal qui se présente fréquemment avec les syntagmes nominaux SN1 et SN2. Ce syntagme verbal constitue le nom du lien reliant les deux concepts représentés par SN1 et SN2. Dans la *Règle 2*, le syntagme verbal SV est le label de la relation assurant le lien entre les deux concepts représentés, respectivement, par deux syntagmes nominaux SN2 et SN3.

– **Le passage à l'échelle** : de plus en plus de domaines représentent leurs connaissances à travers des ontologies réelles de grande tailles. Ces ontologies appartiennent aux domaines de la médecine, l'agronomie, la biologie, etc. Elles contiennent des milliers de concepts et de relations entre ces derniers. Par conséquent, le passage à l'échelle du processus d'alignement est d'une très grande importance. Le partitionnement des ontologies de grandes taille s'avère une condition *"sine qua non"* pour atteindre l'objectif. Afin de réaliser l'objectif, une méthode hybride de partitionnement serait à introduire :

  1. *Bi-partionnement structurel de proximité* : ce bi-partionnement constitue une première étape de subdivision (Hu *et al.* (2006)). Il fournit un ensemble de segments. Le bi-partionnement structurel de proximité est une démarche avec perte d'information mais elle permet d'obtenir des sous-ensembles d'ontologies pouvant être alignés facilement.
  2. *Analyse sémantique et extraction transversale* : Dans ce contexte, il est nécessaire de compléter les informations sémantiques de toutes les entités appartenant aux clusters ou groupes (par exemple voisinage, restrictions, etc.). Il est aussi recommandé d'effectuer une opération de balisage des frontières des classes.

L'alignement d'ontologies multilingues est un domaine de continuité de travaux futurs de cette thèse. Le problème de la prise en charge du multi-linguisme, dans domaine de la recherche d'information et la gestion du contenu, permet le raisonnement sur l'intersection des contextes des domaines. La recherche s'intéresse à l'alignement des ontologies monolangues, par contre, l'alignement des ontologies décrites dans différents langages naturels

est pratiquement non exploré. Jung *et al.* (2009) ont focalisé leurs travaux sur l'alignement indirect en se basant sur l'utilisation directe entre les ontologies. L'aspect multilingue sera introduit dans les travaux futurs par l'intermédiaire des règles associatives. Ces règles construisent des bases de connaissances à partir de corpus parallèles. Ces corpus sont utilisés lors du processus d'alignement d'ontologies.

Bibliographie

Angel, E. et Bellman, R. (1972). *Dynamic Programming and Partial Differential Equations*. Academic Press, New York.

Bach, T., Dieng-Kuntz, R. et Gandon, F. (2004). On ontology matching problems - for building a corporate semantic web in a multi-communities organization. *In Proceedings of ICEIS*, pages 236–243, Porto, Portugal.

Bach, T. L. (2006). *Construction d'un web sémantique multi-points de vues.* Thèse de doctorat, École des Mines à Sophia Antipolis, France.

Bachimont, B. (1999). *L'intelligence artificielle comme écrit dynamique : de la raison graphique à la raison conceputationnelle.* Grasset, Paris.

Bachimont, B. (2000). *Engagement Sémantique et Engagement Ontologique : Conception et Réalisation D'ontologies En Ingénierie Des Connaissances*, chapitre 19, pages 305–324. T. Charlet, M. Zacklad, G. Kassel and D. Bourigault (Édition), *Ingénierie des connaissances, évolutions récentes et nouveaux défis.*

Batini, *C.*, Lenzerini, *M. et* Navathe, *S. (1986). A comparative analysis of methodologies for database schema integration.* ACM Computing Surveys, *18(4):323–264.*

Ben Yahia, *S.*, Gasmi, *G. et* Mephu Nguifo, *E. (2009). A new generic basis of "factual" and "implicative" association rules.* Intelligent Data Analysis, *13(4):633–656.*

Benerecetti, *M.*, Bouquet, *P. et* Ghidini, *C. (2001). Contextual reasoning distilled.* Journal of Experimental and Theoretical Artificial Intelligence, *12(3):279–305.*

Berners-Lee, *T.*, Hendler, *J. et* Lassila, *O. (2001). The semantic web.* Scientific American.

Bock, *J. et* Hettenhausen, *J. (2008).* MapPSO *results for OAEI 2008.* In Proceedings of the 3^{rd} International Workshop on Ontology Matching (OM-2008), *pages 176–181, Karlsruhe, Germany.*

Bouaud, *J.*, Bachimont, *B.*, Charlet, *J. et* Zweigenbaum, *P. (1995). Methodological principles for structuring an ontology.* In Proceedings of IJCAI'95 Workshop on Basic Ontological Issues in Knowledge sharing, *Montrééal, Canada.*

Bouquet, *P.*, Giunchiglia, *F.*, van Harmelen, *F.*, Serafini, *L. et* Stuckjenschmidt, *H. (2003). C-owl - contextualizing ontologies.* In Proceedings of the 2^{nd} International Semantic Web conference (ISWC), *volume 2870 of LNCS Springer-Verlag, pages 164–179, Sanibel Island, US.*

Castano, *S.*, Ferrarra, *A. et* Montanelli, *S. (2006). Matching ontologies in open networked systems : techniques and applications.* Journal on Data Semantics, *V:25–63.*

Chandrasekaran, *B.*, Josephson, *J. et* Benjamins, *V. (1999). What are ontologies, and why do we need them ?* IEEE Intelligent Systems, *14(1):20–26.*

Charlet, *J. (2001). Ingénierie des connaissances : un domaine scientifique, un enseignement ?* In Proceedings des journées Francophones d'Ingénierie des Connaissances (IC'2001), *pages 233–252, Grenoble, France.*

Cocchiarella, *N. (1991). Ontology II : Formal ontology.* In Burkhardt, *H. et* Smith, *B., éditeurs :* Handbook of Metaphysics and Ontology, *pages 640–641. Philosophia Verlag, Munich.*

Connolly, *D.*, Harmelen, *F. V.*, Horrocks, *I.*, McGuinness, *D. L.*, Patel-Schneider, *P. F. et* Stein, *L. A. (2001).* DAML+OIL *: Reference Description. Rapport technique,* W3C *: World Wide Web Consortium, http ://www.w3.org/TR/2001/NOTE-daml+oil-reference-20011218.*

Cormen, *T.*, Leiserson, *C.*, Rivest, *R. et* Stein, *C. (2001).* Introduction to Algorithms, 2^{nd} Edition. *The MIT Press.*

Desprès, *S. et* Szulman, *S. (2007). Merging of legal micro-ontologies from european directives.* Artificial Intelligence Law, *15(2):187–200.*

Do, *H.*, Melnik, *S. et* Rahm, *E. (2002). Comparison of schema matching evaluations.* In Proceedings of the 2^{nd} Int. Workshop on Web Databasess, *pages 221–237, Erfurt, Germany. German Informatics Society.*

Do, *H. H. et* Rahm, *E. (2002).* COMA - *a system for fexible combination of schema matching approaches.* In Proceedings of the 28^{th} International Conference on Very Large Data Base (VLDB)), *pages 610–621, Hong Kong, China.*

Doan, *A.*, Madhavan, *J.*, Domingos, *P. et* Halevy, *A. (2002). Learning to map between ontologies on the semantic web.* In Proceedings of the 11^{th} International WWW Conference, *Hawaii, US.*

Ehrig, *M. et* Staab, *S. (2004).* QOM *: Quick Ontology Mapping.* In Proceedings of the 3^{rd} International Semantic Web Conference (ISWC), *pages 683–697, Hiroshima, Japan.*

Euzenat, *J. (2001). Towards a principled approach to semantic interoperability.* In Proceedings of the IJCAI Workshop on Ontology and Information Sharing, *pages 19–25, Seattle, US.*

Euzenat, *J. (2007). Semantic precision and recall for ontology alignment evaluation.* In Proceedings of the 20^{th} International Joint Conference on Artificial Intelligence (IJCAI'2007), *pages 348–353, Hyderabad, India.*

Euzenat, *J.*, Bach, *T. L.*, Barrasa, *J.*, Bouquet, *P.*, Bo, *J. D.*, Dieng, *R.*, Ehrig, *M.*, Lara, *R.*, Maynard, *D.*, Napoli, *A.*, Starmou, *G.*, Stuckenschmidt, *H.*, Shvaiko, *P.*, Tessaris, *S.*, Acker, *S. V. et* Zaihrayeu, *I. (2004). State of art on ontology alignment. Rapport technique KWEB/2004/D2.2.3/v1.2, Knowledge Web Consortium.*

Euzenat, *J. et* Shvaiko, *P. (2007).* Ontology Matching. *Springer-Verlag, Heidelberg (DE).*

Euzenat, *J. et* Valtchev, *P. (2004). Similarity-based ontology alignement in* OWL-*Lite.* In Proceedings of the 15^{th} European conference on Artificial Intelligence (ECAI), *pages 333–337, Valence, Espagne.*

Giunchiglia, *F.*, Shvaiko, *P. et* Yatskevich, *M. (2003). S-match : an algorithm and an implementation of semantic matching.* In Proceedings of the 1^{st} European

Semantic Web Symposium (ESWS), *volume 3053 of LNCS Springer-Verlag, pages 61–75, Hersounisous, Germany.*

Goclenius, *R. (1613).* Lexicon philosophicum.

Gómez-Pérez, *A.*, Ferández-López, *M. et* De Vicente, *A. (1996). Towards a method to conceptualize domain ontologies.* In Proceedings of the Workshop on Ontological Engineering, held in conjunction with European Conference on Artificial Intelligence (ECAI'96), *pages 41–52, Budapest, Hungary.*

Gómez-Pérez, *A.*, Fernández-López, *M. et* Corcho, *O. (2004).* Ontological Engineering. *Springer, Heidelberg.*

Gruber, *T. (1993). A translation approach to portable ontology specifications.* Knowledge Acquisition, *5(2):199–220.*

Gruber, *T. (2009).* Encyclopedia of Database Systems, *chapitre Ontology, pages 1963–1965. Springer-Verlag.*

Gruber, *T. et* Olsen, *G. (1994). An Ontology for Engineering Mathematics.* In Proceedings of the 4^{th} International Conference on Principles of Knowledge Representation and Reasoning, *pages 258–269, Bonn, Germany. Gustav Stresemann Institut, Morgan Kaufmann.*

Gruber, *T. R. (1995). Towards principles for the design of ontologies used knowledge sharing.* International Journal of Human-Computer Studies, *43:907–928.*

Guarino, *N. (1997). Some organizing principles for a unified top-level ontology.* In Proceedings of AAAI Spring Symposium on Ontological Engineering.

Guarino, *N. (1998). Formal ontology and information systems.* In Proceedings of the International Conference of Formal Ontology in Information Systems, *pages 3–15, Trento, Italy.*

Guarino, *N.*, Carrara, *M. et* Giaretta, *P. (1994a). Formalizing ontological commitments.* In Proceedings of National Conference on Artificial Intelligence (AAAI-94), Seattle, Morgan Kaufmann, *pages 560–567, Seattle, WA.*

Guarino, *N.*, Carrara, *M. et* Giaretta, *P. (1994b). An ontology of meta-level categories.* In Proceedings of the 4^{th} International Conference on Principles of Knowledge Representation and Reasoning, *pages 270–280, San Mateo, CA. Morgan Kaufmann.*

GUARINO, *N. et* GIARETTA, *P. (1995). Ontologies and knowledge bases : Towards a terminological clarification.* In MARS, *N., éditeur :* Towards very large knowledge bases : knowledge building and knowledge sharing, *pages 25–32. IOS Press, Amsterdam.*

GUARINO, *N. et* POLI, *R. (1995). Formal ontology in conceptual analysis and knowledge representation.* Special issue of the International Journal of Human and Computer Studies, *43(5/6):625–640.*

GUARINO, *N. et* WELTY, *C. (2000). A formal ontology of properties.* In Proceedings of the 12^{th} European Workshop on Knowledge Acquisition, Modeling and Management (EKAW'00), *pages 97–12, London, United Kingdom.*

HAFNER, *C. et* FRIDMAN, *N. (1996). Ontological foundations for biology knowledge models.* In Proceedings of the 4^{th} International Conference On Intelligent Systems for Molecular Biology, *pages 78–87.*

HAMMING, *R. (1950). Error detecting and error correcting codes.* Bell System Technical Journal, *26(2):147–160.*

HORROCKS, *I.*, PATEL-SCHNEIDER, *P. F. et* VAN HARMELEN, *F. (2003). From SHIQ and RDF to OWL : The making of a web ontology langage.* Journal of Web Semantics, *1(1):7–26.*

HOSHIAI, *T.*, YAMANE, *Y.*, NAKAMURA, *D. et* TSUDA, *H. (2004). A semantic category matching approach to ontologies alignment.* In Proceedings of the 3rd international workshop on Evaluation of Ontology Based Tools (EON 2004), *Hiroshima, Japan.*

HU, *W.*, JIAN, *N.*, QU, *Y. et* WANG, *Q. (2005). GMO : A graph matching for ontologies.* In Proceedings of K-CAP Workshop on Integrating Ontologies, *pages 43–50, Banff, Canada.*

HU, *W.*, ZHAO, *Y.*, LI, *D.*, CHENG, *G.*, WU, *H. et* QU, *Y. (2007). Falcon-AO : results for OAEI 2007.* In Proceedings of the 2^{nd} International Workshop on Ontology Matching (OM-2007), *pages 170–178, Busan, Korea.*

HU, *W.*, ZHAO, *Y. et* QU, *Y. (2006). Partition-based block matching of large class hierarchies.* In Proceedings of the 1^{st} Asian Semantic Web Conference - ASWC 2006, *pages 72–83.*

ICHISE, *R.*, HAMASAKI, *M. et* TAKEDA, *H. (2002). Discovering relationships among catalogs.* In Proceedings of the 7^{th} International Conference on Discovery Science (CAiSE), *volume 3245 of LNCS Springer-Verlag, pages 371–379, Padova, Italy.*

JARO, *M. (1989). Advanced in record-linkagemethodology as applied to matching the 1985 census of Tampa, Floarida.* Journal of the american Statistical Association, *84(406): 414–420.*

JUNG, *J. J.*, HÅKANSSON, *A. et* HARTUNG, *R. (2009). Indirect alignment between multilingual ontologies : A case study of korean and swedish ontologies.* In Agent and Multi-Agent Systems : Technologies and Applications, *volume 5559 de* LNCS, *pages 233–241.*

KALGOFLOU, *Y. et* SCHORLEMMER, *M. (2003). Ontology mapping : the state of the art.* The Knowledge Engineering Review, *18(1):1–31.*

KASSEL, *G. (2002). OntoSpec : une méthode de spécification semi-formelle d'ontologies.* In Actes des journées francophones d'Ingénierie des Connaissances (IC'2002), *pages 75–87, Rouen, France.*

KASSEL, *G. et* PERPETTE, *S. (1999). Cooperative ontology construction needs to carefully articulate terms, notions and objects.* In Proceedings of the International Workshop on Ontology Engineering on the global Information Infrastructure, *pages 97–12, London, United Kindom.*

KAYSER, *D. (1997).* La représentation des connaissances. *Hermès.*

KIU, *C. et* LEE, *C. (2007). OntoDNA : ontology alignment results for OAEI 2007.* In Proceedings of the 2^{nd} International Workshop on Ontology Matching (OM-2007), *pages 227–235, Busan, Korea.*

KLEIN, *M. (2001). Combining and relating ontologies : an analysis of problems and solutions.* In Proceedings of the IJCAI-Workshop on Ontologies and Information Sharing, *pages 53–62, Seattle, US.*

KLYNE, *G. et* CARROLL, *J. J. (2004). Resource Description Framework (*RDF*) : Concepts and Abstract Syntax. Rapport technique,* W3C *: World Wide Web Consortium, http ://www.w3.org/TR/2004/REC-rdf-concepts-20040210/ (05/24/2010).*

KORTIS, *K.*, VOUROS, *G. et* STERGIOU, *K. (2006). Towards automatic merging of domain ontologies :pproach the hcone-merge a.* Journal of Web Semantics, *4(1):60–79.*

LACHER, *M. et* GROH, *G. (2001). Facilitating the exchange of explicit knowledge through ontology mapping.* In Proceedings of the 14^{th} International FLAIRS Conference, *Florida, US.*

LASSILA, *O. et* MCGUINNESS, *D. (2001). The role of frame-based representation on the semantic web. Technical Report KSL-01-02, Knowledge Systems Laboratory, Stanford University, Stanford, California, United States.*

LEVENSHTEIN, *I. V. (1966). Binary codes capables of corrections, deletions, insertions and reversals.* Soviet Physics-Doklady, *10(8):707–710.*

LI, *Y.*, ZHONG, *Q.*, LI, *J. et* TANG, *J. (2007). Result of ontology alignment with RiMOM at OAEI'07.* In Proceedings of the 2^{nd} International Workshop on Ontology Matching (OM-2007), *pages 196–205, Busan, Korea.*

LORHARD, *J. (1606).* Ogdoas Scholastica, continens Diagraphen Typicam artium : Grammatices (Latinae, Graecae), Logices, Rhetorices, Astronomices, Ethices, Physices, Metaphysices, seu Ontologiae. *St. Gallen.*

MADHAVAN, *J.*, BERNTEIN, *P. et* RAHM, *E. (2001). Generic schema matching with Cupid.* In Proceedings of the 27^{th} International Conference on Very Large Data Base (VLDB)), *pages 48–58, Roma, Italy.*

MAEDCHE, *A.*, SCHNURR, *H. P.*, STAAB, *S. et* STUDER, *R. (2000). Representation language-neutral modeling of ontologies.* In Proceedings of the German Workshop Modellierung.

MARSH, *J. (2001).* XML *base. Rapport technique,* W3C *: World Wide Web Consortium, http ://www.w3.org/TR/2001/REC-xmlbase-20010627/.*

MCCARTHY, *J. et* HAYES, *P. (1969). Some philosophical problems from the standpoint of artificial intelligence.* In Machine Intelligence, *volume 4, pages 463–502.*

MELNIK, *S.*, GARCIA-MOLINA, *H. et* RAHM, *E. (2002). Similarity flooging : a versatile graph matching algorithm.* In Proceedings of the 18^{th} International Conference on Data Engineering (ICDE), *pages 117–128, San Jose, US.*

MHIRI, *M.*, GARGOURI, *F. et* BENSLIMANE, *D. (2006). Détermination automatique des relations sémantiques entre les concepts d'une ontologie.* In Proceedings of the XXIVème édition du Congrès INFORSID, *pages 627–642, Hammamet, Tunisie.*

MILLER, *G. A. (1995).* WORDNET *: a Lexical Database for English.* Communications of the ACM, *38(11):39–41.*

MITRA, *P.*, NOY, *N. et* JAISWAL, *A. (2005). Ontology mapping discovery with uncertainty.* In Proceedings of the 4^{th} International Semantic Web Conference (ISWC), *volume 3729 of LNCS Springer-Verlag, pages 537–547, Galway, IE.*

MIZOGUCHI, *R. (1998). A step towards ontological engineering.* In Proceedings of 12^{th} National Conference on Artificial Intelligence of Japanese Society for Artificial Intelligence, *pages 24–31.*

MIZOGUCHI, *R.*, VANWELKENHUYSEN, *J. et* IKEDA, *M. (1995). Task ontology of reuse of problem solving knowledge.* In Towards Very Large Knowledge Bases : Knowledge Building and Knowledge Sharing, *pages 49–59, Enschede, Netherland.*

MONGE, *A. et* ELKAN, *C. (1996a). The field-matching problem : algorithm and applications.* In Proceedings of the 2^{nd} International Conference on Knowledge Discovery and Data Mining, *pages 267–270, London, UK. Springer-Verlag.*

MONGE, *A. et* ELKAN, *C. (1996b). The field-matching problem : algorithm and applications.* In Proceedings of the 2^{nd} International Conference on Knowledge Discovery and Data Mining, *pages 267–270.*

NOTTELMANN, *H. et* STRACCIA, *U. (2005). splmap ; a probabilistic approacg=h to schema matching.* In Proceedings of the 27^{th} European Conference on Information Retrieval Research (ECIR), *pages 81–95, Santiago de Compostela, Spain.*

NOTTELMANN, *H. et* STRACCIA, *U. (2006). A probabilistic framework for automated web discovery alignment.* In Soft computing in ontologies and the semantic Web, volume 204 of Studies in fuzziness and soft computing, *pages 47–77. Springer-Verlag.*

NOY, *N. et* HAFNER, *C. (1997). The state of the art in ontology design : A survey and comparative review.* AI Magazine, *18(3):53–74.*

NOY, *N. et* MUSEN, *M. (2001). Anchor*-PROMPT *: Using non-local context for semantic matching.* In Proceedings of the IJCAI-Workshop on Ontologies and Information Sharing, *pages 63–70, Seattle, US.*

PAN, *R.*, DING, *Z.*, YU, *Y. et* PENG, *Y. (2005). A bayesian network approach to ontology mapping.* In Proceedings of the 18^{th} International Semantic Web Conference (ISWC), *volume 2342 of LNCS Springer-Verlag, pages 333–347, Chia Laguna, Italy.*

QU, *Y.*, HU, *W. et* CHEN, *G. (2006). Constructing virtual documents for ontology matching.* In Proceedings of the 15^{th} International World Wide Web Conference (WWW), *pages 23–31, Trondheim, NO.*

RAHM, *E. et* BERNSTEIN, *P. (2001). A survey of approaches to automatic schema matching.* VLDB Journal, *10(4):334–350.*

RECTOR, *A. (1999). Clinical terminology : Why is it so hard ?* Methods of Information in Medicine, *38:239–252.*

SHVAIKO, *P. et* EUZENAT, *J. (2005). A survey of schema-based matching approaches.* Journal on Data Semantics, *IV:146–171.*

SMITH, *M. K.*, WELTY, *C. et* MCGUINNESS, *D. L. (2004).* OWL *: Ontology Web Language Guide. Rapport technique,* W3C *: World Wide Web Consortium, http ://www.w3.org/TR/2004/REC-owl-guide-20040210/.*

SOWA, *J. (1999).* Knowledge Representation : Logical, Philosophical and Computational Foundations. *Brooks Cole Publishing Co., Pacific Grove, CA.*

SOWA, *J. (2000). Ontology, metadata, and semiotics.* In Proceedings of the Linguistic on Conceptual Structures (ICCS'2000), *pages 55–81, London, UK. Springer-Verlag.*

STRACCIA, *U. et* TRONCY, *R. (2005).* OMPA *: Combining classifiers for aligning automatically OWL ontologies.* In Proceedings of the 6^{th} International Conference on Web Information System (WISE), *pages 133–147, Budva, ME.*

TANG, *J.*, LI, *J.*, LIANG, *B.*, HUANG, *X.*, LI, *Y. et* WANG, *K. (2006). Using bayesian decision for ontology mapping.* Journal of Web Semantics, *4(1):234–262.*

TEULIET, *R. et* GIRARD, *N. (2001). Des connaissances pour l'action dans les organisations : quelle ingénierie des connaissances pour assister l'activitéé ?* In Proceedings

des journées Francophones d'Ingénierie des Connaissances (IC'2001), *pages 253–272, Grenoble, France.*

Ukkonen, *E. (1992). Approximate string-matching with* q-grams *and maximal matchs.* Theoretical Computer Science, *92(1):191–211.*

Uschold, *M. et* Gruninger, *M. (1996). Ontologies : Principles, methods and applications.* Knowledge Engineering Review, *11(2):93–136.*

Uschold, *M. et* King, *M. (1995). Towards a methodology for building ontologies.* In Proceeding of the Workshop on Basic Ontological Issues in Knowledge Sharing, held in conjunction with IJCAI-95, *pages 1–12, Montréal, Canada.*

Valtchev, *P. (1999).* Construction automatique de taxonomies pour l'aide à la représentation de connaissance par objets. *Thèse de doctorat, Université de Grenoble 1, France.*

Van Heijst, *G.*, Schreiber, *A. et* Wielinga, *E. (1997). Using explicit ontologies in kbs development.* International Journal of Human-Computer Studies, *46(2):183–292.*

Visser, *P. R. S.*, Jones, *D. M.*, Bench-Capon, *T. J. M. et* Shave, *M. J. R. (1998). Assessing heterogeneity by classiying ontology mismatches.* In Proceedings of the 1^{st} International conference on Formal Ontology in Information Systems (FOIS), *pages 148–162, Trento, italy.*

Welty, *C. et* Guarino, *N. (2001). Supporting ontological analysis of taxonomic relationships.* Data and Knowledge Engineering, *1(39):51–74.*

Winkler, *W. (1999). The state of record linkage and current research problems. Rapport technique, Statistical Research Division, U.S. Bureau of the Census.*

Zghal, *S.*, Ben Yahia, *S.*, Mephu Nguifo, *E. et* Slimani, *Y. (2007a). A new alignment method for owl-lite ontologies using propagation of similarity over the graph.* In Proceeding of 18^{th} International Workshop on Database and Expert Systems Applications (DEXA 2007), *pages 524–528, Regensburg, Allemagne.*

Zghal, *S.*, Ben Yahia, *S.*, Mephu Nguifo, *E. et* Slimani, *Y. (2007b). SODA : an owl-dl based ontology matching system.* In Proceedings of the 2^{nd} International Workshop on Ontology Matching (OM-2007) Colocated with the 6th International Semantic Web

Conference (ISWC-2007) and the 2nd Asian Semantic Web Conference (ASWC-2007), *pages 261–267, Busan, Korea.*

Zghal, *S.*, Ben Yahia, *S.*, Mephu Nguifo, *E. et* Slimani, *Y. (2007c).* SODA *: Une approche structurelle pour l'alignement d'ontologies OWL-DL.* In Actes des 1$^{\text{ères}}$ Journées Francophones sur les Ontologies (JFO'2007), *pages 1–20, Sousse, Tunisie.*

Zghal, *S.*, Kachroudi, *M.*, Ben Yahia, *S. et* Mephu Nguifo, *E. (2009). OACAS : Ontologies alignment using composition and aggregation of similarities.* In Proceedings of the 1$^{\text{st}}$ International Conference on Knowledge Engineering and Ontology Development (KEOD 2009), *pages 233–238, Madeira, Portugal.*

Zghal, *S.*, Kamoun, *K.*, Ben Yahia, *S. et* Mephu Nguifo, *E. (2007d). Une nouvelle méthode d'alignement et de visualisation d'ontologies OWL-Lite.* In Actes des 7ème Journées Francophones Extraction et Gestion des Connaissances (EGC'2007), *volume 1, pages 197–198, Namur, Belgique.*

Zghal, *S.*, Kamoun, *K.*, Ben Yahia, *S.*, Mephu Nguifo, *E. et* Slimani, *Y. (2007e). EDOLA : Une nouvelle méthode d'alignement d'ontologies OWL-Lite.* In Actes de la of 4ème Conférence Francophone en Recherche d'Information et Applications (CORIA'2007), *pages 351–366, Saint-Étienne, France.*

Annexe A1 : Module de visualisation

La méthode d'alignement EDOLA est dotée d'un prototype de visualisation. Le prototype permet de visualiser les ontologies de trois manières différentes comme le montre la figure 4.6 :

- **Explorateur d'arborescence** : cette vue permet de montrer la structure arborescence des ontologies à éditer. Elle liste les classes, les propriétés de type objet et type de données en spécifiant le domaine et l'image ;
- **Visionneuse graphique d'ontologie** : cette composante est un éditeur graphique. Elle prend en considération une instance du graphe O-Graph pour effectuer la représentation graphique des ontologies à aligner ;
- **Editeur textuel de l'ontologie** : cet éditeur est muni d'une capacité de coloriage du code OWL. Il comporte aussi les fonctionnalités habituelles pour l'édition de fichiers texte (copier/coller à partir du presse papier, annuler/rétablir pour annuler et récupérer les dernières modifications, et des capacités de recherche et de remplacement du texte).

La visualisation graphique permet d'attribuer à chaque composant du graphe O-Graph une représentation graphique appropriée. En plus, différentes couleurs sont attribuées aux différentes entités ontologiques. Par exemple, la couleur blanche est dédiée pour la coloration des concepts, la couleur jaune est attribuée pour la représentation des propriétés de nature type de données, etc.

Le prototype de visualisation permet aussi de visualiser les deux ontologies à aligner d'une manière juxtaposée. Le prototype offre aussi l'opportunité de positionner les fenêtres de visualisation afin d'adapter la façon de visualiser les deux ontologies. La visualisation simultanée des deux ontologies permet d'aborder le processus d'alignement en les gardant représentées graphiquement. Le prototype assure la visualisation graphique du résultat de

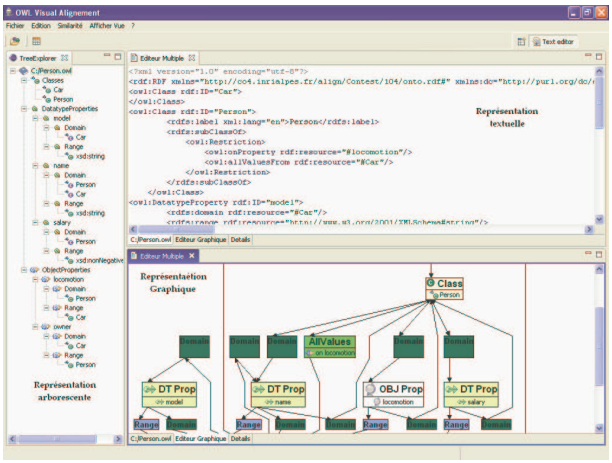


Figure 4.6 — Visualisation d'une ontologie

l'alignement. Ce type de visualisation facilite l'exploration du résultat par l'utilisateur de l'interface graphique. Un simple clique sur un nœud du graphe, de la première ontologie, permet d'afficher le nœud le plus similaire au niveau du second graphe, représentant la deuxième ontologie, ainsi que la valeur de la similarité correspondante. La figure 4.7 illustre la visualisation graphique du résultat de l'alignement.

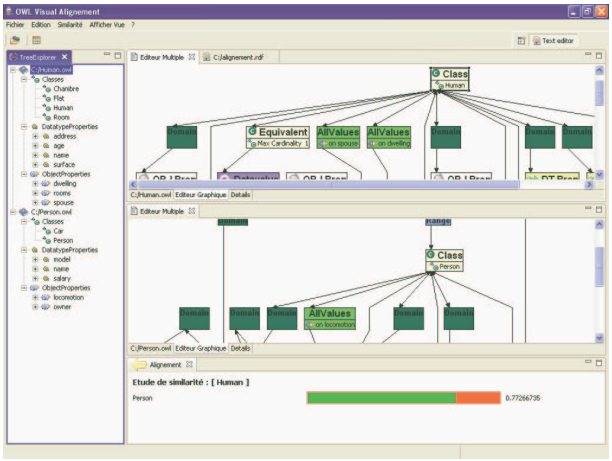


Figure 4.7 — Visualisation graphique de l'alignement

Annexe A2 : Exemple illustratif avec OACAS

Dans cet annexe, un scénario d'exécution, de la méthode OACAS, est proposé pour l'alignement de deux ontologies OWL-DL. Dans une première étape, les deux ontologies, représentées en OWL-DL, sont transformées sous la forme de deux graphes O-Graphs. La table 4.18 représente le code source des deux ontologies. Dans une seconde étape, le processus d'alignement est lancé après avoir effectué le paramétrage nécessaire. Le paramétrage consiste à attribuer l'ensemble de mesures de similarité nécessaire ainsi que les poids adéquats aux différents descripteurs (c-à-d. nom, commentaire et label) des entités ontologiques à aligner. La figure 4.8 décrit les deux graphes O-Graphs des deux ontologies à aligner.

Ontologie O_1	**Ontologie O_2**
<!DOCTYPE owl [	<!DOCTYPE owl [
<!ENTITY onto1 "http ://wwwsop.	<!ENTITY onto2 "http ://wwwsop.
inria.fr/acacia/ontologies/onto1.owl#">	inria.fr/acacia/ontologies/onto2.owl#">
<!ENTITY owl"http ://www.w3.org/2002/07/owl#">]	<!ENTITY owl"http ://www.w3.org/2002/07/owl#">]
<rdf :RDF xmlns :owl	<rdf :RDF xmlns :owl
xmlns :rdf ="http ://www.w3.org/1999/02/22-	xmlns :rdf ="http ://www.w3.org/1999/02/22-
rdf-syntax-ns#" xmlns :rdfs="http ://www.w3.org/	rdf-syntax-ns#" xmlns :rdfs="http ://www.w3.org/
2000/01/rdfschema#"xml :base ="onto1 ;" >	2000/01/rdfschema#"xml :base ="onto2 ;" >
<owl :Class rdf :ID="Person" />	<owl :Class rdf :ID="Human" />
<owl :Class rdf :ID="Adult">	<owl :Class rdf :ID="Woman">
<rdfs :subClassOf rdf :resource="#Person"/>	<rdfs :label>Woman</rdfs :label>
</owl :Class>	<rdfs :label>Adult Female</rdfs :label>
<owl :Class rdf :ID="Adult_Female">	<rdfs :subClassOf rdf :resource="#Human"/>
<rdfs :label>Adult Female</rdfs :label>	</owl :Class>

```
<rdfs :label>Woman</rdfs :label>
<rdfs :subClassOf rdf :resource="#Adult"/>
</owl :Class> <owl :Class rdf :ID="Man" />
<owl :Class rdf :ID="Adult_Male">
<rdfs :subClassOf rdf :resource="#Person"/>
</owl :Class> <owl :Class rdf :ID="Book">
<rdfs :subClassOf rdf :resource="#Publication"/>
<owl :disjointWith rdf :resource="#Magazine"/>
</owl :Class>
<owl :Class rdf :ID="Magazine">
<rdfs :subClassOf rdf :resource="#Publication"/>
</owl :Class>
<owl :Class rdf :ID="Publication" />
<owl :ObjectProperty rdf :ID="write">
<rdfs :domain rdf :resource="#Person"/>
<rdfs :range rdf :resource="#Publication"/>
</owl :ObjectProperty>
</rdf :RDF>
```

```
<owl :Class rdf :ID="Man">
<rdfs :subClassOf rdf :resource="#Human"/>
</owl :Class>
<owl :Class rdf :ID="Booklet" />
<owl :ObjectProperty rdf :ID="write">
<rdfs :subPropertyOf
rdf :resource="#create"/>
</owl :ObjectProperty>
<owl :ObjectProperty rdf :ID="create">
<rdfs :domain rdf :resource="#Human"/>
<rdfs :range rdf :resource="#Booklet"/>
<rdfs :domain rdf :resource="#Human"/>
<rdfs :range rdf :resource="#Booklet"/>
</owl :ObjectProperty>
</rdf :RDF>
```

Tableau 4.18: Code source des deux ontologies considérées dans l'exemple illustratif

Les différentes étapes de calcul de la similarité sont réalisées en fonction des liens d'équivalence qui peuvent exister entre entités ontologiques des deux ontologies. Ces étapes de calcul de similarité permettent de déterminer les trois vecteurs de similarité : le vecteur de la similarité linguistique composée (V_{SLC}), le vecteur de la similarité de voisinage (V_{SV}) et le vecteur de la similarité agrégée (V_{SA}). La première partie de la table 4.19 représente les valeurs de la mesure de similarité linguistique composée (V_{SLC}) pour les différentes entités des deux ontologies O_1 et O_2. Le calcul de ces valeurs est réalisé par l'intermédiaire de la meilleure combinaison des mesures de similarité possibles (c-à-d. Levenshtein pours les noms, Q-Gram pour les commentaires et Jaro-Winkler pour les labels). Par la suite, les valeurs du vecteur de la similarité de voisinage (V_{SV}) sont déterminées pour les deux ontologies O_1 et O_2. Ces valeurs sont résumées dans la seconde partie de la table 4.19. Une fois que les valeurs de la similarité linguistique composée et la similarité du voisinage sont calculées, l'étape d'agrégation est réalisée. Cette étape détermine les valeurs de

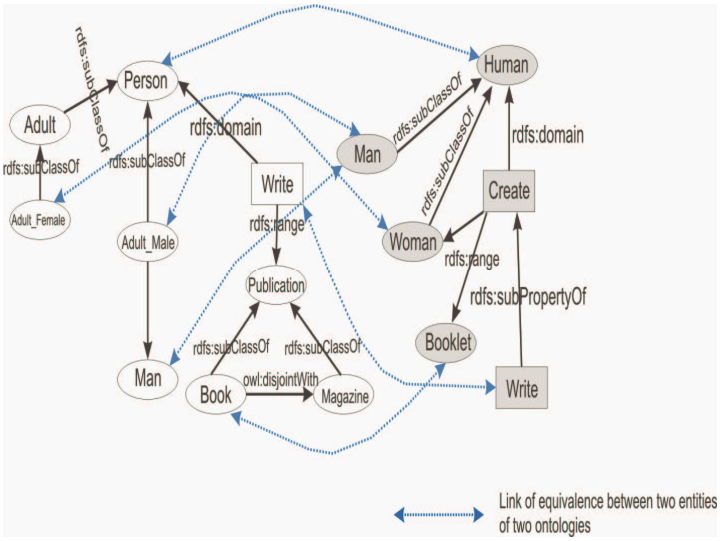


Figure 4.8 — Graphes associés aux deux ontologies de l'exemple illustratif

la similarité agrégée (V_{SA}) en combinant les valeurs des deux mesures de similarité. Les valeurs de la similarité agrégée sont illustrées dans la troisième partie de la table 4.19.

O_1	O_2	V_{SLC}	V_{SV}	V_{SA}
Man	Human	1.00	1.00	1.00
Write	Create	0.60	0.55	0.58
Adult_Male	Human	0.00	0.98	0.39
Magazine	Human	0.71	0.79	0.74
Publication	Booklet	0.56	0.59	0.57
Adult	Man	0.00	0.00	0.00
Person	Human	0.00	0.00	0.00
Adult_Female	Woman	0.00	0.98	0.39
Book	Booklet	1.00	1.00	1.00

Tableau 4.19 — Valeurs des vecteurs de similarité (V_{SLC}, V_{SV} et V_{SA}) pour les ontologies O_1 et O_2

Le résultat de l'alignement est un fichier RDF. Ce fichier contient une description détaillée des degrés de correspondance entre les entités des deux ontologies. Un exemple de ce fichier est proposé dans la table 4.20. Le fichier est composé d'un ensemble de cellules. Chaque cellule est composée de quatre champs. Les deux premiers champs indiquent les chemins des deux entités ontologiques mises en correspondances. Le troisième champ représente le degré de similarité. Le quatrième champ évoque le type de relation (l'égalité).

```
<?xml version="1.0" encoding="iso-8859-1"?>"
<rdf :RDF 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>11</type>
<onto1>http ://www.sop.inria.fr/acacia/ontologies/onto1.owl</onto1>
<onto2>http ://www.sop.inria.fr/acacia/ontologies/onto2.owl</onto2>
<uri1>http ://www.sop.inria.fr/acacia/ontologies/onto1.owl</uri1>
<uri2>http ://www.sop.inria.fr/acacia/ontologies/onto2.owl</uri2>
<map>
<cell>
<entity1 rdf :resource='http ://www.sop.inria.fr/acacia/ontologies/onto1.owl#Man'/>
<entity2 rdf :resource='http ://www.sop.inria.fr/acacia/ontologies/onto2.owl#Human'/>
<measure rdf :datatype='http ://www.w3.org/2001/XMLSchema#float'>1.0</measure>
<relation>=</relation>
</cell>
</map>
<map>
<cell>
<entity1 rdf :resource='http ://www.sop.inria.fr/acacia/ontologies/onto1.owl#Write'/>
<entity2 rdf :resource='http ://www.sop.inria.fr/acacia/ontologies/onto2.owl#Create'/>
<measure rdf :datatype='http ://www.w3.org/2001/XMLSchema#float'>0.58</measure>
<relation>=</relation>
</cell>
</map>
<map>
<cell>
<entity1 rdf :resource='http ://www.sop.inria.fr/acacia/ontologies/onto1.owl#Adult_Male'/>
<entity2 rdf :resource='http ://www.sop.inria.fr/acacia/ontologies/onto2.owl#Human'/>
<measure rdf :datatype='http ://www.w3.org/2001/XMLSchema#float'>0.39</measure>
<relation>=</relation>
</cell>
</map>
...
...
</Alignment>
</rdf :RDF>
```

Tableau 4.20 — Aperçu du fichier RDF