

Ministère des Enseignements Secondaire et Supérieur (M.E.S.S.)

-----  
Secrétariat Général (S.G.)  
-----

Université Polytechnique de Bobo-Dioulasso (U.P.B.)  
-----

Ecole Supérieure d'Informatique (E.S.I.)



Cycle des Ingénieurs de Conception en Informatique (C.I.C.I.)  
Troisième année

## MEMOIRE DE FIN DE CYCLE

### **THEME :**

***Prise en compte de l'interactivité dans le processus d'alignement d'ontologies***

**Présenté et soutenu par**

**SAVADOGO Mahamadi**

**En vue de l'obtention du diplôme d'ingénieur de conception en informatique**

**Directeur de mémoire**

**Dr. Borlli Michel Jonas SOME**

Maitre-assistant en informatique/CAMES

*Université Polytechnique de Bobo-Dioulasso/Burkina Faso*

A handwritten signature in black ink, appearing to be 'Borlli Michel Jonas SOME', is written over a light blue rectangular background.

**Co-encadreur**

**Dr. Gayo DIALLO**

Maitre de conférences en informatique

*Université de Bordeaux /France*

Année académique 2012-2013

## Remerciements

Je tiens à remercier toutes les personnes qui de près ou de loin ont contribué d'une manière ou d'une autre au succès de ce travail.

Mes remerciements s'adressent particulièrement à :

- » Monsieur Gayo DIALLO, maître de Conférences en Informatique à l'université Bordeaux 2, mon encadrant depuis Bordeaux, pour sa disponibilité, son aide et sa compréhension durant tout le stage ;
- » Monsieur Michel Borlli J. SOME, enseignant chercheur à l'école supérieure d'informatique (Université polytechnique de Bobo-Dioulasso, mon directeur de mémoire, pour sa disponibilité, ses motivations, ses conseils et encouragements ;
- » La direction de l'école supérieure d'informatique et tout le corps enseignant pour la qualité de la formation reçue.
- » Le personnel de l'école supérieure d'informatique pour l'accueil, la bonne cohésion reçue ;
- » Ma famille, mes proches et amis pour leur compréhension et leurs encouragements et à tous ceux dont je n'ai pas cité de nom et qui ont contribué de leur manière à la réussite de ce travail.

# Table des matières

Remerciements . . . . .	i
Sigles et abréviations . . . . .	v
Avant-propos . . . . .	vi
Résumé . . . . .	vii
Abstract . . . . .	viii
Introduction générale . . . . .	1
<b>1 GENERALITES</b>	<b>2</b>
<b>2 ETAT DE L'ART</b>	<b>6</b>
1 Web sémantique . . . . .	6
2 Ontologie . . . . .	7
3 Alignement d'ontologies . . . . .	7
3.1 Définition . . . . .	7
3.2 Objectifs . . . . .	8
3.3 Processus d'alignement d'ontologies . . . . .	8
3.4 Techniques d'alignement . . . . .	9
3.5 Application . . . . .	12
3.6 Alignement interactif d'ontologies . . . . .	13
3.7 Qualité de l'alignement . . . . .	14
3.8 OAEI challenge . . . . .	15
<b>3 ETUDE DE L'EXISTANT ET CAPTURE DES BESOINS</b>	<b>17</b>
1 Etude de l'existant . . . . .	17
1.1 Etude du système ServOMap . . . . .	17
1.2 Processus d'alignement de ServOMap . . . . .	18
1.3 Limites du système et besoin en interactivité . . . . .	20
2 Spécification des besoins . . . . .	20
2.1 Les besoins fonctionnels . . . . .	20
2.2 Les besoins non fonctionnels . . . . .	21
3 Conclusion . . . . .	24
<b>4 CONCEPTION</b>	<b>25</b>
1 Analyse conceptuelle du système . . . . .	25

1.1	Processus d'alignement interactif du système . . . . .	26
1.2	Diagramme de composant du système . . . . .	28
1.3	Diagramme de paquetages . . . . .	30
1.4	Diagramme de classes du système . . . . .	31
2	Mise en œuvre du système . . . . .	32
2.1	Environnement de travail . . . . .	32
2.2	Implémentation . . . . .	34
2.3	Evaluation de l'approche . . . . .	42
	Conclusion générale . . . . .	44
	<b>BIBLIOGRAPHIE</b> . . . . .	45
	<b>ANNEXES</b> . . . . .	49

# Table des figures

1.1	Cycle de vie en V . . . . .	5
2.1	Processus d'alignement de deux ontologies . . . . .	9
2.2	schéma de processus d'alignement interactif . . . . .	14
3.1	Processus d'alignement avec ServOMap . . . . .	18
3.2	diagramme de cas d'utilisation de l'alignement interactif d'ontologies . . . . .	22
3.3	diagramme de séquence de l'alignement interactif d'ontologies	23
4.1	Processus d'alignement interactif du système . . . . .	27
4.2	Diagramme de composant du système . . . . .	29
4.3	Diagramme de paquetages de ServOMap . . . . .	30
4.4	Diagramme de classes de ServOMap . . . . .	31
4.5	Architecture de l'application . . . . .	33
4.6	page d'accueil de l'application . . . . .	35
4.7	page de chargement d'une ontologie . . . . .	35
4.8	paramétrage et indexation d'une ontologie . . . . .	36
4.9	page de chargement d'une référence . . . . .	37
4.10	page de proposition de correspondances . . . . .	37
4.11	paramétrage de l'alignement . . . . .	38
4.12	paramétrage de l'alignement : choix des métriques et valeurs de confiance . . . . .	39
4.13	liste de correspondances soumise à la validation par l'utilisateur	39
4.14	détails sur un couple candidat sélectionné . . . . .	40
4.15	liste des couples sélectionnées comme valide par l'utilisateur .	40
4.16	Résultat de l'alignement . . . . .	41
4.17	résultat de l'évaluation . . . . .	41
4.18	Évaluation du processus d'alignement d'ontologies . . . . .	43
4.19	une partie de l'ontologie des êtres vivants . . . . .	50
4.20	pyramide des langages du web sémantique . . . . .	52
4.21	schéma du triplet RDF . . . . .	53
4.22	la structure des ontologies OWL 2 . . . . .	56

## Sigles et abréviations

Sigle	Signification
AJAX	Asynchronous Javascript and XML
API	Application Programming Interface
ERIAS	Equipe de Recherche en Informatique Appliquée à la Santé
ESI	Ecole Supérieure d'Informatique
IHM	Interface Homme-Machine
IM	Indexing Module
ISPED	Institut de Santé Publique, d'Epidémiologie et de Développement
J2EE	Java Enterprise Edition
JSF	Java Server Faces
MM	Mapping Module
MVC	Model View Controller
OAEI	Ontology alignment evaluation initiative
OBO	Open Biomedical Ontologies
OMM	Ontology Manager Module
OWL	Ontology web language
RAM	Read Access Memory
RDF	Resource Description Framework
RDF(S)	Resource Description Framework Schema
RI	Recherche d'Information
ServO	Serveur d'ontologies
ServOMap	Server of Ontologies Mapping
SKOS	Simple Knowledge Organisation System
UML	Unified Modeling Language
URI	Uniform Resource Identifier
XML	eXtensible Markup Language
W3C	World Wide Web Consortium

## Avant-propos

L'École Supérieure d'Informatique est une école de l'Université Polytechnique de Bobo-Dioulasso. Elle forme des ingénieurs de travaux et de conception en informatique.

Le cycle des ingénieurs de travaux informatiques a une durée de trois (03) ans et présente deux options : analyse-programmation, réseau et maintenance informatique.

Le diplôme d'ingénieur de conception en informatique se prépare en cinq (05) ans soit deux (02) années après le diplôme d'ingénieur de travaux informatiques.

A l'issue de chaque cycle de formation un stage d'une durée de trois (03) mois pour les ingénieurs de travaux et quatre (04) mois pour les ingénieurs de conception en informatique est effectué dans une entreprise. Un stage au cours duquel l'étudiant met en œuvre ses compétences pratiques et théoriques acquises à l'école au profit de l'entreprise et réalise un travail qui sera évalué au cours d'une soutenance publique lui donnant droit au diplôme valu. Ainsi, le travail qui sera présenté dans ce document se situe dans ce contexte.

**Résumé.** — La construction d'ontologies ne repose pas sur une technique universelle puisqu'il existe une certaine hétérogénéité liée à la syntaxe, à la terminologie utilisée et à la conception. Pour des besoins d'intégration et d'interopérabilité entre les systèmes manipulant ces ontologies, l'implémentation de mécanismes d'identifications de correspondances appelé alignement s'est avéré nécessaire. Ces mécanismes pour la plus part ne prennent pas en compte l'intervention d'un utilisateur expert du domaine dans leur processus. Nous proposons dans ce document une approche qui permet une interaction avec l'utilisateur afin de produire un alignement aussi précis que possible et de très bonne qualité. L'intervention de l'utilisateur qui est un expert du domaine est caractérisée en premier lieu par le paramétrage du système d'alignement. Le paramétrage du système consiste au choix des méthodes de calcul des similarités et la définition des niveaux de fiabilité des couples candidats. Ensuite la proposition de correspondances initiales non susceptible d'être trouvé par le système et la validation de couples candidats sélectionnés par le système. En outre, il est possible d'utiliser comme ressources externes, des alignements de référence disponibles, pour une validation automatique des couples candidats. Enfin, le processus permet de fournir des résultats intermédiaires en offrant la possibilité à l'utilisateur de procéder à une validation de manière récursive. Cette approche a été implémentée dans le système d'alignement ServOMap, qui permet de traiter de larges ontologies.

**Mots-clés :** Ontologies, alignement d'ontologies, alignement interactif d'ontologies.



**Abstract.** — Ontologies building is not based on a universal technique since there is some heterogeneity related to the syntax, terminology and the design. For purposes of integration and interoperability between systems handling these ontologies, implementation of mechanisms for identification of correspondences called alignment has proved necessary. These mechanisms for the most part don't take involvement of user domain expert into their matching process. In this document, we propose an approach that allows interaction with the user to produce as accurate alignment as possible and very good quality. The involvement of the user who is a domain expert is characterized by setting of alignment system. The system setting is to choose methods of similarities computation and definition of reliability levels of candidates. Then proposition of initial correspondences which cannot be found by the system and validating of candidates selected by the system. Furthermore, it is possible to use as external resources, reference alignments available for automatic validation of couples candidates. Finally, this process permit to provide intermediate results in allowing user to perform validation recursively. This approach has been implemented in the matching system ServOMap which can handle large ontologies.

**Key words:** Ontologies, ontologies alignment, ontologies interactive alignment

## Introduction générale

Ces dernières années, les ontologies ont suscité de nombreux travaux et recherches dans le domaine du web sémantique. Elles sont utilisées pour fournir le vocabulaire sémantique permettant de rendre la connaissance du domaine disponible pour l'échange et l'interprétation au travers des systèmes d'information. Toutefois, en raison de la nature décentralisée du web sémantique, les ontologies sont très hétérogènes. Cette hétérogénéité provoque le problème de la variation de sens ou ambiguïté dans l'interprétation des entités et, par conséquent, elle empêche le partage des connaissances du domaine. L'alignement d'ontologies, qui a pour but la découverte des correspondances sémantiques entre des ontologies, devient une tâche cruciale pour résoudre ce problème d'hétérogénéité dans les applications du web sémantique. Les principaux défis dans le domaine de l'alignement d'ontologies ont été décrits dans des études récentes. Parmi eux, le choix des mesures de similarité, l'utilisation des informations et des ressources appropriées, la vérification de la cohérence sémantique, l'implication de l'utilisateur dans le processus d'alignement car la plupart des systèmes d'alignement jusque-là sont comme une boîte noire et automatique (génération automatique des correspondances), etc. Par ailleurs, la difficulté du problème augmente avec la taille des ontologies. C'est notamment le cas dans le domaine biomédical où certaines ontologies peuvent avoir des centaines de milliers d'entités [Savadogo et al. 2014].

Pour faire face à ces défis, plusieurs systèmes ont été implémentés, en utilisant différentes approches, mais rares sont ceux qui prennent en compte l'intervention d'un expert. Lors de ce projet, nous proposons et implémentons une approche qui va prendre en compte l'intervention d'un expert du domaine dans le processus d'alignement. Cette intervention se caractérise par le paramétrage du système, le choix des mesures de similarité et la validation des alignements proposés par le système (acceptation ou réfutation de correspondances).

# Chapitre 1

## GENERALITES

### Introduction

De façon générale, un stage a une raison d'être, un cadre de déroulement, un objectif et plan ou méthode de réalisation. Dans ce chapitre, il sera question de présenter de façon générale ces éléments essentiels qui revêt ce stage.

### Contexte du stage

Cet stage est le fruit d'une collaboration entre l'université de Bordeaux Segalen ISPED (L'Institut de Santé Publique, d'Epidémiologie et de Développement) et l'Ecole Supérieure d'Informatique (ESI) de l'université polytechnique de Bobo-Dioulasso à travers les enseignants Dr. Gayo DIALLO, Maître de conférences en informatique à l'université de Bordeaux et Dr. Michel Borli J. SOME , enseignant chercheur en informatique de l'université polytechnique de Bobo-Dioulasso. Ce stage s'est déroulé à l'université polytechnique de Bobo-Dioulasso, au sein de la direction de l'école supérieure d'informatique sous la supervision et l'encadrement technique du Dr. Michel Borli J. SOME et du Dr. Gayo DIALLO, mon Co encadrant en France. Il porte sur la prise en compte de l'interactivité dans le processus d'alignement d'ontologies : cas du système ServOMap.

### L'Institut de Santé Publique, d'Epidémiologie et de Développement

L'Institut de Santé Publique, d'Epidémiologie et de Développement (ISPED), au sein de l'Université Bordeaux Segalen, est la première expérience française de création d'une École de Santé Publique distincte, bien que très proche, des facultés de médecine. Son objectif est de contribuer à relever les grands défis posés par la santé publique contemporaine : l'accroissement de l'espérance de vie, la refonte des systèmes de santé, la résurgence des maladies infectieuses dans le monde, l'impact de l'industrialisation et de la mondialisation sur l'environnement et la santé des populations. La force de l'ISPED est de disposer d'une équipe de chercheurs hautement qualifiés, re-

groupés autour du Centre de recherche Inserm U897 "Epidémiologie et bio statistique" et d'équipes de recherche hospitalières et universitaires. Les principaux travaux des équipes scientifiques portent sur le vieillissement cérébral, le VIH, la recherche clinique, la nutrition, les traumatismes, la santé au travail, la santé et l'environnement, le cancer, la bio statistique, l'informatique médicale, l'évaluation des politiques publiques, la gestion hospitalière, etc<sup>1</sup>.

## Problématique et objectif

Un grand nombre de systèmes d'alignement d'ontologies ont été développés au cours de ces dernières années. La plupart de ces systèmes sont automatiques : la génération de correspondance est automatique. Cependant, la génération automatique des correspondances entre deux ontologies est d'une extrême difficulté qui est dû aux divergences (conceptuelle, habitudes, etc.) entre communautés différentes de développement des ontologies et altère un peu la qualité de l'alignement. Etant donné que les ontologies à aligner décrivent les connaissances d'un domaine spécifique, il est donc utile et même souhaitable de faire recourt aux connaissances et riche expériences d'un expert du domaine pour décider si certaines correspondances proposées par le système sont correctes ou pas. Cette intervention permet d'améliorer la précision et la qualité de l'alignement. En outre, après plusieurs années d'expériences, l'OAEI<sup>2</sup> qui organise des campagnes d'évaluation sur les systèmes d'alignement automatique a émis un constat sur le fait qu'il existe une limite sur la qualité de l'alignement(en termes de précision, rappel, F-mesure) difficile à dépasser par les systèmes d'alignement automatique et que l'intervention d'un expert pourrait considérablement améliorer les résultats des alignements. C'est ainsi que la prise en compte de l'utilisateur dans le processus d'alignement est devenu une nécessité et a d'ailleurs été l'un des principaux enjeux de la campagne de l'OAEI 2012<sup>3</sup> et c'est à partir de là que les organisateurs de cette campagne ont décidé d'inclure dans les campagnes d'évaluation à venir une évaluation réservée aux systèmes d'alignement interactif et la première évaluation des systèmes d'alignement interactif a été réalisée au cours de la campagne de l'OAEI 2013<sup>4</sup> à Sydney. Par ailleurs l'alignement d'ontologies se révèle assez difficile dès lors que la taille de ces dernières devient importante. La plupart des méthodes d'alignement existantes rencontrent des difficultés à produire des alignements de bonne qualité quand la taille des ontologies est très grande. Un certain nombre d'algorithmes pour l'alignement d'ontologies à large échelle ont été implémenté tels que les algo-

---

1. <http://www.isped.u-bordeaux2.fr/>

2. <http://oaei.ontologymatching.org>

3. <http://oaei.ontologymatching.org/2012>; <http://om2012.ontologymatching.org>

4. <http://oaei.ontologymatching.org/2013>

rithmes d'alignement incrémentaux [Ngo et al. 2012] qui prennent en compte l'évolution des ontologies, mais reste encore insuffisant.

L'objectif de ce projet est de proposer et implémenter une approche d'alignement interactif des ontologies. Cette approche sera implémentée sur le système ServOMap afin de lui permettre de prendre en compte l'intervention de l'utilisateur dans son processus d'alignement d'ontologies. En pratique c'est de réaliser une couche pour l'alignement interactif d'ontologies qui sera intégrée au système actuel.

## Méthodologie

Pour la conduite d'un projet de développement il nous revient de choisir un langage de modélisation et d'adopter une méthode pour mener l'étude et le développement. Pour ce qui est du langage de modélisation notre choix s'est porté sur UML (Unified Modeling Language). A travers ses diagrammes, il nous permettra de modéliser, documenter et décrire le système. Notre système est divisé en module et constitue la suite d'un projet développé en utilisant la méthode de développement du cycle de vie en V<sup>5</sup> (Figure 1.1). Des tests sont effectués pour confirmer la conformité des spécifications conceptuelles et techniques des modules développés puis la conformité aux spécifications conceptuelles et techniques du système résultant de l'intégration de tous les modules. J'ai gardé la méthodologie du cycle de vie en V et pour mieux gérer et organiser le projet j'ai utilisé l'outil de gestion de projet Maven<sup>6</sup> qui permet également d'effectuer des intégrations et des tests unitaires aisés. Cette approche me permet d'augmenter la qualité et la réutilisation de mon système. Maven est un outil de gestion de projet (voire annexe).

## Organisation du document

La suite du mémoire est organisée en trois chapitres. Le chapitre 2 intitulé « Etat de l'art ». Dans ce chapitre je présenterai les concepts auxquels mon travail fait référence à savoir le web sémantique, les ontologies et l'alignement d'ontologies.

Le chapitre 3 nommé « Etude de l'existant et capture des besoins » me permet d'étudier le système existant et d'analyser les besoins du système. Dans ce chapitre je présenterai le système ServOMap et les techniques d'alignement utilisées dans son processus d'alignement en un premier point et en un deuxième point je décrirai les besoins fonctionnels et non fonctionnels de la nouvelle version du système.

---

5. [http://fr.wikipedia.org/wiki/Cycle\\_en\\_V](http://fr.wikipedia.org/wiki/Cycle_en_V)

6. [http://fr.wikipedia.org/wiki/Apache\\_Maven](http://fr.wikipedia.org/wiki/Apache_Maven)

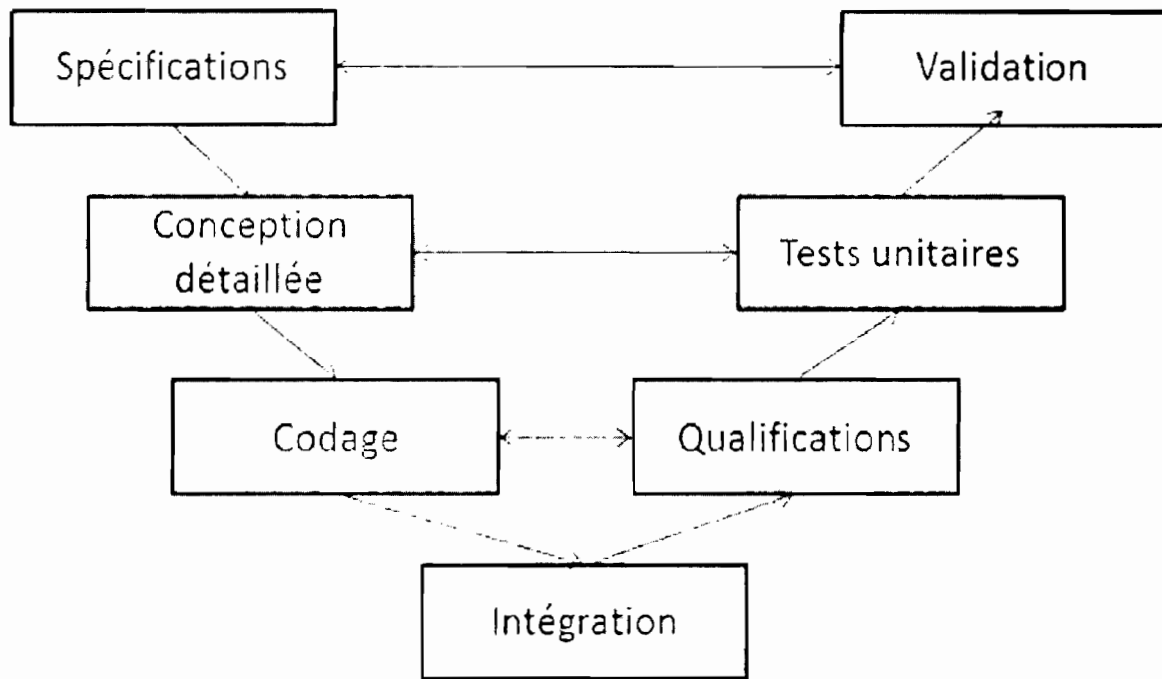


FIGURE 1.1 · Cycle de vie en V

Le chapitre 4 que j'ai appelé « Conception » présente en détail l'approche que j'ai proposé pour la prise en compte de l'interactivité dans le processus d'alignement d'ontologies du système ServOMap et les différents outils et technologies utilisés pour la mise en place de ma solution.

## Conclusion

Ce chapitre a présenté de façon générale, le contexte du stage, le thème à travers la problématique et les objectifs, les langages de modélisation et la méthode de développement utilisés pour la réalisation du projet. Cette présentation se termine par la définition de l'organisation de la suite du mémoire.

## Chapitre 2

# ETAT DE L'ART

### Introduction

Dans ce chapitre je vais présenter un état des connaissances existantes sur l'alignement d'ontologies et certaines notions de bases liées au thème. Cette présentation débutera par une définition du web sémantique (WS), la notion d'ontologie dans son ensemble. Ensuite je définirai le terme alignement d'ontologie et présenterai les techniques d'alignement existantes. Cette présentation se terminera par la définition de la notion d'alignement interactif.

### 1 Web sémantique

Selon Tim Berners-Lee. *"Le Web sémantique est une extension du web classique, dans laquelle l'information reçoit une signification bien définie, améliorant les possibilités de travail collaboratif entre les ordinateurs et les personnes."* [Tim Berners-Lee, 2001]. C'est un moyen d'extraction et d'exploitation de l'intelligence collective du Web, en rendant le contenu plus facile à échanger et à partager entre les différents utilisateurs. Il a comme objectif majeur d'apporter de la sémantique à l'information manipulée afin de faciliter la communication entre agents (hommes et machines). Tim Berners-Lee, auteur du premier article sur ce thème, et ses collègues prenant acte du fait que le Web actuel n'est "compréhensible" que par les humains, indique que le but du Web sémantique est de rendre explicite le contenu sémantique des ressources dans le Web (documents, pages web, services ...) et fournir une représentation de ce contenu sous une forme plus facilement traitable par la machine. Les ordinateurs et les agents logiciels pourraient donc "comprendre" les informations contenues dans ces ressources et aider les utilisateurs à exécuter et compléter leurs tâches, leurs requêtes de façon plus automatique et plus efficace. Ces ressources sont décrites par des informations structurées additionnelles "les métadonnées" qui sont fournies par le composant pivot du WS, les ontologies. Ces dernières représentent une technologie dont le but est d'améliorer l'organisation, la gestion et la compréhension de l'informa-

tion électronique. Elles servent de vocabulaire standardisé pour le partage de connaissances. Les langages et technologies du Web sémantique sont parfois présentés comme des outils de représentation des connaissances adaptés à l'environnement Web, permettant de transformer automatiquement les données en information, et les informations en connaissances.

## 2 Ontologie

La notion d'ontologie provient d'une branche philosophique traitant de la science de l'être. Cette branche, initiée par Aristote tente de définir les propriétés générales de l'être à travers ce qui le caractérise de façon essentielle. Le terme "ontologie" a été introduit en informatique dans les années 70 par McCarthy dans le domaine de l'Intelligence Artificielle. Par analogie, le terme "ontologie" est repris pour désigner l'ensemble structuré des termes et concepts représentant le sens d'un domaine d'informations. La définition la plus répandue de cette notion d'ontologie est celle proposée initialement par [Gruber, 1993] et complétée par [Borst, 1997] : *"une ontologie est une spécification formelle d'une conceptualisation partagée"*. Une conceptualisation est une abstraction d'un domaine d'application, elle identifie les concepts pertinents de ce domaine. Une conceptualisation partagée signifie que les concepts dans l'ontologie sont acceptés par une communauté d'utilisateurs. L'ontologie est la spécification explicite parce que tous les concepts et les contraintes utilisées de ces concepts sont explicitement définis. La spécification formelle implique que l'ontologie doit être représentée par un langage formel avec lequel la machine peut opérer. Pour cette définition, une ontologie fournit un vocabulaire partagé, incluant les concepts importants, les propriétés, les définitions, et les contraintes qui sont employés dans un domaine communicatif entre les personnes et les systèmes d'application hétérogènes et distribués.

## 3 Alignement d'ontologies

### 3.1 Définition

L'alignement d'ontologies est le processus de mise en correspondance sémantique des entités qui les composent [Euzenat et al, 2004]. Il s'agit généralement de construire des appariements entre les éléments décrits dans différentes ontologies (classes, propriétés, individus ou instances, etc.). L'appariement d'ontologies [Euzenat et Shvaiko, 2007] consiste à trouver des correspondances entre entités d'ontologies différentes. Ces correspondances reposent notamment sur l'existence de propriétés similaires : des relations d'équivalence, de conséquence, de subsumption entre entités, etc. Pour effectuer un alignement, la connexité des deux domaines de connaissance modéli-



sés par les ontologies est requise, sans quoi aucun lien ne peut être établi entre concepts [Kefi et al. 2006]. De plus, les formalismes de représentation d'ontologie utilisés doivent être au moins compatibles, ainsi que les paradigmes conceptuels [Furst. 2004].

### 3.2 Objectifs

#### 1. Réduire l'hétérogénéité

- ★ L'hétérogénéité syntaxique : concerne essentiellement le fait qu'une ontologie peut être modélisée dans différents langages comme OWL, KIF, DAML+OIL, etc. :
- ★ L'hétérogénéité terminologique : concerne les différences de nommage des entités. Ces différences sont principalement liées à la complexité des langues naturelles. Par exemple, il peut y avoir des cas de polysémie (deux entités identiques nommées de manières différentes " *automobile* " vs " *véhicule* " ), de synonymie (deux entités différentes nommées de la même manière), de multilinguisme (deux entités identiques nommées de la même manière, mais dans des langues différentes), etc.
- ★ L'hétérogénéité conceptuelle : regroupe trois aspects : le point de vue duquel s'est placé le concepteur de l'ontologie (la perspective), le domaine du monde qu'il a cherché à représenter (la couverture), et le niveau de détail qu'il a voulu atteindre (la granularité). La perspective est issue de la difficulté à créer une ontologie avec un regard objectif.

#### 2. Permettre une utilisation conjointe de plusieurs ontologies.

Le résultat de cette tâche assure et facilite l'échange, le partage, la fusion des données et des informations entre systèmes ou communautés dans le web sémantique.

### 3.3 Processus d'alignement d'ontologies

[Euzenat et Shvaiko. 2007] définissent une correspondance entre deux ontologies  $\mathbf{o}$  et  $\mathbf{o}'$  comme étant un 5-uplet  $(id, e, e', r, n)$  dans lequel  $\mathbf{id}$  représente l'identifiant de la correspondance,  $\mathbf{e}$  et  $\mathbf{e}'$  sont des entités respectives de  $\mathbf{o}$  et  $\mathbf{o}'$ ,  $\mathbf{r}$  est une relation entre  $\mathbf{e}$  et  $\mathbf{e}'$ , et  $\mathbf{n}$  est une valeur de confiance attribuée à la correspondance. La relation  $\mathbf{r}$  est souvent un lien d'équivalence ( $\equiv$ ) ou de subsumption (plus-spécifique-que,  $\sqsubseteq$ ), mais peut également être une disjonction ( $\neq$ ), une généralisation (plus-général-que,  $\sqsupseteq$ ), une intersection ( $\cap$ ), etc. En règle générale, un processus d'alignement fait intervenir plusieurs méthodes qui ont chacune pour objectif de trouver des correspondances entre

deux ontologies. Chaque processus prend en entrée deux ontologies  $\mathbf{o}$  et  $\mathbf{o}'$ , un alignement  $\mathbf{a}$  (éventuellement vide) et un ensemble de paramètres et de ressources et produit un alignement  $\mathbf{a}'$  en sortie (voir figure 2.1).

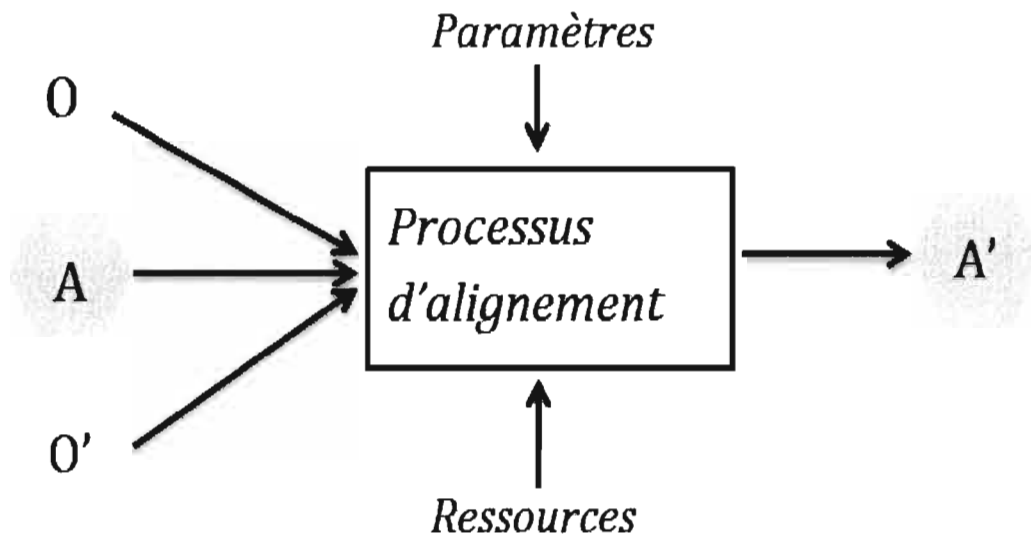


FIGURE 2.1 – Processus d'alignement de deux ontologies

### 3.4 Techniques d'alignement

Dans la littérature, plusieurs méthodes ou techniques d'alignement d'ontologies ont été proposées. Elles tirent parti des différents aspects des ontologies. Et elles s'intéressent à l'alignement des ontologies décrites dans différents langages d'ontologie [Djakhdjakha et Hemam, 2010]. Ces techniques s'appuient en général sur des mesures de calcul de similarité entre concepts d'ontologies différentes. Ces mesures sont pour la plupart fondées sur les caractéristiques lexicales des labels des concepts et/ou les caractéristiques structurelles des ontologies. Ce qui implique la comparaison de chaque description de concept d'une ontologie avec les descriptions de tous les concepts de l'autre ontologie [Hamdi, 2008].

#### 3.4.1 Les techniques terminologiques

Les techniques terminologiques comparent les termes utilisés comme labels pour désigner les différents éléments composant les ontologies. Elles s'appliquent sur les noms, les commentaires et les propriétés des concepts afin de trouver ceux qui sont similaires. Ces techniques se répartissent en deux familles : les techniques syntaxiques et les techniques linguistiques.

**Techniques syntaxiques** : ces techniques sont basées sur la comparaison des chaînes de caractères : elles supposent que deux concepts sont sémantiquement proches si les termes qui les désignent sont similaires syntaxiquement. Deux chaînes qui partagent des caractères ou des mots en commun seront considérées comme similaires [Zghal, 2010]. Les résultats obtenus par

ces méthodes sont utiles si les concepteurs utilisent des chaînes de caractères similaires pour définir la même entité. en revanche, s'il y a des synonymes avec des structures différentes ces méthodes donnent une mauvaise estimation de la similarité.

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, la mesure Q-Gram, distance de Monge-Elkan (confère annexe pour plus de détail sur ces mesures).etc.

**Techniques linguistiques :** les entités ici sont considérées comme un texte (des mots) dans un langage donné. elles font appel généralement aux techniques du traitement automatique du langage (TAL) pour extraire les termes les plus représentatifs à partir d'un texte, elles utilisent généralement des ressources externes telles que les dictionnaires, les thésaurus ... L'inconvénient de ces méthodes est que les ressources externes ne sont pas toujours disponibles pour certaines langues.

#### 3.4.2 Techniques structurelles

Ces techniques déduisent la similarité de deux entités en exploitant des informations structurelles lorsque les entités en question sont reliées aux autres par des liens sémantiques ou syntaxiques, formant ainsi une hiérarchie ou un graphe des entités. On distingue deux types de méthodes structurelles [Euzenat et Shvaiko, 2005] : celles basées sur la structure interne d'une entité et celles basées sur la structure externe. La structure interne d'une entité représente les attributs possédés par l'entité (attributs de type simple, cardinalités, restrictions). La structure externe représente les relations qu'entretient l'entité avec d'autres.

**Méthodes structurelles internes :** Ces méthodes calculent la similarité entre deux entités en exploitant des informations des structures internes de ces entités. Dans la plupart des cas, ce sont des informations concernant des attributs de l'entité, telles que des informations du domaine des attributs, celles de la cardinalité des attributs, celles des caractéristiques des attributs (la transitivité, la symétrie), ou celles des autres types de restriction sur des attributs. Ces méthodes sont généralement combinées avec celles des techniques terminologiques.

**NB :** il n'offre pas vraiment beaucoup d'information sur les entités à comparer. plusieurs entités très hétérogènes peuvent être décrites par la même structure interne et des entités très proches peuvent avoir des structures différentes.

**Méthodes structurelles externes** : qui utilisent la comparaison externe en mettant par exemple en jeu la disposition des entités dans leur hiérarchie, le voisinage ... Les méthodes structurelles externes exploitent des relations entre des entités elles-mêmes, qui sont souvent des relations de subsomption (is-a ou spécialisation) ou de méréologie (part-whole). Avec ces relations, les entités sont considérées dans des hiérarchies et la similarité entre elles est déduite de l'analyse de leurs positions dans ces hiérarchies. L'idée de base est que si deux entités sont similaires, leurs voisines pourraient également être d'une façon ou d'une autre similaires. Cette observation peut être exploitée de plusieurs manières différentes en regardant des relations avec d'autres entités dans des hiérarchies. Deux entités peuvent être considérées similaires si [Bach, 2006] :

- Leurs super-entités directes (ou toutes leurs super-entités) sont similaires.
- Leurs sœurs (ou toutes leurs sœurs, 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.

Cependant, cette approche peut rencontrer quelques difficultés dans les cas, où les hiérarchies sont différentes au niveau de granularité. Par exemple, si dans une hiérarchie, l'entité « Personne » a deux sous-entités « Enfant » et « Adulte », et si dans une autre hiérarchie, la même entité « Personne » est divisée en deux autres sous-entités « Femme » et « Homme », la déduction que « Enfant » et « Femme » ou « Enfant » et « Homme » sont similaires, est incorrecte dans tous les cas. L'alignement qui utilise ces méthodes est très performant parce qu'il prend en compte toutes les relations entre les entités ce qui nécessite l'utilisation d'autres méthodes terminologiques ou structurelles internes.

### 3.4.3 Les techniques extensionnelles

Ces méthodes déduisent la similarité entre deux entités qui sont notamment des concepts ou des classes en analysant leurs extensions, c.-à-d leurs ensembles des instances. Lorsque deux concepts représentent un même ensemble d'instances, il y a de fortes chances que ces concepts soient similaires. À l'inverse, deux concepts pour lesquels il n'existe aucune instance commune ont peu de chances d'être similaires [Larson et al. 1989] et [Sheth et al. 1988]. Cependant, cette analyse n'a de sens que si les deux ontologies comparées décrivent des ensembles de données identiques. Pour cela, il est nécessaire,

de disposer de deux ontologies à comparer peuplées avec un jeu de données identiques. Généralement, le savoir de la partie extension des ontologies est inestimable pour l'alignement car elle est indépendante de toute conceptualisation. cependant si l'ensemble des instances n'est pas disponible, ces méthodes ne pourraient pas s'appliquer. dans tel cas. les autres techniques sont plus adéquates.

#### 3.4.4 Les techniques sémantiques

Les méthodes sémantiques utilisent la sémantique de la théorie des modèles. Ce sont des méthodes déductives dont la mise en œuvre nécessite de disposer au départ d'alignements initiaux.

#### 3.4.5 Les méthodes hybrides

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

### 3.5 Application

L'alignement d'ontologies revêt toute son importance dans des applications nécessitant la prise en compte d'une interopérabilité sémantique. Voici quelques exemples d'utilisation de l'alignement des ontologies :

- La Communication entre agents : L'alignement d'ontologies permet aux systèmes multi-agents de s'attaquer au problème d'hétérogénéité sémantique dans la communication où chaque agent a sa propre représentation du monde [Kammoun, 2013c].
- L'intégration des web services : Les ontologies peuvent jouer un rôle primordial pour décrire explicitement la sémantique des services[Kammoun, 2013c].
- Le partage d'informations dans les réseaux P2P sémantiques : Les technologies du web sémantique ne se limitent pas à la recherche des informations. Elles sont aussi utiles pour le partage des informations dans les systèmes pair à pair où chaque paire a sa propre ontologie afin de définir les concepts qu'elle manipule[Kammoun, 2013c].

Les alignements sont également la base de processus complexes appliqués sur les ontologies [Cerqueus .2012] :

- fusion d'ontologies : création d'une nouvelle ontologie à partir d'ontologies existantes [Bruijn et al., 2006] ;
- intégration d'ontologies : intégration d'une ontologie dans une autre ;
- la traduction /transformation d'ontologies : expression d'une ontologie dans le " vocabulaire " d'une autre ontologie

– réconciliation d'ontologies : transformation de deux ontologies dans le but de les uniformiser [Hameed et al. ,2004] .Harmonisation des contenus d'ontologies.

Ils sont également utilisés pour travailler sur les données :

— traduction de données : intégration des instances d'une ontologie dans une autre ontologie .

— intégration sémantique de données : possibilité d'interroger les données indépendamment de la manière dont elles sont représentées [Saïs ,2007] ;

### 3.6 Alignement interactif d'ontologies

La réflexion sur l'alignement interactif est née du fait que dans le passé, les recherches sur les outils d'alignement d'ontologies s'étaient focalisées sur le développement d'outils d'alignement entièrement automatiques [Paulheim et al. 2013], il existe une limite en termes de qualité de l'alignement difficile à dépasser par les systèmes d'alignement automatiques d'ontologies. Par ailleurs [Falconer et Noy. 2011] stipulent qu'un utilisateur doit interagir avec un système d'alignement pour examiner les appariements produits par l'outil et indiquer ceux qui sont corrects, ceux qui ne le sont pas, et de créer des correspondances supplémentaires que le système n'a pas trouvées. Toutefois, ce processus nécessite la compréhension des deux ontologies qui seront alignées et les relations qui existent entre elles (comment elles sont reliées l'une à l'autre). Ce processus de validation est une tâche difficile qui nécessite une patience et la connaissance d'un expert du domaine de l'ontologie. Comme l'a déclaré [Falconer et Noy. 2011], l'alignement d'ontologies est « un problème très difficile à la fois pour l'homme et la machine », et doit donc faire appel à des approches semi-automatiques combinant des algorithmes d'alignement automatiques et l'expertise des experts du domaine étudié dans le processus d'alignement avec un minimum d'interaction possible de l'utilisateur car les experts du domaine sont une ressource rare et chère. Ainsi, intégrer l'interaction de l'utilisateur dans les systèmes d'alignement d'ontologies est encore un défi majeur dans l'alignement d'ontologies aujourd'hui [Shvaiko et Euzenat. 2013]. En général, il y a plusieurs possibilités de comment et à quel moment impliquer l'utilisateur dans le processus d'alignement. Cela peut être, soit avant, pendant ou après le processus de mise en correspondance. Des exemples courants d'amélioration de l'alignement d'ontologies en impliquant l'utilisateur comprennent entre autre une définition des configurations du système, la création de correspondances initiales, la correction des correspondances proposées, ou l'évaluation de l'alignement créé. Plusieurs systèmes d'alignement fournissent une configuration qui peut être adaptée par l'utilisateur selon l'alignement à effectuer. Etant donné que définir une configuration

est une tâche difficile pour les experts de domaine (car l'utilisateur est un expert du domaine des ontologies étudiées mais pas de l'application qu'il utilise), certaines approches demandent plutôt aux utilisateurs des exemples de correspondances ou la validation de correspondances produites [Ritze et Paulheim, 2011] et [Feng Shi et al, 2009]. Certains systèmes interactifs impliquent l'utilisateur en demandant la validation des correspondances créées [Duan et al. 2010], [Jiménez-Ruiz et al. 2012], [Lambrix et Kaliyaperumal, 2013] et [Lambrix et Tan, 2006] ou demandent une première liste de correspondances et non correspondances [Ichise et al. 2009]. D'après [Paulheim et al, 2013], de telles approches sont capable de surpasser les systèmes d'alignement automatique (en termes de qualité de l'alignement produit). Les chercheurs ont développé un certain nombre de système faisant intervenir l'utilisateur dans le processus d'alignement. Dans le modèle standard introduit par [Euzenat et Shvaiko ,2007], les systèmes d'alignement d'ontologie prennent deux ontologies et un alignement (facultatif) en entrée, et, éventuellement certains paramètres et de ressources externes et fournissent un alignement final. Ce modèle est étendu afin d'inclure l'interaction avec un utilisateur [Paulheim et al, 2013]. La Figure 2.2 dépeint le processus d'alignement interactif, pendant le déroulement du processus on a des suggestions d'alignement ( $A_{sug}$ ) proposé par le système à l'utilisateur et celui-ci effectue des corrections ( $A_{corr}$ ).

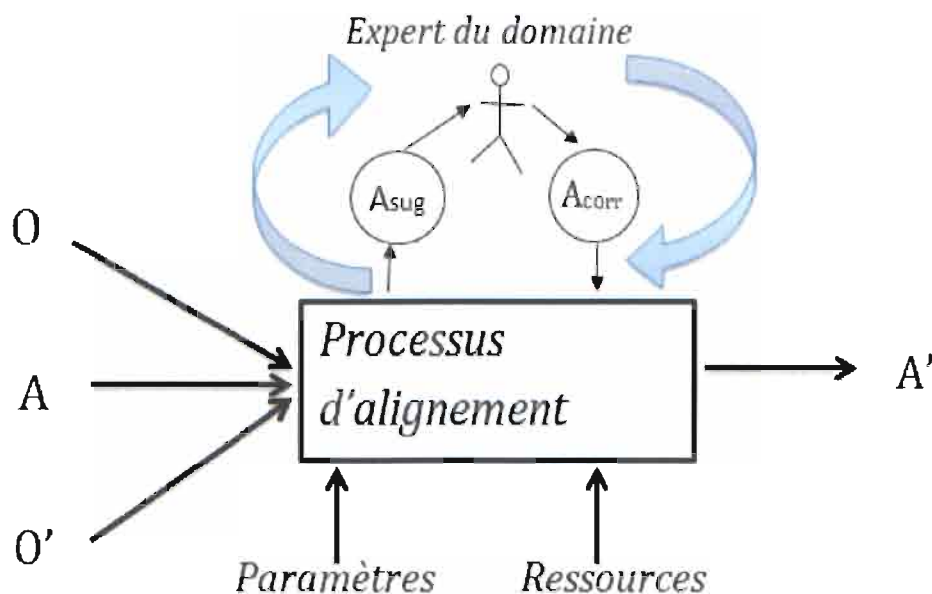


FIGURE 2.2 – schéma de processus d'alignement interactif

### 3.7 Qualité de l'alignement

Les mesures de Précision, Rappel et F-mesure ont été des métriques largement exploitées pour estimer la qualité des alignements obtenus. L'OAEI<sup>1</sup>

1. Ontology Alignment Evaluation Initiative

retient ces mesures pour l'évaluation de la qualité de l'alignement. L'objectif principal de ces mesures est l'automatisation du processus de comparaison des méthodes d'alignement ainsi que l'évaluation de la qualité des alignements produits. La première phase dans le processus d'évaluation de la qualité de l'alignement consiste à résoudre le problème manuellement. Le résultat obtenu manuellement est considéré comme l'alignement de référence. La comparaison du résultat de l'alignement de référence avec celui de l'appariement obtenu par la méthode d'alignement produit trois ensembles : **Nfound**, **Nexpected** et **Ncorrect**.

- L'ensemble **Nfound** représente les paires alignées avec la méthode d'alignement.
- L'ensemble **Nexpected** désigne l'ensemble des couples appariés dans l'alignement de référence.
- L'ensemble **Ncorrect** est l'intersection des deux ensembles **Nfound** et **Nexpected**. Il représente l'ensemble des paires appartenant à la fois à l'alignement obtenu et l'alignement de référence.

La précision est le rapport du nombre de paires pertinentes trouvées, i.e., "**Ncorrect**", rapporté au nombre total de paires, i.e., "**Nfound**". Il renvoie ainsi, la partie des vraies correspondances parmi celles trouvées.

Ainsi, la fonction Précision est définie par :

$$\text{Précision} = \frac{|N_{correct}|}{|N_{found}|}, 0 \leq \text{Précision} \leq 1$$

Le Rappel est le rapport du nombre de paires pertinentes trouvées, "**Ncorrect**", rapporté au nombre total de paires pertinentes, "**Nexpected**". Il spécifie ainsi, la part des vraies correspondances trouvées. La fonction rappel est définie par :

$$\text{Rappel} = \frac{|N_{correct}|}{|N_{expected}|}, 0 \leq \text{Rappel} \leq 1$$

La F-mesure combine la précision et le rappel avec une pondération

$$\text{F-mesure} = \frac{\text{Précision} \times \text{Rappel}}{\text{Précision} + \text{Rappel}} \times 2, 0 \leq \text{F-mesure} \leq 1$$

### 3.8 OAEI challenge

OAEI (Ontology Alignment Evaluation Initiative) est une initiative internationale lancée en 2004 pour coordonner et organiser des évaluations sur le nombre croissant des systèmes d'alignement d'ontologies. Chaque année une campagne d'évaluation est organisée. Les buts de cette initiative sont de :

- Dégager les forces et les faiblesses des systèmes d'alignement.
- Comparer les performances et les techniques.
- Augmenter la communication entre les développeurs.



Alors le but principal est de faire évoluer les techniques, les recherches et les travaux sur les systèmes d'alignement.

## **Conclusion**

J'ai présenté dans ce chapitre un état de l'art des notions liées à l'alignement d'ontologies et en particulier l'alignement interactif qui vise à améliorer la qualité des résultats d'alignement grâce notamment à l'introduction dans le processus d'alignement de l'utilisateur expert du domaine. Cette présentation me permet de donner une vue d'ensemble du domaine auquel appartient mon thème.

## Chapitre 3

# ETUDE DE L'EXISTANT ET CAPTURE DES BESOINS

### Introduction

Ce chapitre contient deux sections intitulées respectivement « étude de l'existant » et « spécification des besoins ». Dans la première section je présente le système ServOMap et son processus d'alignement et dans le second je présente une capture des besoins des utilisateurs auxquels mon application doit pouvoir répondre.

### 1 Etude de l'existant

#### 1.1 Etude du système ServOMap

ServOMap est un système d'alignement d'ontologies à large échelle développé par l'ERIAS de l'Université de Bordeaux en France. Il est capable de traiter des ontologies de grosses tailles (ontologies contenant plusieurs centaines de milliers d'entités). Il charge les ontologies décrites en RDF(S) et le langage standard OWL. ServOMap repose sur un système de gestion conjointe de plusieurs ontologies appelé ServO (pour serveur d'ontologies). Ce système s'appuie sur l'utilisation des techniques de la recherche d'information (RI) et des technologies du Web Sémantique (WS) [Diallo, 2006]. Ainsi, ServOMap se base sur des techniques de RI pour calculer les similarités entre les entités [Diallo et Ba, 2012]. Il a été classé parmi les 3 meilleurs systèmes lors du challenge international « Ontology Alignment Evaluation Initiative (OAEI) » en 2012. Pour le traitement d'ontologies de grande taille, ServOMap repose sur une stratégie d'indexation pour réduire l'espace de recherche et calcule un ensemble initial de candidats basé sur la description terminologique des entités des ontologies en entrée [Diallo et Kamoun, 2013].

## 1.2 Processus d'alignement de ServOMap

L'alignement d'ontologies est le processus de découverte des correspondances entre deux ontologies différentes. Le résultat de ce processus, c'est-à-dire l'expression des correspondances, représente un ensemble de couples d'entités. Le calcul de cet ensemble se fait sur plusieurs étapes et en utilisant différentes approches et techniques. Certaines correspondances sont dégagées grâce à une similarité lexicale et d'autres à partir de la ressemblance du contexte entre les entités à aligner. La Figure 3.1 montre les différentes étapes d'alignement des ontologies avec le système ServOMap. Un alignement contient 3 types de correspondances : alignement des concepts, alignement des propriétés et l'alignement des instances [Kammoun, 2013a].

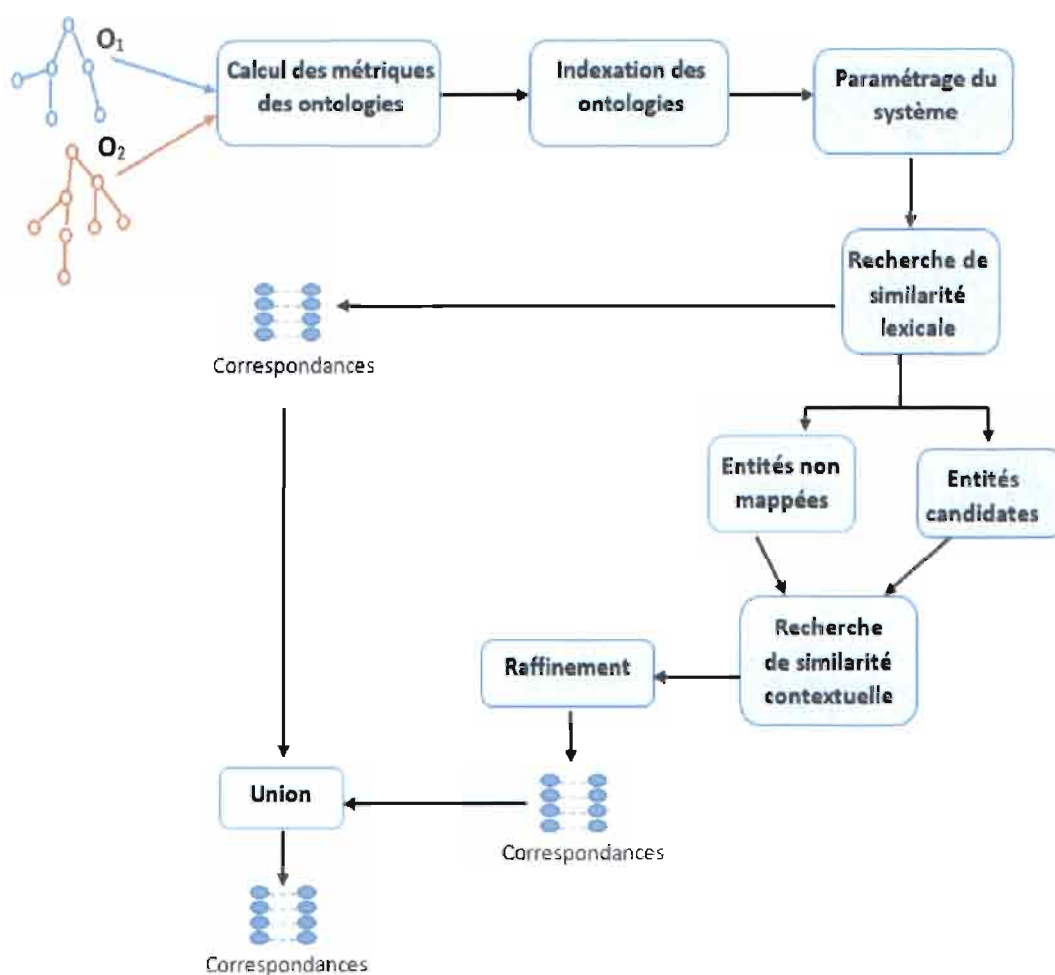


FIGURE 3.1 – Processus d'alignement avec ServOMap

**Calcul des métriques :** La première étape après le chargement d'une ontologie consiste à calculer les métriques (Métadonnées) qui permettent de décrire cette ontologie en termes de nombre de concepts, d'individus, de propriétés, d'axiomes, etc.

**Indexation d'ontologie :** ServOMap traite l'ontologie comme étant un ensemble de documents. Chaque unité (Concept + Propriété + Instance) représente un document à traiter. Pour faciliter le traitement de ces unités

ServOMap construit pour chaque ontologie un système d'indexation. Il utilise les informations dégagées des différents concepts, propriétés et instances de l'ontologie. Puis, en se basant sur les métadonnées calculées précédemment, ServOMap génère une description dynamique des entités. Cette étape d'indexation a pour but d'optimiser le temps de recherche plus tard.

**Paramétrage du système** : cette étape consiste au choix du type d'alignement (concept, propriété, instance), le choix de la précision pour la recherche dans les index (double sens ou sens unique) et un chemin de fichier dans lequel sera stocké le résultat de l'alignement.

**Alignement lexical** : il s'agit de dégager les similarités entre les différentes entités des deux ontologies en se basant sur le vocabulaire relié à ces entités. Le serveur d'alignement génère pour chaque entité dans la première ontologie une requête composée des différents termes qui la définissent. Puis, il cherche dans l'index de la deuxième ontologie pour déterminer les entités qui possèdent une similarité avec la requête générée. Ces entités trouvées représentent des candidats qui peuvent être alignés avec l'entité de base. Le choix des correspondances retenues parmi les candidates se base sur le score de similarité calculé avec une combinaison de méthodes de détermination de similarité entre les chaînes de caractères. On peut également retenir un seul alignement pour chaque entité ou bien plusieurs.

**Alignement contextuel** : Généralement, l'alignement lexical a une très bonne précision. Mais il reste généralement des correspondances à dégager encore. Pour cela, ServOMap utilise l'alignement à base contextuel. L'idée générale repose sur le fait que deux concepts synonymes ont une très grande probabilité d'avoir un entourage similaire. Ainsi, un concept est décrit par l'ensemble de ses Concepts fils, Concepts Pères et Concept frères. Cette approche est appliquée seulement sur les concepts et non sur les propriétés. Elle ne concerne que les concepts non encore mappés lors de l'alignement lexical.

**Raffinement** : Cette étape permet de s'assurer de la précision de l'alignement trouvé. En général, les deux premières méthodes permettent de dégager la plupart des alignements corrects mais il résulte aussi une perte de précision à cause des correspondances non consistantes. Pour remédier à ce problème, les résultats passent par une phase de nettoyage. Il s'agit d'éliminer dans le résultat tous les couples de correspondances qui sont en contradiction avec ceux de l'alignement exact. Le processus d'alignement fourni un ensemble de couples comme résultat.

De façon succincte le processus d'alignement se déroule de la manière suivante :

En premier lieu, un processus de recherche des correspondances lexicales est lancé. Ce processus donne 2 résultats : un ensemble d'alignements exacts et un ensemble de candidats. Ensuite, on passe à l'alignement sur une base

contextuelle, ce processus s'applique sur les entités non mappés lors de la première étape. Il prend en paramètre également les candidats dégagés lors de l'alignement lexical pour vérifier leurs consistances. Après cette phase de recherche contextuelle, les alignements trouvés passent par une phase de raffinement pour éliminer les correspondances non cohérentes afin de ne pas perdre la précision. Puis, l'intersection de ces deux ensembles résultats est nettoyée par un analyseur d'incohérences sur les correspondances dégagées, ici le client du LogMap [Kammoun, 2013b], qui permet d'éliminer toute incohérence. Dans chaque sens on prend l'une des deux ontologies comme référence et on génère des requêtes de recherche dans l'index de la deuxième ontologie. Ces deux processus d'alignement donnent deux résultats. Le résultat final est l'ensemble résultant de l'intersection de ces deux sous-ensembles [Kammoun, 2013b].

### 1.3 Limites du système et besoin en interactivité

Comme décrit précédemment, le fonctionnement du système ServOMap est entièrement automatique. Cependant plusieurs étapes peuvent bénéficier d'une intervention humaine afin de produire des résultats de meilleure qualité. Par exemple, si les ontologies en entrée ne présentent pas suffisamment de similarité terminologique, les résultats du système se trouvent affectés. Ainsi, à défaut d'une disponibilité d'un alignement partiel initial, une intervention humaine peut proposer un ensemble d'alignement de départ pour le calcul de la similarité contextuelle. De même, au lieu d'une sélection uniquement automatique des meilleurs couples candidats, une validation humaine sur certains couples peut améliorer la qualité des résultats en augmentant le taux de rappel sans perdre en précision. C'est le but du présent travail.

## 2 Spécification des besoins

### 2.1 Les besoins fonctionnels

La capture des besoins fonctionnels s'intéresse à l'architecture du système d'un point de vue fonctionnel. Elle consiste à expliciter les besoins des différents utilisateurs du système. Ces besoins permettent de déterminer les principales fonctionnalités que le système devra assurer. Ces fonctionnalités sont mises en exergue par les cas d'utilisation du système. Par le biais de ces cas d'utilisation, nous aurons un œil permanent sur les besoins exprimés et ainsi éviter de trop s'éloigner des besoins réels des utilisateurs. La phase de spécification constitue la base de départ de la conception.

### 2.1.1 Cas d'utilisation

Le système ServOMap est un module du système ServO chargé de faire des alignements sur les ontologies. ServO est lui-même composé de trois modules interdépendants : un module de gestion d'ontologie (OMM : Ontologie Manager Module), un module d'indexation et de Recherche (IM : Indexing Module) et un module d'alignement d'ontologies (Matching Module : MM). Ces modules sont également considérés comme des serveurs, puis qu'ils permettent la création et le stockage des résultats des différentes tâches exécutées par chacun sous forme de texte ou de fichier XML.

- OMM : il s'agit du module qui permet le chargement et la manipulation des ontologies.
- IM : c'est le module d'indexation et de recherche sur les ontologies. Il se base sur la bibliothèque Lucene : il traite les entités comme un ensemble de documents afin de faciliter et d'accélérer la recherche. C'est l'intermédiaire entre le module de chargement et le module d'alignement.
- MM : il a pour rôle d'aligner selon plusieurs axes deux ontologies en entrée. Il utilise le serveur d'index pour extraire les informations et chercher dans une ontologie.

Les acteurs de ce système sont divisés en 2 catégories :

**Acteurs externes** : qui sont les utilisateurs du système. Ils peuvent bénéficier directement des services de l'application en interrogeant chacun de ses serveurs.

**Acteurs internes** : les acteurs internes sont : le serveur de gestion des ontologies (OMM), le serveur d'index (IM) et le serveur d'alignement (MM). La représentation des cas d'utilisation est faite à travers un diagramme de cas d'utilisation UML. Ce diagramme permet de recueillir, d'analyser et d'organiser les besoins et de recenser les grandes fonctionnalités d'un système. Le diagramme de cas d'utilisation du système ServOMap est représenté par la figure 3.2.

### 2.1.2 Diagramme de séquence

Nous présentons dans cette section, un scénario d'alignement interactif de deux ontologies à travers un diagramme de séquence système (Figure 3.3). Le diagramme de séquence système considère le système comme un tout, représente les messages échangés entre ce système et les acteurs tout en mettant en exergue l'enchaînement chronologique des opérations.

## 2.2 Les besoins non fonctionnels

Les besoins non fonctionnels agissent comme des contraintes sur les solutions, mais leur prise en compte fait éviter plusieurs incohérences dans le

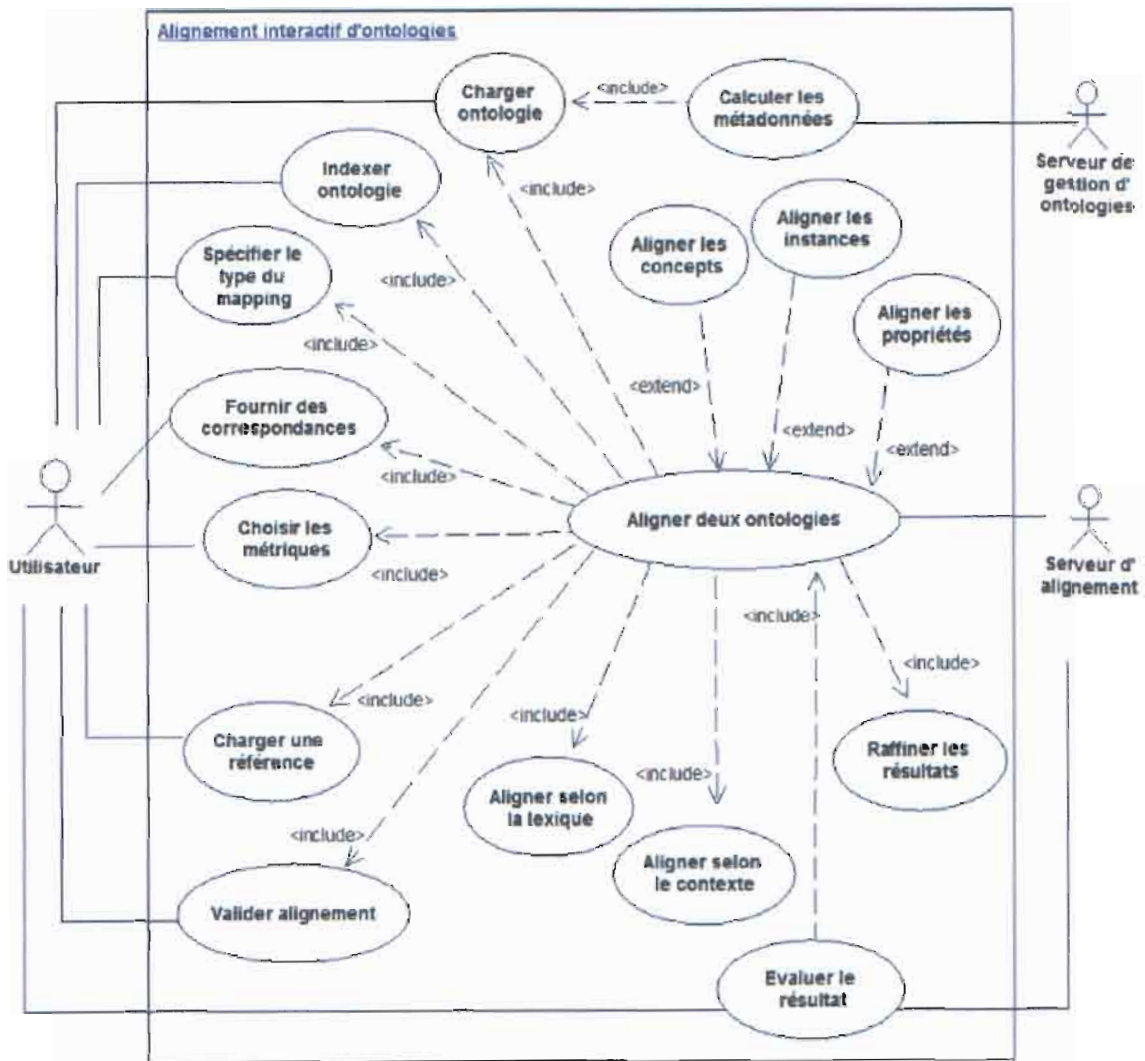


FIGURE 3.2 – diagramme de cas d'utilisation de l'alignement interactif d'ontologies

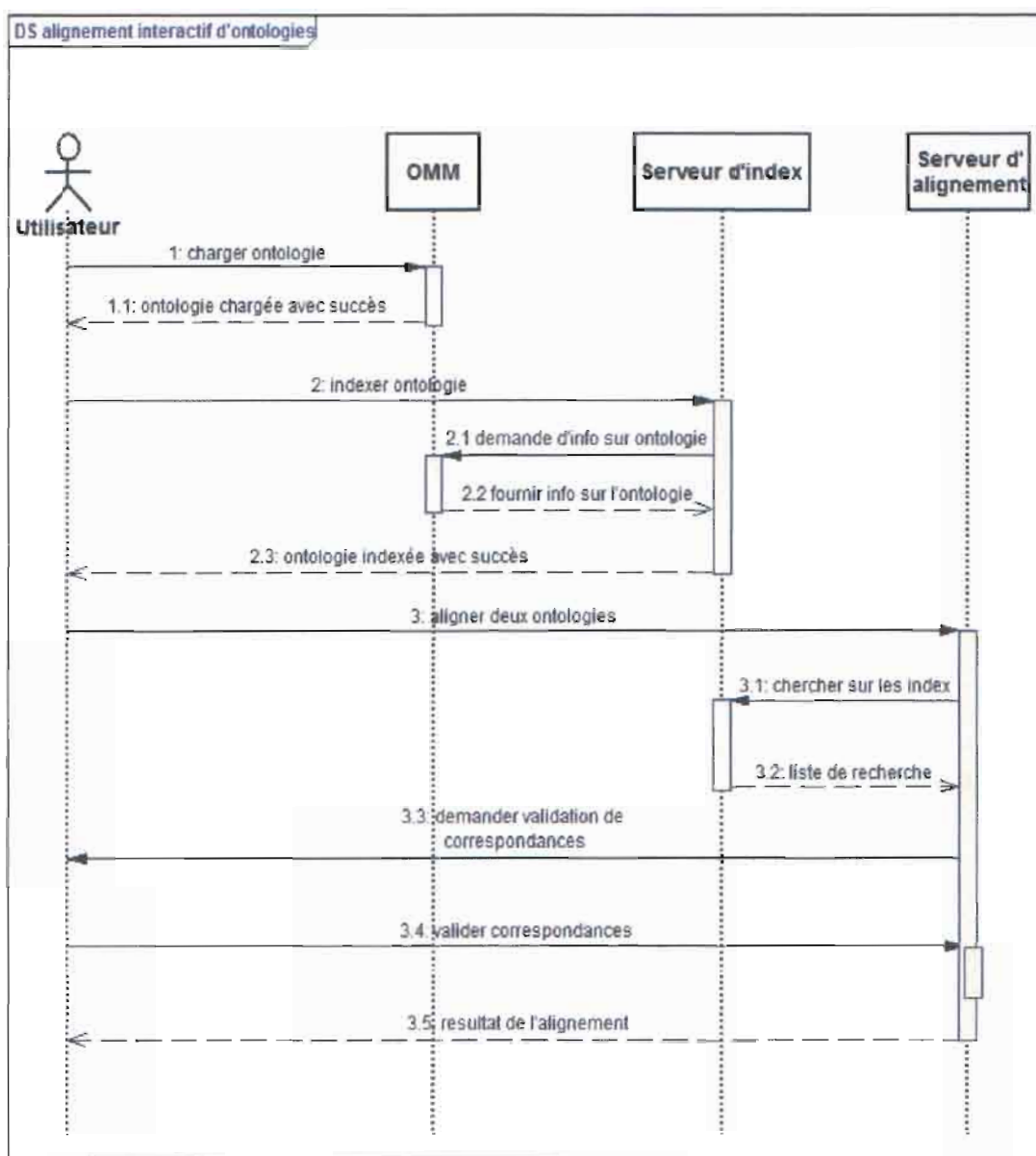


FIGURE 3.3 – diagramme de séquence de l'alignement interactif d'ontologies



système. En dehors de l'aspect fonctionnel, le système d'alignement interactif ServOMap doit répondre à certains besoins non fonctionnels.

- L'ergonomie : l'application doit offrir des interfaces simples et pratiques illustrant d'une manière claire les différentes étapes de l'alignement d'ontologies ;
- La simplicité : elle consiste à offrir aux utilisateurs une application simple à manipuler et une structuration de navigation claire ;
- La performance : les données manipulées par l'application peuvent être de taille très importante. L'application doit donc fournir un temps de réponse minimal lors des différents traitements.

### 3 Conclusion

Mon travail consiste à faire évoluer le système ServOMap en intégrant une interactivité dans son processus d'alignement d'ontologie. Il était donc nécessaire de faire une présentation de ce qu'était le système avant mon intervention et de dégager les besoins auxquels la nouvelle version de l'application doit satisfaire.

# Chapitre 4

## CONCEPTION

### Introduction

Ce chapitre est divisé en deux points. Le premier point intitulé, « analyse conceptuelle du système » présente en détaille la conception de notre système à travers des diagrammes de modélisations. Le second point présente la mise en œuvre du système à travers l'environnement de travail, les outils logiciel et plateforme de développement utilisés. Nous avons utilisé le langage de modélisation UML pour la modélisation et la présentation sous forme de diagramme du système et de ses composants.

### 1 Analyse conceptuelle du système

Dans le processus d'alignement, l'utilisateur peut interagir de plusieurs façons. Il peut fournir des valeurs de similarité entre entités (en déterminant par exemple un certain nombre de correspondances). Ces données servent alors à amorcer et à faciliter les méthodes automatiques d'alignement. Ensuite l'utilisateur peut intervenir dans le choix des méthodes à appliquer et l'ordre dans lequel elles doivent être appliquées. Enfin l'utilisateur peut porter un jugement sur les résultats produits par un processus d'alignement. Ce jugement sert à corriger le résultat, soit pour l'utiliser directement, soit pour le fournir en entrée d'une autre méthode.

Nous prévoyons dans notre approche l'intervention de l'utilisateur sur le résultat de chaque étape de calcul de similarité dans le processus d'alignement et l'obtention de résultats intermédiaire après chaque étape. Ainsi nous définissons l'intervention de l'utilisateur par les étapes suivantes :

#### ★ Paramétrage du système

Le paramétrage se caractérise par le choix de type d'alignement (selon les concepts, propriétés, individus), la précision pour la recherche dans les indexes en sens unique ou dans les deux sens et dans le souci de rendre le système flexible nous avons défini un choix entre le processus interactif et le processus automatique (non interactif).

★ Proposition de correspondances

L'utilisateur peut décider de proposer des correspondances d'entités qu'il juge que le système n'est pas susceptible de trouver. Il peut s'agir également d'abréviations ou de synonymes spécifiques au domaine.

★ Choix des méthodes de calcul de similarité

A ce niveau l'utilisateur a face à lui un certain nombre de méthodes de calcul de similarité et peut choisir celles qu'il veut utiliser pour le calcul des similarités dans le processus. Les méthodes choisies par l'utilisateur seront combinées entre elles s'il y en a plus d'une dans son choix. L'utilisateur peut également définir à ce niveau les seuils maximaux et minimaux de fiabilité. Ce seuil permet d'indiquer qu'un couple candidat aligné est correct ou incorrect ; c'est une valeur numérique comprise entre 0 et 1. Plus cette valeur tend vers 1, plus le couple est plus fiable. A l'inverse, plus elle tend vers 0, plus le couple est douteux et son appartenance au résultat ne peut plus être décidée par le système et c'est là que les connaissances de l'expert du domaine sont sollicitées pour aider le système à décider.

★ Validation d'alignement candidats

La validation dans notre cas est l'étape à laquelle le système est face à un ensemble de couples alignés dont il ne peut décider de leur exactitude et fait donc appel à l'expertise de l'utilisateur. Le système sélectionne tous les couples qu'il juge douteux selon le niveau de fiabilité défini par l'utilisateur au départ et lui demande de décider si ces derniers sont corrects ou incorrects. La sélection s'effectue sur le résultat de l'alignement proposé par le système à une étape donnée.

### 1.1 Processus d'alignement interactif du système

La figure 4.1 décrit le scénario de l'alignement d'ontologies du système ServOMap prenant en compte l'intervention de l'utilisateur.

Le processus d'alignement débute par une phase de paramétrage du système dans laquelle l'utilisateur choisit le type d'alignement (Concept, Propriétés, instances), le mode d'exécution (automatique ou interactif) car pour garder une flexibilité dans notre programme nous avons intégré deux modes d'exécution ; automatique qui est le mode d'exécution classique du système d'alignement mais avec la possibilité de la prise en compte de certains paramètres définis par l'utilisateur tels que le pré mapping (correspondances proposées au départ), l'utilisation d'une référence et le mode interactif. L'utilisateur choisit également la précision pour la recherche dans les index (double sens ou sens unique). Une fois le mode d'exécution interactif choisi, l'utilisateur, toujours dans le paramétrage, choisit les différentes méthodes de calcul

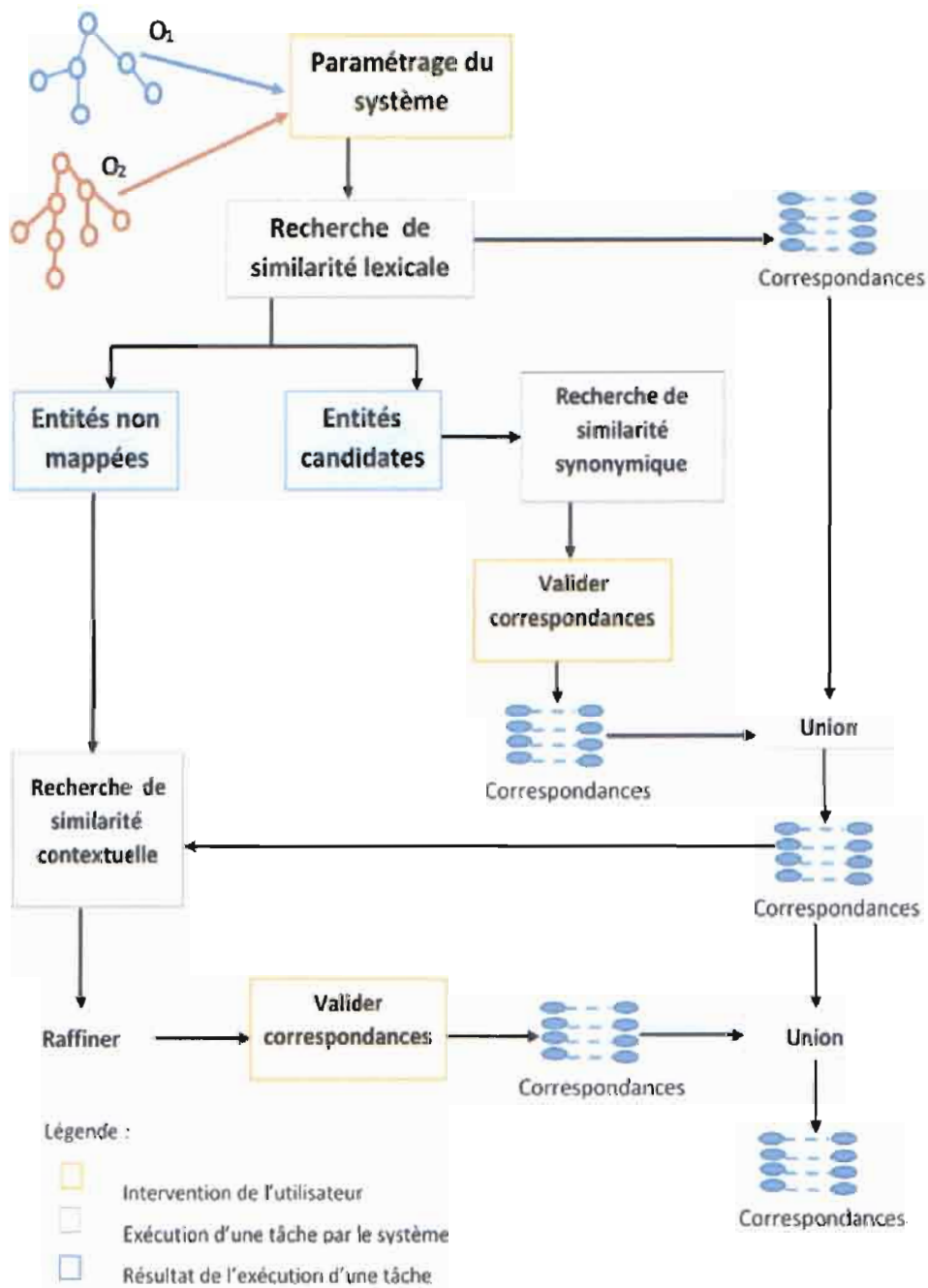


FIGURE 4.1 – Processus d'alignement interactif du système

de similarité qu'il compte utiliser pour la recherche de similarité tout au long du processus et défini ensuite les différents seuils de confiance que le système doit utiliser pour juger de la nature (correct, incorrect ou douteux) de chaque couple d'entité aligné. Lorsque l'utilisateur valide cette étape, le système effectue une recherche de similarité lexicale basée sur le vocabulaire des entités et produit un ensemble de correspondances exactes, un ensemble d'entités candidates susceptibles d'être alignées (couples de correspondances douteux) et un ensemble d'entités non alignées (qui n'ont eu de correspondances). Les entités candidates sont passées à une étape de recherche étendue ; cette recherche permet de retrouver les couples dont les entités sont des synonymes et deux entités synonymes forment un couple correct. A l'issue de cette étape de recherche de similarité synonymique le système sélectionne tous les couples dont la valeur de confiance est comprise entre le seuil minimal et le seuil maximal définis par l'utilisateur et demande à ce dernier de les valider. Les correspondances obtenues à ce niveau sont ajoutées à celles de l'alignement lexical. Le système passe ensuite à une phase de recherche de similarité basée sur le contexte. Cette étape de recherche prend en entrée les entités non mappées lors de l'alignement lexical et le résultat de l'alignement courant et produit un alignement qui après avoir été nettoyé par la phase raffinement est soumis à une validation de l'utilisateur, le résultat obtenu après cette validation est ensuite ajouté au résultat de l'alignement en cours pour obtenir l'alignement final.

## 1.2 Diagramme de composant du système

La figure 4.2 définit les différentes dépendances entre les composants du système. Elle représente également l'architecture physique et statique de l'application. Le module des données représente les données manipulées et générées par le système au cours de son exécution.

**OntologiesFiles** : représentent les fichiers chargés par le système et contiennent plusieurs types de fichiers par exemple : fich.rdf, fich.owl, fich.obo.

**MappingFiles** : contiennent les fichiers représentant les résultats des alignements effectués par le système. Ces fichiers contiennent un ensemble de couples d'URI avec un score pour chaque couple. Ils peuvent être soit un fichier texte ou bien un fichier «.rdf».

**ReferencesMappingFiles** : contiennent tous les alignements corrects utilisés comme références.

**fr.grisa.OntologyManager** : responsable du chargement et de la gestion d'une ontologie à partir d'un fichier.

**fr.grisa.servo.IndexEngine** : permet l'indexation de toutes les informations extraites à partir d'une ontologie. Ce composant utilise le module « OntologyManager » pour collecter les informations nécessaires sur l'ontolo-

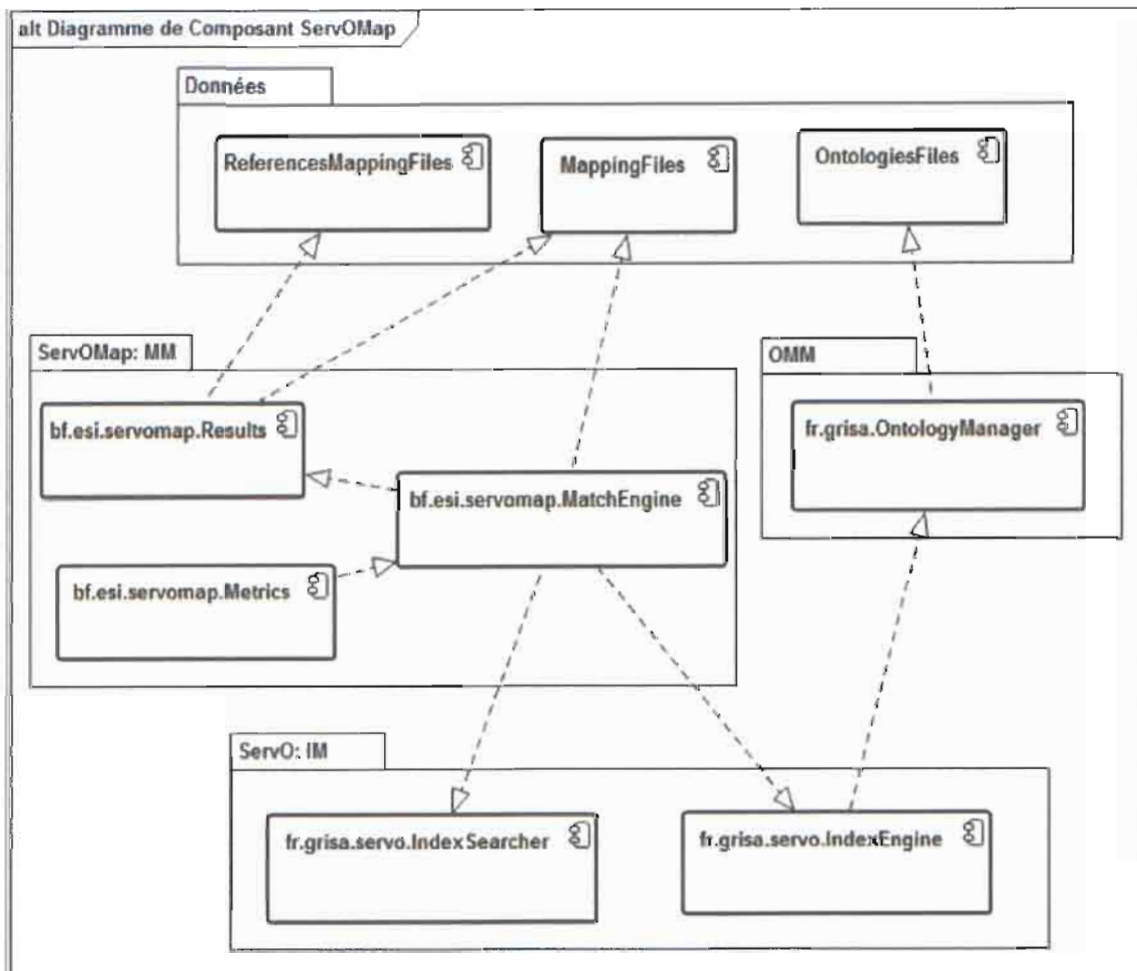


FIGURE 4.2 Diagramme de composant du système

gie pour effectuer l'indexation.

**fr.grisa.servo.IndexSearcher** : contient un générateur de requêtes de recherche sur les index. Il permet plusieurs types de recherche et utilise un serveur de terme pour lire les données à partir d'un index.

**fr.grisa.servomap.MatchEngine** : exécute des algorithmes d'alignement qui se basent sur l'index créé par ServO afin de fournir des fichiers d'alignement. Il contient toutes les stratégies et méthodes utilisées pour effectuer un alignement.

**fr.grisa.servomap.Results** : représente les résultats et les utilise avec certains fichiers de références pour calculer les métriques d'évaluation de l'alignement.

### 1.3 Diagramme de paquets

Les composants peuvent être organisés en paquets, qui définissent des sous-systèmes. Un paquetage est une collection d'éléments d'un modèle. Il permet de structurer un modèle en unités plus petites et plus faciles à gérer. Nous décrivons dans cette partie le diagramme de paquets associés aux composants du module d'alignement précédemment cités (Figure 4.3). Le paquetage « fr.grisa.ServOMap.dictionnaire » implémente une classe qui

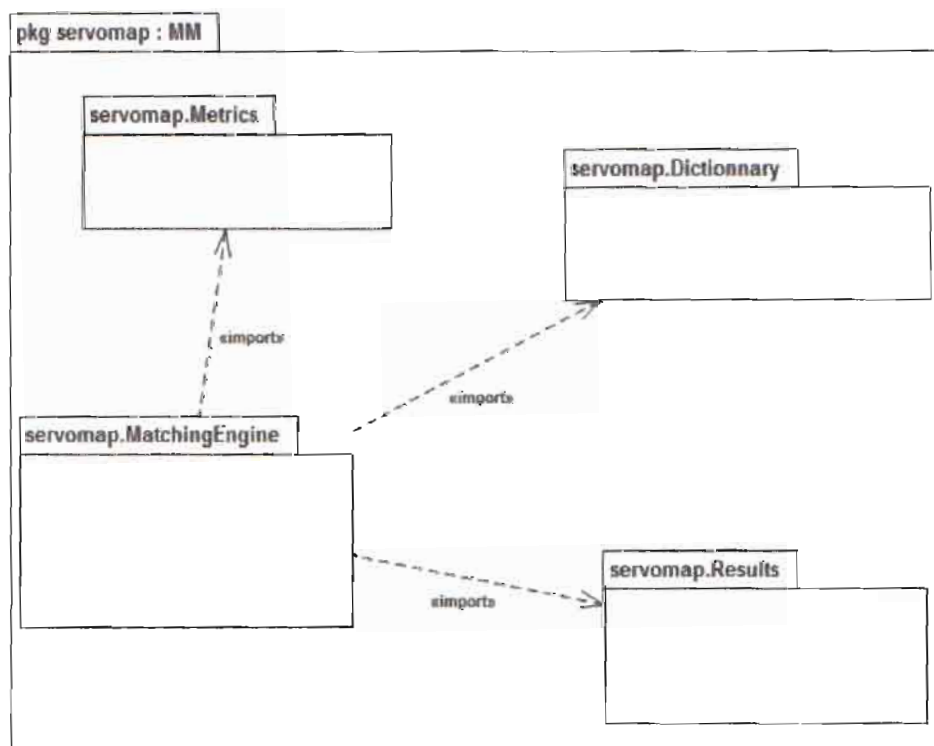


FIGURE 4.3 – Diagramme de paquets de ServOMap

permet l'enrichissement et la traduction des termes d'un index grâce à un dictionnaire. Le paquetage « fr.grisa.ServOMap.Metrics » contient une classe implémentant plusieurs méthodes de calcul de la similarité entre deux chaînes de caractère. Ces méthodes retournent une valeur numérique allant de 0 à 1

qui caractérise le niveau de similarité de ces deux chaînes. Le paquetage « fr.grisa.ServOMap.MatchingEngine » exploite ces méthodes dans ses stratégies d'alignement par une importation.

#### 1.4 Diagramme de classes du système

Dans la figure 4.4, on trouve toutes les classes du module d'alignement divisés sur quatre paquetages. La classe « MatchEngine » est la classe principale du module. Elle est responsable du processus d'alignement. Cette classe implémente tous les algorithmes utilisés pour aligner deux ontologies données. Pour évaluer la similarité entre deux chaînes de caractères, elle fait appel à la classe « Metrics ». De plus, elle utilise la classe « Dictionary » pour enrichir les vocabulaires utilisés lors de l'alignement. Elle stocke les résultats fournis dans un Objet « LResult ». La classe « LResult » n'est qu'une liste de couples d'entités avec un poids. Chaque couple est représenté par un Objet « EResult ». Afin d'évaluer ces résultats d'alignement, la classe « Evaluator » prend en paramètres une liste des résultats et le transforme en fichier texte. Puis, elle compare ce fichier avec un fichier de référence pour dégager les couples manquants et les faux couples. Enfin, l'évaluateur calcule les métriques d'évaluation : la Précision, le Recall, et la F-Mesure.

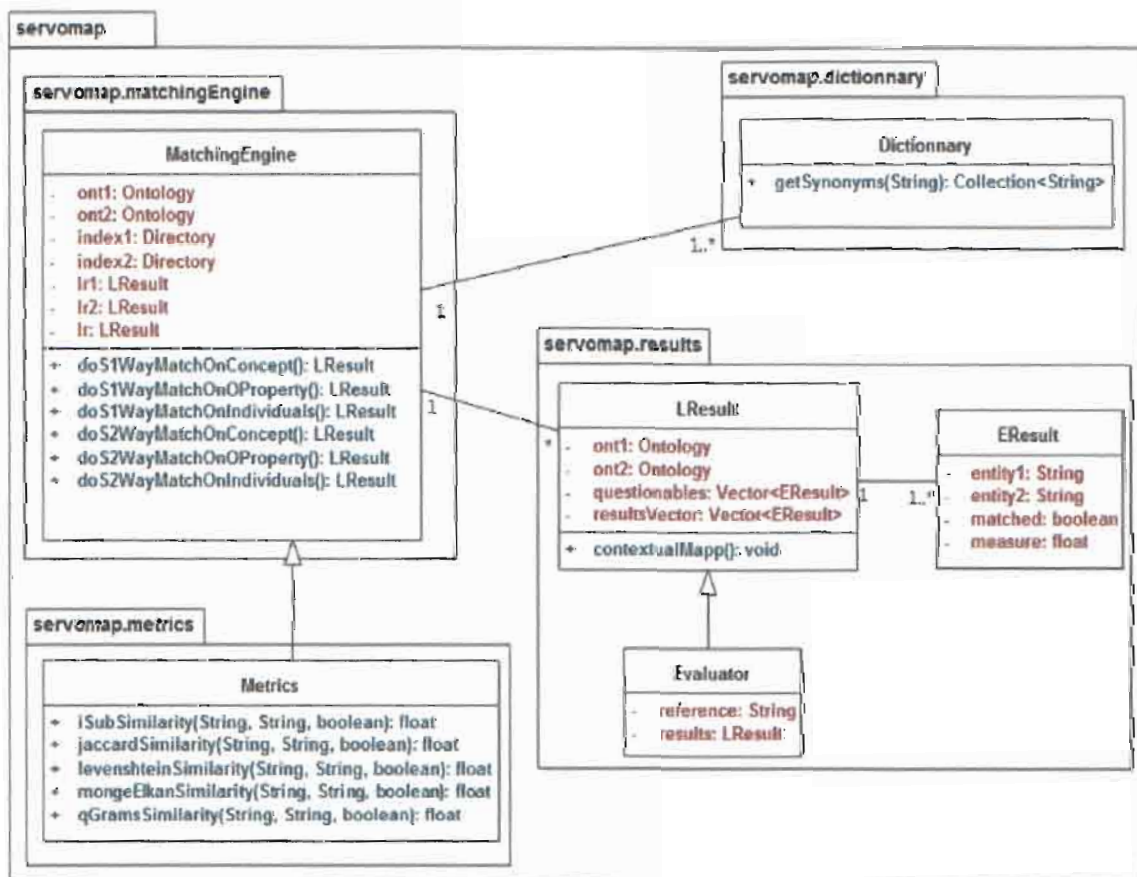


FIGURE 4.4 – Diagramme de classes de ServOMap



## 2 Mise en œuvre du système

### 2.1 Environnement de travail

Cette section présente les outils, technologies et plateforme de développement utilisés pour la mise en œuvre de l'application.

#### 2.1.1 Environnement matériel

L'application a été réalisée sur un ordinateur portable de marque DELL dont les caractéristiques sont les suivantes :

- Système d'exploitation : Windows 7 Professionnel
- Processeur : Intel® Core i3-2370 CPU 2.40 GHz
- Mémoire vive : 4,0 Go.
- Disque Dur : 465 Go.

#### 2.1.2 Langage et environnement logiciel

Pour la mise en œuvre de notre application, le langage de développement utilisé est le langage Java. Ce choix est une suite logique du projet car le langage Java a été utilisé pour la réalisation des bases du projet et de ses versions antérieures. Nous avons également utilisé les outils logiciels suivants :

- L'outil de développement : NetBeans IDE 8.0
- Le serveur d'application : apache-tomcat-8.0.5
- Les outils de modélisation UML : Power amc et Entreprise Architect.
- Latex : La rédaction de ce rapport a été réalisée avec Latex en utilisant l'éditeur Texmaker.
- Protégé : système de création et de gestion d'ontologies.
- Lucene : système de visualisation de recherche sur un index Lucene.

#### 2.1.3 Environnement technique

**Architecture utilisée :** l'architecture de notre application est divisée en trois couches distinctes :

**Une couche de présentation :** encore appelée IHM (Interface Homme Machine), elle permet l'interaction de l'application avec l'utilisateur, la gestion des saisies au clavier et à la souris et la présentation des informations à l'écran. Cette couche contient une logique applicative qui décrit les traitements à réaliser par l'application. C'est l'ensemble de traitements nécessaires pour répondre aux besoins des utilisateurs.

**Une couche métier** qui contient les règles de l'application. Dans notre application cette couche a pour rôle le chargement, l'indexation et l'alignement

des ontologies.

Une couche accès aux données qui regroupe l'ensemble des mécanismes permettant la gestion des informations stockées par l'application. Dans notre cas il s'agit des fichiers owl, rdf ou obo qui représentent les ontologies.

Cette décomposition montre que notre application est développée selon l'architecture 3 tiers qui offre une meilleure flexibilité et souplesse. La Figure 4.5 montre l'architecture de notre application ainsi que les interactions entre ces différentes couches.

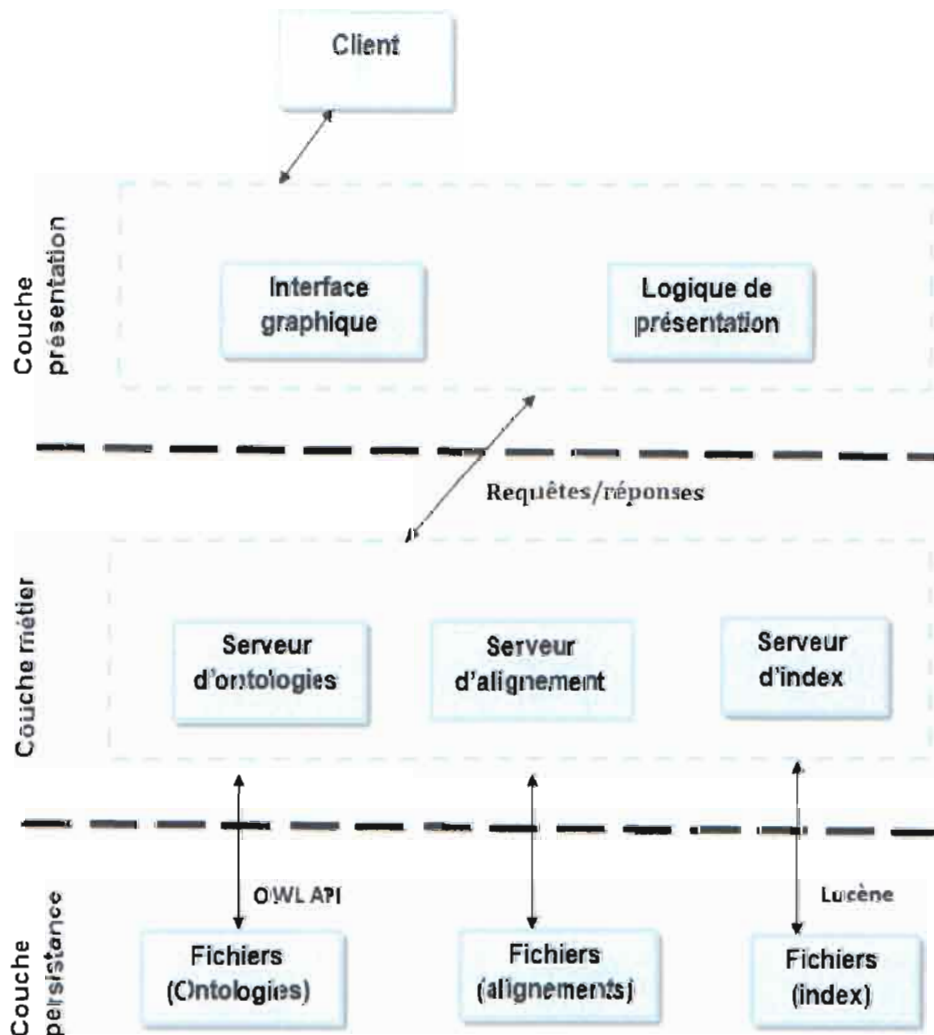


FIGURE 4.5 – Architecture de l'application

**Le Framework JSF** Java Server Faces (abrégé en JSF) est un Framework Java, pour le développement d'applications Web suivant le modèle MVC. Nous avons utilisé dans l'implémentation de notre application le framework JSF 2.2.

**Modèle MVC :** Le patron Modèle-vue-contrôleur (en abrégé MVC, de l'anglais Model-View-Controller) est un modèle destiné à répondre aux besoins des applications interactives en séparant les problématiques liées aux différents composants au sein de leur architecture respective. Ce paradigme

regroupe les fonctions nécessaires en trois catégories :

- un modèle (modèle de données) ;
- une vue (présentation, interface utilisateur) ;
- un contrôleur (logique de contrôle, gestion des événements, synchronisation).

Les deux jeux de composants standards de JSF s'avèrent trop limités et insuffisants pour le développement d'applications d'entreprise. Nous avons utilisé un jeu de composants additionnels qui offrent de nouveaux composants plus riches : PrimeFaces 4.0<sup>1</sup>.

**PrimeFaces** : c'est un jeu de composants open-source supportant Ajax, Jsf 2. Push. L'avantage avec PrimeFaces, c'est qu'on a juste un seul jar qui ne nécessite aucune configuration et aucune dépendance. Il fournit plus de 100 composants développés sur le principe d'AJAX. Les Composants de PrimeFaces sont développés avec un principe de conception qui stipule qu'un composant d'interface utilisateur doit masquer la complexité, mais garder la "souplesse".

## 2.2 Implémentation

### 2.2.1 Page d'accueil de l'application

La Figure 4.6 présente la page d'accueil de notre application. Elle contient une brève présentation du système ServO ainsi qu'un menu latéral et un menu bouton « Options » à droite que nous avons ajouté pour permettre à l'utilisateur de choisir une langue. pour l'instant nous avons défini deux langues français et anglais, l'application fonctionne en français tout comme en anglais. Le menu permet la navigation entre les différentes pages de l'application. L'utilisateur peut naviguer vers les pages de chargement, d'indexation ou de recherche. Il peut également naviguer vers les pages d'alignement des deux ontologies et de chargement de référence que nous avons ajouté au menu pour permettre à l'utilisateur d'utiliser une référence pour la validation et/ou l'évaluation de résultat au cours du processus d'alignement .

### 2.2.2 Scénario de chargement d'une ontologie

Pour réaliser une tâche d'alignement d'ontologies, il faut au préalable charger les ontologies à aligner. Notre système traite des ontologies de très grandes tailles et la taille d'une ontologie est déterminée par le nombre de concepts quelle contient. La Figure 4.7 présente un exemple de chargement d'une ontologie. En effet pour charger une ontologie l'utilisateur doit saisir une « uri » et cliquer sur le bouton « load ». Cela est possible que lorsque vous

---

1. <http://primefaces.org>,

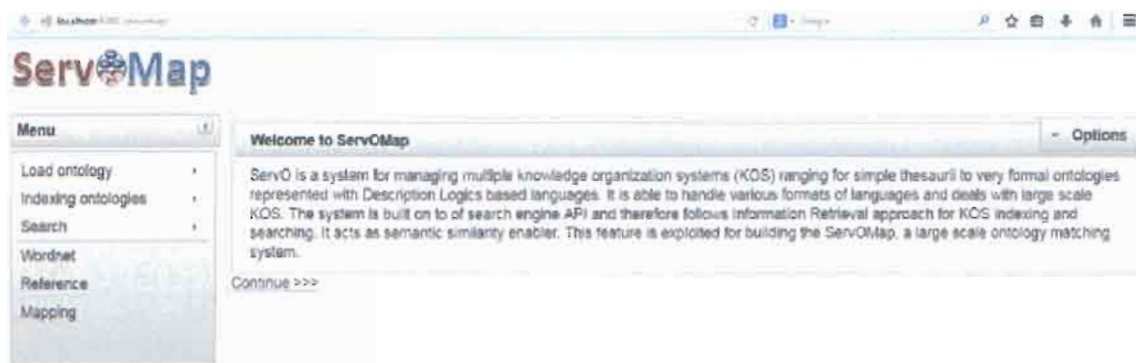


FIGURE 4.6 – page d'accueil de l'application

êtes connectés à internet. Pour permettre l'exécution de l'application sur son poste sans une connexion internet, nous avons implémenté et ajouter un chargement local des ontologies. Cependant il faut que les ontologies à charger soient préalablement stockées sur votre disque dur. Un message de succès de chargement sera affiché si le chargement a réussi et un message d'erreur sinon.

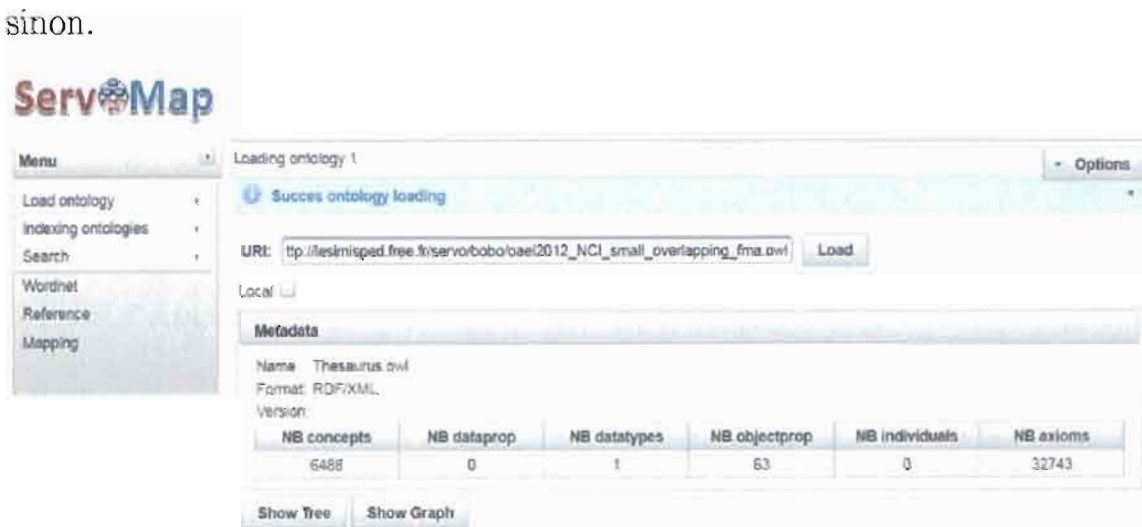


FIGURE 4.7 – page de chargement d'une ontologie

### 2.2.3 Scénario d'indexation d'une ontologie

Avant l'exécution du processus d'alignement, une indexation des ontologies chargées est effectuée. Cette tâche a pour but de réduire le temps de recherche de similarités au cours de l'alignement. L'indexation d'une ontologie pour sur ses concepts, ses propriétés, ses instances ou les métadonnées. Une ontologie est défini par des concepts, des propriétés et des ins-

tances également appelées individus; les concepts contiennent des concepts parents (super concepts), des concepts fils (sous concepts) et des concepts frères (concepts voisins), et chaque concept est défini par un label, une propriété, des commentaires et un contexte. Lorsqu'on choisit d'effectuer une indexation selon les concepts, il faut donc choisir également les paramètres (label, propriétés, etc.) des concepts à indexer. On peut également ajouter les sous concepts, les concepts voisins de ces derniers si l'on veut approfondir l'indexation. En ce qui concerne les instances, leurs descriptions sont en fait leurs désignations (leurs noms), leurs propriétés et leurs types qui peuvent être choisis lorsqu'on désire faire une indexation selon les instances. La Figure 4.8 présente un exemple d'indexation d'une ontologie. Comme indiqué dans la figure, la tâche d'indexation exige la saisie d'un répertoire pour enregistrer cet index sur le disque, ou bien le choix d'indexer seulement dans la mémoire RAM. Il faut cocher ensuite les types d'indexation à effectuer et éventuellement les paramètres qui les accompagnent. Cette opération peut prendre un peu de temps, cela est relatif à la taille de l'ontologie ainsi que les paramètres choisis.

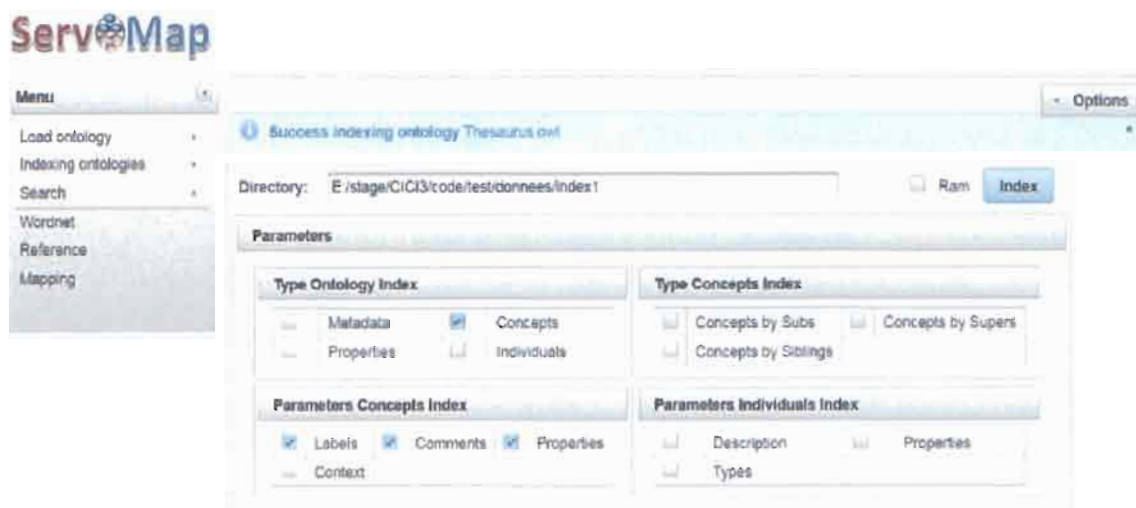


FIGURE 4.8 – paramétrage et indexation d'une ontologie

#### 2.2.4 Scénario de chargement d'une référence

On peut utiliser une référence à la place de l'utilisateur pour valider les correspondances générées par le système. En effet, la référence est un alignement obtenu manuellement et qui contient l'ensemble des correspondances exactes des ontologies à aligner. La Figure 4.9 présente la fenêtre de chargement de l'alignement référence à utiliser dans le processus d'alignement à venir. Pour charger une référence il faut saisir le chemin relatif du fichier de référence et cliquer ensuite sur le bouton « Load ». Si l'on coche le bouton check box « pre mapping » nous obtenons la page présentée par la Figure 4.10.

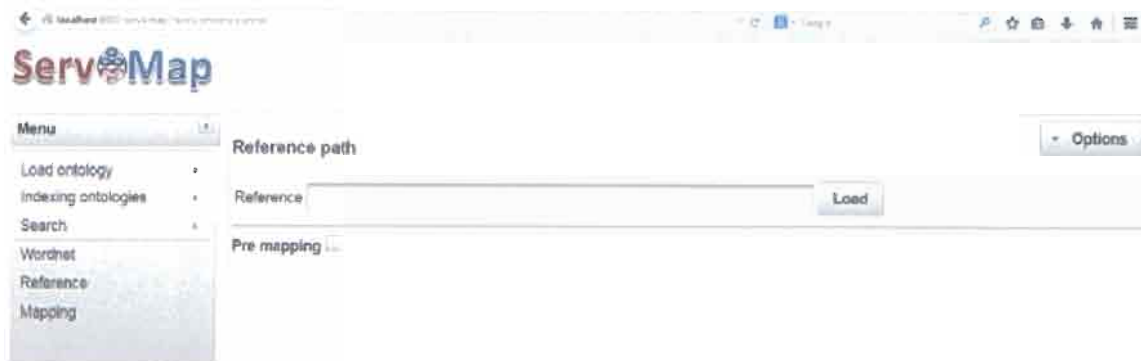


FIGURE 4.9 – page de chargement d'une référence

### 2.2.5 Scénario de proposition de correspondances

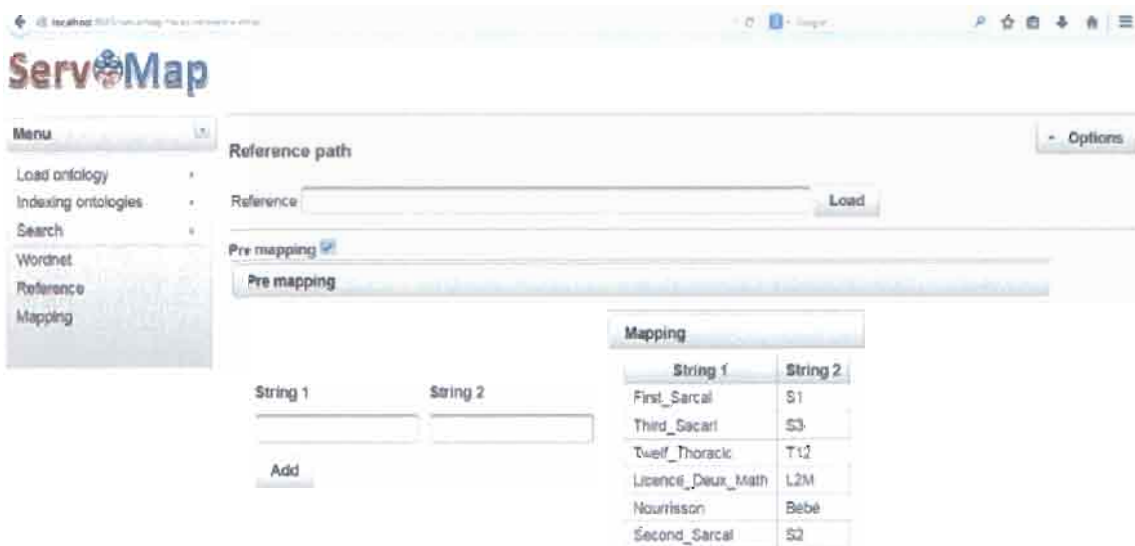


FIGURE 4.10 – page de proposition de correspondances

La Figure 4.10 présente la page de saisie de correspondances par l'utilisateur. Etant donné que l'utilisateur est un expert du domaine, c'est lui qui sait comment certains concepts sont notés (abréviation, synonymes) dans le domaine. Pour certaines commodités ou conventions, les experts d'un domaine peuvent décider de noter ou d'appeler un même concept de manière différente. Pour faciliter l'alignement pour ce genre de cas afin d'augmenter la précision et le nombre de correspondance dans l'alignement, nous permettons à l'utilisateur de saisir une liste de correspondances pour les notations, les abréviations (ex; **First\_sarcal** qui peut être abrégé par **S1**) et des sy-

onymes de ces entités que le système n'est pas susceptible de trouver. Cette liste sera utilisée par les méthodes d'alignements au cours de l'exécution de la tâche d'alignement.

## 2.2.6 Scénario d'alignement de deux ontologies



FIGURE 4.11 – paramétrage de l'alignement

### Paramétrage général du processus d'alignement

Le processus d'alignement débute par une phase de paramétrage au cours de laquelle l'utilisateur peut choisir le mode d'alignement (automatique ou interactif) qu'il désire effectuer. En effet, le système garde une flexibilité et ne contraint pas l'utilisateur à intervenir dans le déroulement du processus d'alignement. La Figure 4.11 présente la page de paramétrage de l'alignement. L'utilisateur spécifie un chemin de fichier dans lequel le résultat sera stocké, ensuite, il choisit le ou les types d'alignement (concepts, propriétés, individus), la précision pour la recherche dans les index et le mode d'alignement, interactif s'il coche « Interactive ») et automatique s'il prend le bouton « Execute » à ce niveau. Si l'utilisateur choisit le mode interactif la fenêtre représentée par la Figure 4.12 apparaît.

### Paramétrage du processus d'alignement interactif

L'approche développée utilise un certain nombre de méthodes pour déterminer les similarités entre les entités des ontologies. Ces méthodes sont essentiellement ngram, jaccard, levenshtein, mongeElkan, isub. Chaque méthode implémente un algorithme de calcul de similarité entre deux chaînes de caractère et fournit une valeur numérique comprise entre 0 et 1 caractérisant le degré de similarité des entités. Lorsque le degré vaut 1 cela signifie

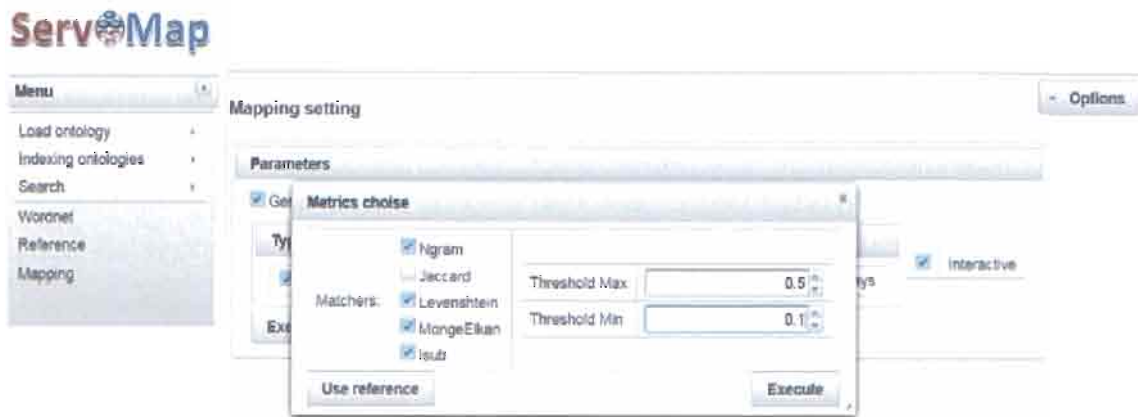


FIGURE 4.12 – paramétrage de l'alignement : choix des métriques et valeurs de confiance

que les chaînes comparées sont similaires (cent pour cent similaire) et lorsqu'il tend vers 0 les chaînes sont moins similaires. La Figure 4.12 présente la fenêtre de sélection des métriques. L'utilisateur, après avoir choisi le mode d'alignement interactif, il choisit ensuite les méthodes de calculs de similarité qu'il désire utiliser et définit les valeurs minimale et maximale des seuils de fiabilités des correspondances. Les méthodes sélectionnées seront combinées pour déterminer la valeur de similarité des entités. L'utilisateur a également la possibilité de choisir s'il veut utiliser une référence comme validateur en cliquant sur le bouton « Use reference » ou s'il veut être lui-même le validateur en cliquant sur le bouton « Execute ». Dans tous les cas le choix de l'un ou l'autre des deux boutons démarre le processus d'alignement.



FIGURE 4.13 – liste de correspondances soumise à la validation par l'utilisateur



### Validation d'alignements

Le système sélectionne un certain nombre de couples candidats qu'il soumet à l'utilisateur pour validation (Figure 4.13). Ces couples sont sélectionnés selon le niveau de fiabilité défini par l'utilisateur dans le paramétrage du système. Pour guider l'utilisateur dans ces décisions de validation, le système fournit certains détails sur chaque entité du couple, ces détails sont entre autres les synonymes, les URIs, les concepts parents (superconcepts), les concepts fils (subconcepts) qui sont des informations pouvant éclairer l'utilisateur sur les relations entre les entités (Figure 4.14).

The screenshot shows the ServoMap interface. On the left is a menu with options: Load ontology, Indexing ontologies, Search, Wordnet, Reference, and Mapping. The main area displays a table of candidate pairs. The selected pair is 'Segment' (Entity 1) and 'Segmentation' (Entity 2) with a score of 0.46. Below this, details are provided for both entities, including their URIs, synonyms, super concepts, and sub concepts.

Entity 1	Entity 2	Score
Spiral_Organ_of_Corti	Spiral_organ_of_cochlea	0.5
Granular_component_of_nucleolus	Nucleolonema	0.38
Sebum	Sebum_MMHCC	0.35
<b>Segment</b>	<b>Segmentation</b>	<b>0.46</b>
Celiac_trunk	Celiac_Artery	0.35

Details for the selected pair (Segment / Segmentation):

- uri entity 1: <http://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#Segment>
- synonymes entity 1: {section, segment}
- super concepts entity 1: []
- sub concepts entity 1: []
- uri entity 2: [http://bioontology.org/projects/ontologies/AmefmsOwlDIComponent\\_2.owl#Segmentar](http://bioontology.org/projects/ontologies/AmefmsOwlDIComponent_2.owl#Segmentar)
- synonymes entity 2: {cleavage, segmentation, division, partition, partitioning}
- super concepts entity 2: []
- sub concepts entity 2: []

FIGURE 4.14 – détails sur un couple candidat sélectionné

The screenshot shows the ServoMap interface displaying a list of 16 selected hits. The interface includes a menu on the left and a table of selected pairs with scores. The table is titled '16 Hits selected'.

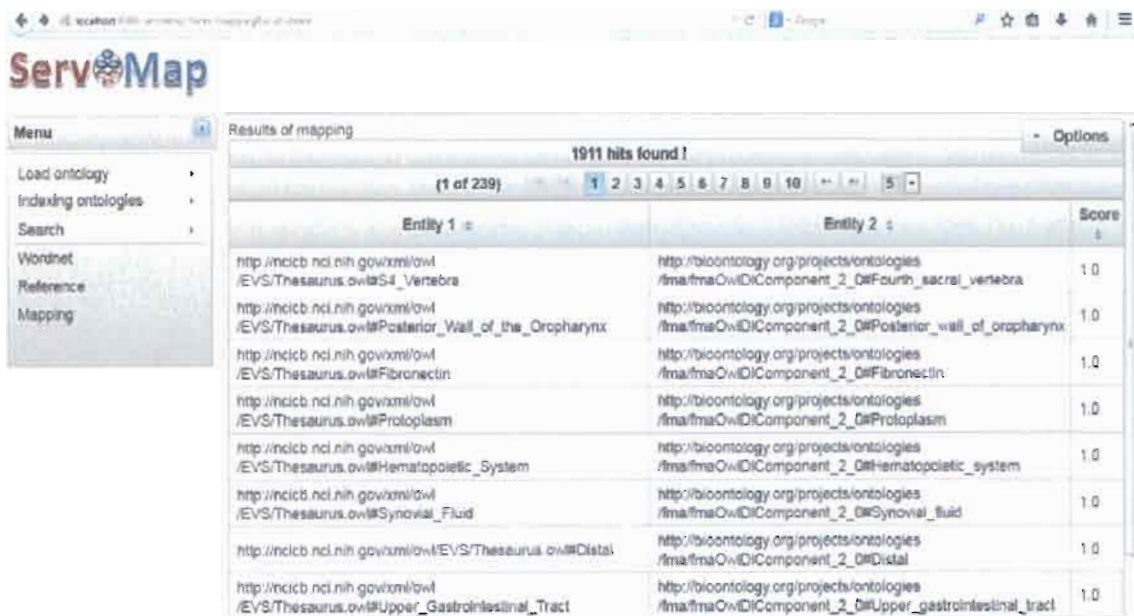
Entity 1	Entity 2	Score
Radix_Dentis	Root_of_tooth	0.16
Prepuce_of_penis	Male_Prepuce	0.3
Sebum	Sebum_MMHCC	0.35
Segment	Segmentation	0.46
Transfer_nucleic_acid	Transfer_RNA	0.4
Arch_of_aorta	Aortic_Arch	0.32
Actin_filament	F-actin	0.5
Renal_Pyramid	Set_of_renal_pyramids	0.37
Arch_of_the_Vertebræ	vertebral_arch	0.35
Hair	Set_of_hairs	0.18
Follitropin	Follitropin	0.45
Bronchus	Set_of_bronchi	0.23
Hand_Phalanx	Phalanx_of_finger	0.37
Spleen	Spleen_Part	0.41
Eyebrow	Set_of_eyebrows	0.33
Tenascin	Tenascin-C	0.45

FIGURE 4.15 – liste des couples sélectionnés comme valide par l'utilisateur

### Couples d'alignements sélectionnés comme corrects

La Figure 4.15 représente les couples sélectionnés par l'utilisateur comme

étant des correspondances correctes. Ces couples seront ajoutés à la liste des couples exacts déjà trouvés par le système à l'étape courante.



Results of mapping			Options
1911 hits found !			
(1 of 239)			
Entity 1 :	Entity 2 :	Score	
http://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#S4_Verebra	http://bioontology.org/projects/ontologies/fma/fmaOwlDiComponent_2_DiffFourth_sacral_verebra	1.0	
http://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#Posterior_Wall_of_the_Oropharynx	http://bioontology.org/projects/ontologies/fma/fmaOwlDiComponent_2_DiffPosterior_wall_of_oropharynx	1.0	
http://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#Fibronectin	http://bioontology.org/projects/ontologies/fma/fmaOwlDiComponent_2_DiffFibronectin	1.0	
http://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#Protoplasm	http://bioontology.org/projects/ontologies/fma/fmaOwlDiComponent_2_DiffProtoplasm	1.0	
http://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#Hematopoietic_System	http://bioontology.org/projects/ontologies/fma/fmaOwlDiComponent_2_DiffHematopoietic_system	1.0	
http://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#Synovial_Fluid	http://bioontology.org/projects/ontologies/fma/fmaOwlDiComponent_2_DiffSynovial_fluid	1.0	
http://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#Distal	http://bioontology.org/projects/ontologies/fma/fmaOwlDiComponent_2_DiffDistal	1.0	
http://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#Upper_Gastrointestinal_Tract	http://bioontology.org/projects/ontologies/fma/fmaOwlDiComponent_2_DiffUpper_gastrointestinal_tract	1.0	

FIGURE 4.16 – Résultat de l'alignement

### Résultat de l'alignement

La Figure 4.16 représente le résultat de l'alignement effectué sur les deux ontologies. Ce résultat est une liste de correspondance **entre les entités** des deux ontologies chargées. Une correspondance est caractérisée par les deux URIs définissant les deux entités et une valeur numérique appelée score (comprise entre 0 et 1) définissant le degré de correspondance qui relie ces deux entités. Les couples dont le score est 1 sont à cent pour cent similaires.



Evaluation			Options
Results:			
Precision :	Recall :	F-Measure :	
0.926	0.612	0.737	

FIGURE 4.17 – résultat de l'évaluation

### Résultat de l'évaluation

L'approche proposée est évaluée pour déterminer la qualité de l'alignement produit. Cette évaluation est effectuée en utilisant un alignement de référence qui est la liste de correspondances exactes des entités des ontologies alignées. Le résultat est évalué en termes de précision, de rappel et de F-mesure présentés dans la section 3.8 du chapitre 1 du mémoire. La Figure 4.17 présente le résultat de l'évaluation effectuée sur le résultat de l'alignement trouvé. La précision représente le rapport du nombre de couples exacts trouvés sur le nombre total de couples générés par le système. Ce rapport est un nombre qui varie entre 0 et 1. Plus ce rapport tend vers 1 plus le résultat de l'alignement est plus précis et fiable. Quant au rappel, il correspond au rapport du nombre de couples exacts trouvés sur le nombre total des couples possibles à trouver en occurrence (le nombre de couple contenu dans la référence). Si ce rapport se rapproche de 1, cela implique que la plupart des correspondances possibles à déterminer ont été trouvées par le système. Enfin, la F-mesure est une moyenne harmonique des deux derniers (précision et rappel).

### 2.3 Evaluation de l'approche

Pour tester l'efficacité de ce système, des ontologies exemples de diverses tailles ont été choisies. L'évaluation de l'approche a consisté à faire des tests en utilisant des ontologies de grandes tailles et des ontologies de petites tailles et à comparer les alignements obtenus à des références (alignements de référence créés manuellement). La comparaison des alignements à une référence fournira des résultats de l'évaluation en termes de précision, de rappel et de F-mesure. En outre, l'évaluation sur notre approche porte sur le nombre de correspondances trouvées et le temps de traitement pour l'obtention des résultats. Les résultats de l'évaluation sont présentés dans la Figure 4.18 en fonction des types d'ontologies (larges ou petites) et en termes de durée d'exécution, de nombre de correspondances, précision, rappel et de F-mesure.

### Conclusion

Dans ce chapitre, j'ai présenté l'architecture générale du système à travers le diagramme de paquetages, j'ai également présenté en détail dans le développement de ce chapitre la mise en œuvre du projet à travers une présentation des outils et plateformes de développement utilisés pour la réalisation de l'application et quelques captures d'écran effectuées dans la phase d'implémentation montrant le fonctionnement des services assurés par l'application.

- [Euzenat et Shvaiko, 2007] Jérôme Euzenat and Pavel Shvaiko. *Ontology matching*. Springer-Verlag, 2007. (Cité pages 4, 19, 20, 21, 24 et 64.)
- [Falconer et Noy, 2011] S. M. Falconer and N. F. Noy. *Interactive Techniques to Support Ontology Matching*. In Z. Bellahsene, A. Bonifati, and E. Rahmi, editors. *Schema Matching and Mapping*, pages 29-51. Springer, 2011.
- [Feng Shi et al. 2009] Feng Shi, Juanzi Li, Jie Tang, Guotong Xie, and Hanyu Li. *Actively learning ontology matching via user interaction*. In Abraham Bernstein, David R. Karger, Tom Heath, Lee Feigenbaum, Diana Maynard, Enrico Motta, and Krishnaprasad Thirunarayan, editors. *The Semantic Web - ISWC 2009*, volume 5823 of *Lecture Notes in Computer Science*, pages 585-600. Springer Berlin Heidelberg, 2009.
- [Furst,2004] Furst F.. *Contribution à l'ingénierie des ontologies : une méthode et un outil d'opérationnalisation*. thèse de doctorat, Université de Nantes ,2004.
- [Gruber, 1993] Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2) :199-220.
- [GUARINO, 1998] Guarino, N., « *Formal Ontology and Information Systems*», *Formal Ontology in Information Systems*. IOS Press, 1998.
- [Hameed et al., 2004] Adil Hameed, Alun Preece and Derek Sleeman. *Ontology reconciliation*. *Handbook on ontologies*. Pages 231-250, 2004.
- [Heijst et al., 1997] Van Heijst G., Schreiber A., et Wielinga B. (1997) : *Using explicit ontologies in KBS development*. *Int. J. of Human-Computer Studies*, 46(2/3) :183-292.
- [HERNANDEZ, 2005] Hernandez, N.. *Ontologies de domaine pour la modélisation du contexte en recherche d'information*. Thèse de doctorat, Université de Toulouse. 2005.
- [Hitzler, 2011] Pascal Hitzler. Kno.e.sis Center, Wright State University, Dayton, USA. "The OWL API : A Java API for OWL Ontologies" . 2011.
- [Ichise et al. 2009] H.-V. To, R. Ichise, and H.-B. Le. *An Adaptive Machine Learning Framework with User Interaction for Ontology Matching*. In *Proc. of the IJCAI2009 Workshop on Information Integration on the Web*, pages 35-40, 2009.
- [Jiménez-Ruiz et al. 2012] E. Jiménez-Ruiz, B. C. Grau, Y. Zhou, and I. Horrocks. *Large-scale interactive ontology matching : Algorithms and*

- implementation. In Proc. of the 20th European Conference on Artificial Intelligence, 2012.
- [kammoun, 2013a] Amal KAMMOUN, Conception et réalisation d'un système d'alignement d'ontologies à large échelle, p.55, 2013
- [kammoun, 2013b] Amal KAMMOUN, Conception et réalisation d'un système d'alignement d'ontologies à large échelle. p.65. 2013
- [kammoun, 2013c] Amal KAMMOUN. Conception et réalisation d'un système d'alignement d'ontologies à large échelle. p.38, 2013
- [Kefi et al.,2006] Kefi H. , Safir B., Reynaud C.," Alignement de taxonomies pour l'interrogation de sources d'information hétérogènes " INRIA, 2006.
- [Lanbrix et Kaliyaperumal, 2013] Patrick Lanbrix and Rajaram Kaliyaperumal, A Session-based Approach for Aligning Large Ontologies, 2013
- [Lanbrix et Tan. 2006] P. Lanbrix and H. Tan, "SAMBO " a system for aligning and merging biomedical ontologies. Journal of Web Semantics, vol. 4. no. 1, pp. 196-206. 2006.
- [Lassila et Swick, 1999] Lassila, O. et Swick. R. R. (1999). Resource description framework (rdf) model and syntax specification. World Wide Web Consortium W3C Recommendation 22 February 1999 / W3C
- [Leacock et al., 1998] Leacock C., Chodorow M. « Combining Local Context and WordNet Similarity for Word Sense Identification », In WordNet : An Electronic Lexical Database, MIT Press, 1998.
- [Lénart ,2007] Lénart Michèle, « SKOS. un langage de représentation de schémas de concepts », Documentaliste-Sciences de l'Information 1/ 2007 (Vol. 44). p. 75-75.
- [Levenshtein .1966] Levenshtein, I. V. (1966). Binary codes capables of corrections. deletions. insertions and reversals. Soviet Physics-Doklady. 10(8) :707-710.
- [McBride. 2004] McBride. B. (2004). 'The resource description framework (rdf) and its vocabulary description language rdfls. Handbook on Ontologies. pages 51-66.
- [Miller. 1995] George A. Miller (1995). WordNet : A Lexical Database for English. Communications of the ACM Vol. 38. No. 11 : 39-41.
- [Monge et Elkan, 1996] Monge, A. et Elkan. C. (1996). The field-matching problem : algorithm and applications.In Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, pages 267-270. London. UK. Springer-Verlag.

- [Ngo, 2012] D.H. Ngo, Z. Bellahsene. (2012), YAM+- : A Multi-strategy Based Approach for Ontology Matching Task. EKAW 2012 : 421-425
- [Noy et al., 2001] Noy N. et Musen M. Anchor. « PROMPT : Using non-local context for semantic matching », In Proceedings workshop on ontology and information sharing. IJCAI, 2001. Seattle (WA US), p. 63-70.
- [Paulheim et al, 2013] Heiko Paulheim, Sven Hertling, and Dominique Ritze ; Towards Evaluating Interactive Ontology Matching Tools
- [Ritze et Paulheim, 2011] D. Ritze and H. Paulheim. Towards an automatic parameterization of ontology matching tools based on example mappings. In Proc. of the 6th Int. Workshop on Ontology Matching. 2011.
- [Saïs ,2007] Fatiha Saïs. Intégration Sémantique de Données guidée par une Ontologie. PhD thesis, Université Paris-Sud, 2007.
- [Savadogo et al ,2014] M. Savadogo, B.M.J. Some, G. Diallo. Alignement interactif d'ontologies avec ServOMap. 5th Journées Francophones sur les Ontologies (JFO), 14-16 Nov. 2014, Hammamet, Tunisie.
- [Shvaiko et Euzenat, 2013] P. Shvaiko and J. Euzenat. Ontology Matching : State of the Art and Future Challenges. IEEE Transactions on Knowledge and Data Engineering. 25(1) :158-176. 2013.
- [Tim Berners-Lee.2001] Tim Berners-Lee, James Hendler, Ora Lassila. The Semantic Web. Scientific american. May 2001.
- [Zghal,2010] Sami Zghal :Contributions à l'alignement d'ontologies OWL par agrégation de similarités, 2010.

# ANNEXES

## ONTOLOGIES

### Utilisation et domaine d'application

Anciennement réservée aux systèmes experts simulant des raisonnements humains dans des domaines spécifiques, l'ontologie se retrouve maintenant dans une large famille de systèmes d'information. Elle est utilisée pour décrire et traiter des ressources multimédia; asseoir l'interopérabilité d'applications en réseaux; piloter des traitements automatiques de la langue naturelle; construire des solutions multilingues et interculturelles; permettre l'intégration de sources hétérogènes d'information; décrire des protocoles d'interactions complexes; vérifier la cohérence de modèles; permettre les raisonnements temporel et spatial; faire des approximations logiques; etc. Ces utilisations des ontologies se retrouvent dans de nombreux domaines d'application: intégration d'informations géographiques, gestion de ressources humaines, aide à l'analyse en biologie, commerce électronique, enseignement assisté par ordinateur, bibliothèques numériques, échanges commerciaux entre partenaires industriels, suivi médical informatisé, etc.

### Typologie des ontologies

En fonction de leur usage, on distingue classiquement cinq catégories d'ontologies [Heijst et al. 1997]: génériques, de domaine, d'application, de représentation et de méthodes.

**Ontologies de haut niveau**: décrivent des concepts très généraux comme l'espace, le temps, la matière, les objets, les événements, les actions, etc. Ces concepts ne dépendent pas d'un problème ou d'un domaine particulier, et doivent être, du moins en théorie, consensuels à de grandes communautés d'utilisateurs [GUARINO, 1998].

**Ontologies de domaine**: décrivent le vocabulaire lié à des domaines particuliers comme la physique, la mécanique, la chimie, la médecine, etc.

**Ontologies de tâche**: décrivent le vocabulaire concernant une tâche générique (ex.: enseigner, diagnostiquer...), notamment en spécialisant les concepts d'une ontologie de haut niveau [GUARINO, 1998]. Certains auteurs emploient le nom « ontologie du domaine de la tâche » pour faire référence

à ce type d'ontologie [HERNANDEZ, 2005].

**Ontologies d'application** : contiennent des concepts dépendants d'un domaine et d'une tâche particuliers, qui sont généralement subsumés par des concepts de ces deux ontologies. Ces concepts correspondent souvent aux rôles joués par les entités du domaine lors de l'exécution d'une certaine activité [GUARINO, 1998]. Il s'agit donc ici de mettre en relation les concepts d'un domaine et les concepts liés à une tâche particulière, de manière à en décrire l'exécution (ex. : apprendre les statistiques, effectuer des recherches dans le domaine de l'astronomie, etc.).

### Structure d'une ontologie

Une ontologie possède une structure bien déterminée. Elle est composée de concepts, relations, fonctions, axiomes et instances. La Figure 4.19 présente un exemple d'ontologie qui décrit une partie du domaine des êtres vivants : Les nœuds de ce graphe représentent les concepts et les arcs représentent les

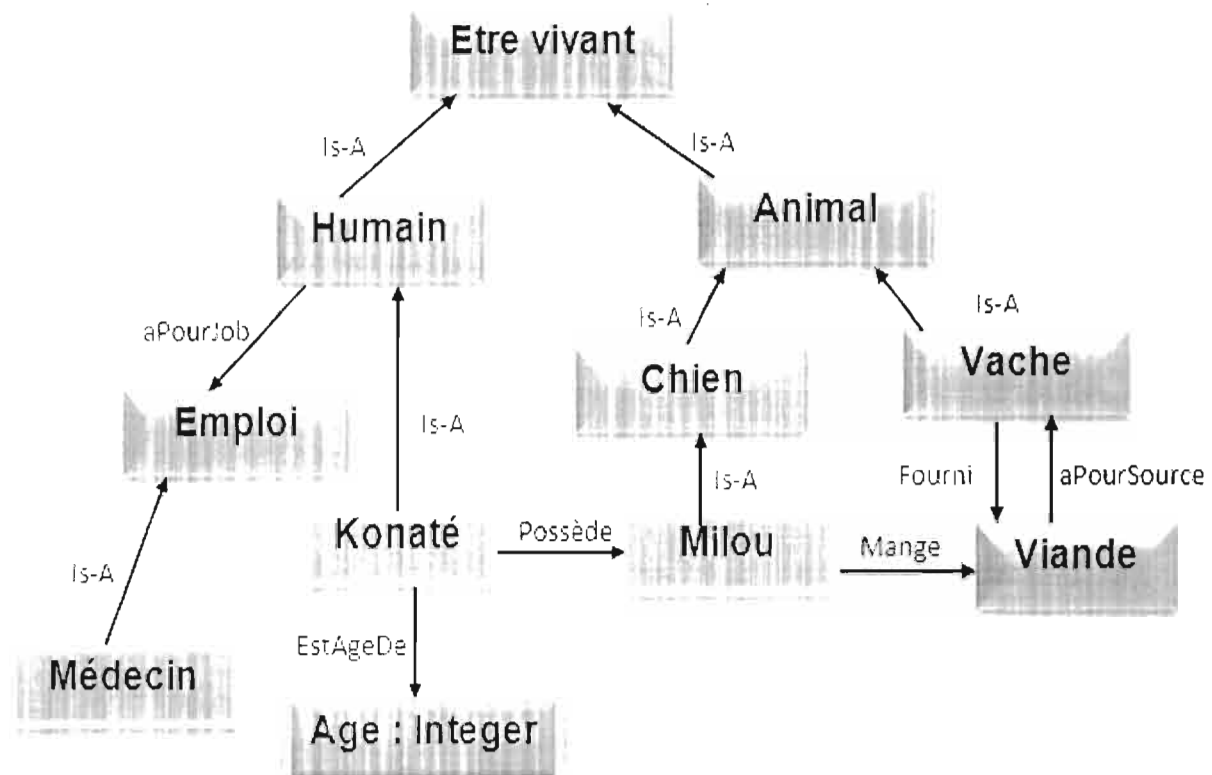


FIGURE 4.19 – une partie de l'ontologie des êtres vivants

relations. On remarque 2 types de relations :

- Relation qui relie 2 concepts : Par exemple chaque humain a pour job un emploi.
- Relation qui relie un concept à un type primitif : Par exemple, chaque humain possède un âge représenté sous forme d'un entier.

On remarque également qu'une relation est représentée par une flèche qui définit la signature de la relation :



- Le domaine (d'où la relation part)
- Le range (où la relation arrive)

**Les concepts/Classes :** Un concept représente un ensemble d'objets (matériel, une notion, une idée, etc.) et leurs propriétés communes. Il est décrit par un terme (ou ensemble de termes). Exemples : être vivant, viande. Les concepts constituent les objets de base manipulés par les ontologies. Ils correspondent aux abstractions pertinentes du domaine du problème, retenues en fonction des objectifs qu'on se donne et de l'application envisagée pour l'ontologie. Les concepts doivent être compris dans un sens très large, ils peuvent être classifiés selon plusieurs dimensions :

- Niveau d'abstraction : les concepts représentent les objets abstraits ou concrets du monde réel.
- Atomicité : les concepts représentent les objets élémentaires ou composites du monde réel.
- Niveau de réalité : les concepts représentent les objets réels ou fictifs du monde réel.

Un concept pourrait être aussi la description d'une tâche, d'une fonction, d'une action, d'une stratégie, d'un processus de raisonnement, etc.

**Les propriétés :** On différencie en général :

- Les attributs : propriétés simples qui relient un concept à un type primitif  
Exemples : Est-âgé-de
- Les relations : les relations traduisent les associations (pertinentes) existant entre les concepts du domaine. Ces relations incluent les associations suivantes :
  - Sous-classe-de (IS-A) : les classes sont organisées par des relations taxonomiques selon un ordre de généralisation ou de spécialisation.
  - Partie de (Part-OF) : une classe peut définir une partie d'une autre classe par une relation d'agrégation ou de composition.
  - Instance-de : on peut relier une instance avec sa classe d'origine.
  - Les rôles, relations sémantiques : association entre concepts. Exemples : est-source, possède.

**Les fonctions :** les fonctions sont un cas spécial de relations dans lesquelles le nième élément de la relation est unique pour les n-1 éléments précédents.

**Les restrictions :** chose qui doit être vraie pour que ce qui est exprimé soit valable. Exemple : l'exemple suivant définit la classe anonyme «Enfants de médecins » des individus dont au moins un parent est un médecin. «Enfants de médecins » est équivalente à « aPourParent » someValuesFrom « médecin

». La relation `someValuesFrom` est une restriction de propriété utilisée pour exprimer une restriction existentielle.

**Les règles :** affirmation sous la forme antécédent > conséquent décrivant des inférences possibles. Exemples : Tous les « chiens » sont des « êtres vivants »

**Les axiomes :** assertion générale sur les fondements de l'ontologie. Exemples :

- « Is-A » est transitive
- « Personne » est un sous concept de « être vivant »
- Le « range » de la propriété «est-âgé-de » est de type « integer »

**Les instances :** les instances représentent les éléments extensionnels spécifiques au domaine du problème modélisé. Exemples : « Carl » est une instance du concept « Personne »

### Langages de description des ontologies

Le langage de spécification est l'élément central sur lequel repose l'ontologie. La Figure 4.20 présente des langages de spécification d'ontologie, qui ont été récemment développés. Cette figure représente les rapports principaux entre tous ces langages sous la forme d'une pyramide des langages du WS. La plupart de ces langages se basent ou sont proches de la logique du

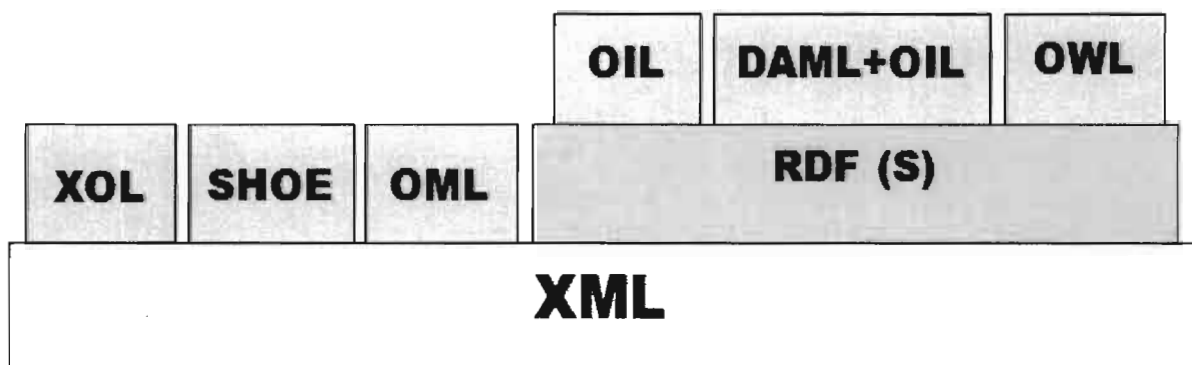


FIGURE 4.20 – pyramide des langages du web sémantique

premier ordre, et représentent donc les connaissances sous forme d'assertion (sujet, prédicat, objet). Ces langages sont typiquement conçus pour s'abstraire des structures de données et se concentrer sur la sémantique. Parmi les formalismes les plus employés se basant sur la logique des prédicats, on retrouve des langages comme N3 ou N-Triple. On peut aussi évoquer le langage DEF-\*. Par ailleurs, dans le cadre de ses travaux sur le Web sémantique, le W3C a mis en place en 2002 un groupe de travail dédié au développement de langages standards pour modéliser des ontologies utilisables et échangeables sur le Web. S'inspirant de langages précédents comme DAML+OIL et des fondements théoriques des logiques de description, ce groupe a publié en 2004 une recommandation définissant le langage OWL (Web Ontology Language),

fondé sur le standard RDF (langage de description de ressources) et en spécifiant une syntaxe XML. Plus expressif que son prédécesseur RDFS, OWL a rapidement pris une place prépondérante dans le paysage des ontologies et est désormais, de facto, le standard le plus utilisé. Bien que développé pour la représentation des vocabulaires contrôlés et structurés (thésaurus), SKOS peut être utilisé pour élaborer et gérer des ontologies légères multilingues<sup>2</sup>.

### Le langage RDF et RDF Schéma

RDF (Resource Description Framework) [Lassila et Swick, 1999] est une recommandation du W3C pour décrire des ressources. RDF est créé pour représenter l'information de la manière la moins contraignante, la plus flexible possible. La raison d'être de RDF est de permettre que les informations sur les ressources soient manipulées par des applications, plutôt que d'être simplement affichées aux utilisateurs du web. C'est entre autres pour cette raison qu'une syntaxe XML a été proposée pour véhiculer des informations modélisées en RDF. RDF décrit les ressources en exprimant des propriétés et en leur attribuant des valeurs. Il utilise pour cela le vocabulaire défini par RDF Schema noté RDF(S) [Lassila et Swick, 1999]. L'élément de base d'un document RDF est la ressource, correspondant à la représentation conceptuelle d'une entité. Une ressource est identifiée par son URI (Uniform Resource Identifier), de la forme `http://uri/du/document#courant`. La structure fondamentale de toute expression en RDF est une collection de triplets, chacun composé d'un sujet, un prédicat et un objet. Un ensemble de tels triplets est appelé un graphe RDF. Ceci peut être illustré par un diagramme composé de nœuds et d'arcs dirigés (Figure 4.21), dans lequel chaque triplet est représenté par un lien nœud-arc-nœud (d'où le terme de "graphe"). Dans un

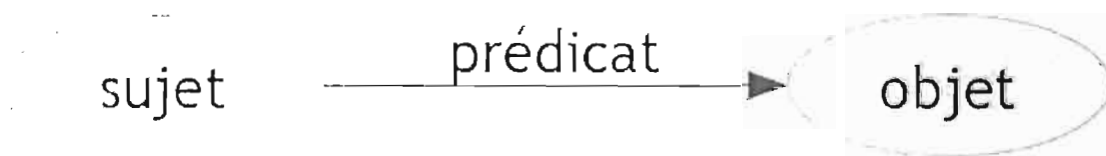


FIGURE 4.21 – schéma du triplet RDF

graphe, chaque triplet représente l'existence d'une relation entre les choses symbolisées par les nœuds qui sont joints. Le sens de l'arc a une signification : l'arc pointe toujours vers l'objet. L'écriture d'un triplet RDF indique qu'une relation, caractérisée par le prédicat, existe entre les objets représentés par le sujet et l'objet du triplet. L'assertion d'un graphe RDF est équivalente à l'assertion de tous ses triplets, et par conséquent la signification d'un graphe RDF est la conjonction (ET logique) des faits correspondant à tous ses triplets. RDF(S) fournit un ensemble de primitives simples, mais puis-

<sup>2</sup> [http://20-fr.wikipedia.org/wiki/Ontologie\(informatique\)#Langages\\_pour\\_les\\_ontologies](http://20-fr.wikipedia.org/wiki/Ontologie(informatique)#Langages_pour_les_ontologies)

santes, pour la structuration de la connaissance d'un domaine en classes et sous-classes, propriétés et sous-propriétés avec la possibilité de restreindre leur domaine d'origine (`rdf:domain`) et leur domaine d'arrivée (`rdf:range`). RDF(S) est indiqué pour la description de ressources, cependant il présente assez rapidement des limites lorsqu'il s'agit de l'utilisation comme langage de représentation d'ontologies ayant de fortes contraintes [McBride, 2004] :

- ★ la portée locale des propriétés : `rdf:range` permet de définir les domaines de valeurs (`range`) des propriétés. Cependant, il n'est pas possible avec RDF(S) de limiter leur application à un certain nombre de classes seulement. Par exemple, pour la propriété `mange` de la classe `Etre Vivant`, on peut lui assigner comme domaine de valeur `nourriture`, mais il est par la suite impossible de déclarer que certains êtres vivants mangent des plantes et que d'autres mangent de la viande ;
- ★ l'expression de la disjonction de classes : il arrive souvent que l'on veuille déclarer que deux classes sont disjointes (l'intersection de leurs instances est vide). Par exemple, les classes `Homme` et `Femme` sont disjointes. Dans RDF(S), il n'est pas possible de l'exprimer ; on ne peut définir que des relations de type "sous-classes" : (exemple : `Femme` est sous-classe de `Humain`) ;
- ★ l'expression de la combinaison booléenne de classes : on aimerait parfois construire des classes à partir de la combinaison d'autres classes en utilisant les opérateurs d'Union, d'Intersection et de Complément. Par exemple on peut vouloir définir la classe `Humain` comme l'union disjonctive des classes `Femme` et `Homme`. RDF(S) ne permet pas de le faire ;
- ★ l'expression de la restriction de cardinalités : parfois on voudrait définir le nombre exact de valeurs qu'une propriété donnée peut avoir. Par exemple, on peut vouloir dire qu'une personne a exactement deux parents ou qu'un cours est enseigné par au moins un professeur. Ce genre de restriction est impossible à définir dans RDF(S) ;
- ★ l'expression de certaines caractéristiques de propriétés : on aimerait exprimer les caractéristiques de certaines propriétés telle que la transitivité (exemple : un grand-parent est le parent d'un parent), l'inverse (`Est-Parent` est l'inverse de `Est-Enfant`), etc. De telles caractéristiques sont impossibles à exprimer avec RDF(S).

## Le langage OWL

Développé par un groupe de travail mis en place par le W3C en Novembre 2001 pour étudier la création d'un langage standard de manipulation d'ontologies web, OWL est, tout comme RDF, un langage XML profitant de

l'universalité syntaxique de XML. Il est Fondé sur la syntaxe de RDF/XML et offre un moyen d'écrire des ontologies web. Il devient une recommandation du W3C le 10 Février 2004. Le langage OWL permet d'étendre les technologies de base (XML, RDF, RDFS) pour apporter :

- d'interopérabilité ;
- Plus de raisonnements (logique de description) ;
- Plus d'évolution (intégration d'ontologies).

Le langage OWL apporte à l'utilisateur la capacité de décrire des classes et des propriétés. Il intègre, en plus, des outils de comparaison des propriétés et des classes : identité, équivalence, contraire, cardinalité, symétrie, transitivité, disjonction. Ainsi, OWL offre aux machines une plus grande capacité d'interprétation du contenu web que RDF et RDFS, grâce à un vocabulaire plus large et à une vraie sémantique formelle. Le langage OWL offre trois sous langages d'expression offrant des capacités d'expression croissantes et destinés à des communautés différentes d'utilisateurs :

- OWL Lite est le sous langage d'OWL le plus simple. Il est destiné aux utilisateurs qui ont besoin d'une hiérarchie de concepts simples. Par exemple, quoiqu'OWL Lite gère des contraintes de cardinalité, il ne permet que des valeurs de cardinalité de 0 ou 1
- OWL DL est plus complexe qu'OWL Lite, permettant une expressivité bien plus importante. OWL DL est fondé sur la logique descriptive (d'où son nom, OWL Description Logics). OWL DL comprend toutes les structures de langage d'OWL avec des restrictions comme la séparation des types (une classe ne peut pas être en même temps un individu ou une propriété, une propriété doit être un individu ou une classe).
- OWL Full est la version la plus complexe d'OWL, mais également celle qui permet le plus haut niveau d'expressivité et une liberté syntaxique de RDF sans garantir le calcul (raisonnement). Le langage OWL Full permet à une ontologie d'augmenter la signification du vocabulaire prédéfini (RDF ou OWL). Un système de raisonnement ne pourra probablement pas mettre en œuvre toutes les caractéristiques d'OWL Full.

Il existe entre ces trois sous langage une dépendance de nature hiérarchique : Toute ontologie OWL Lite valide est également une ontologie OWL DL valide, et toute ontologie OWL DL valide est également une ontologie OWL Full valide. OWL 2 est une évolution de l'OWL avec une très grande similarité au niveau des structures utilisées. OWL 2 a ajouté de nouvelles fonctionnalités en respectant OWL. Structure d'une ontologie OWL 2 : La Figure 4.22 montre la structure d'une ontologie OWL 2. Une ontologie OWL 2 est un

ensemble d'axiomes. Une ontologie ne peut pas contenir deux axiomes équivalents. Elle peut contenir aussi des annotations comme elle peut importer d'autres ontologies. Pour définir la signature d'une ontologie OWL utilise un ensemble d'entités qui définissent le vocabulaire (les termes) d'une ontologie.

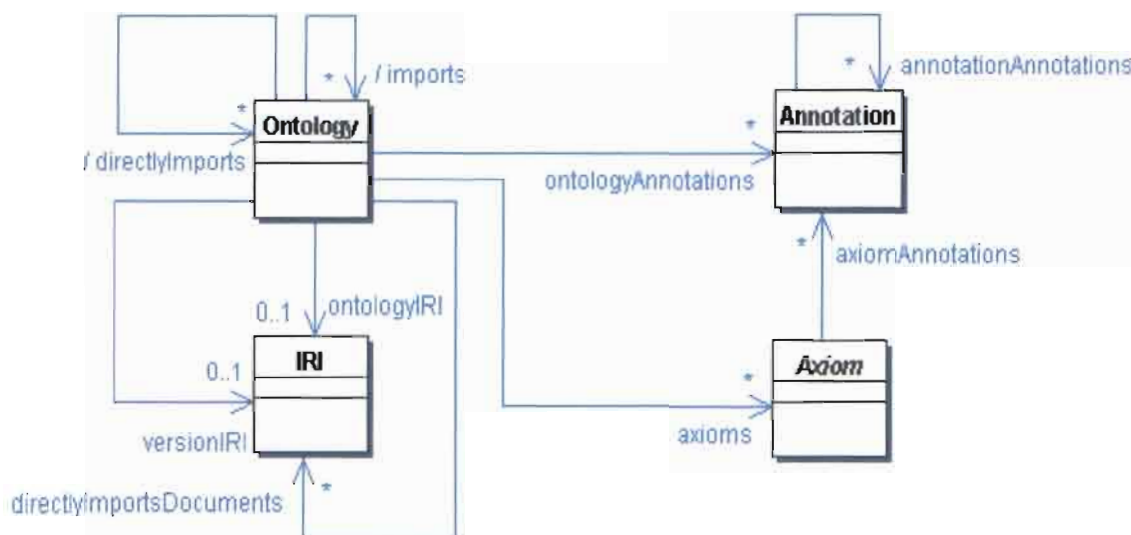


FIGURE 4.22 – la structure des ontologies OWL 2

## Le langage SKOS

**SKOS (Simple Knowledge Organisation System ou Système simple d'organisation des connaissances)** est un langage de représentation de schémas de concepts, ce qui recouvre les langages documentaires tels que les thésaurus, classifications, taxonomies, etc. Son nom a été choisi pour mettre en évidence l'objectif même visé par ce langage : proposer un système permettant d'exprimer et de gérer des modèles interprétables par les machines dans la perspective du web sémantique. Ce modèle est défini comme « simple » par opposition à d'autres modèles, comme OWL (Ontologic Web Language), plus à même de représenter des structures sémantiques plus riches telles que les ontologies, mais de ce fait également plus complexes à utiliser. Le projet a d'abord été initié par l'Union européenne dans le cadre du projet SWAD-Europe (Semantic Web Advance Development for Europe). Ces travaux ont abouti en 2003 aux premières publications de SKOS Core Guide et SKOS Mapping Guide. La réflexion a ensuite été reprise par le World Wide Web Consortium (W3C) dans le cadre du groupe de travail sur les bonnes pratiques et le déploiement des standards RDF (Resource Description Framework) et le langage SKOS est devenu une recommandation du W3C publiée le 18 août 2009 pour représenter des thésaurus, classifications ou d'autres types de vocabulaires contrôlés ou de langages documentaires... Le formalisme de représentation utilisé par SKOS repose sur les graphes

RDF. Le concept constitue le centre du graphe auquel peuvent notamment être attachés en tant que propriétés RDF [Lénart ,2007] :

- les indications portant sur le concept lui-même :
  - des termes préférentiels ou alternatifs, les équivalents dans d'autres langues,
  - les termes cachés, très pratiques pour gérer des variantes correspondant à des fautes d'orthographe courantes, ce qui permettra de les prendre en compte en recherche sans qu'elles apparaissent en affichage ou en impression du thésaurus,
  - la représentation par une image :
- les différents types de notes : notes de définition et d'application (scope note), exemples, notes historiques, etc. :
- les relations sémantiques : hiérarchie et association.

L'élément essentiel est le « SKOS Core », ou le noyau de SKOS. Ce terme de noyau est à prendre au sens propre car il s'agit bien des classes et des propriétés de base. Elles peuvent être complétées par les « SKOS Extensions », les extensions de SKOS, qui permettent :

- de représenter les relations de manière plus fine : il est possible, par exemple, de préciser si la nature d'une relation de hiérarchie est de type tout/partie ou classe instance :
- de préciser certains attributs d'un concept : une note historique, par exemple.

### **Le langage du projet OBO**

Le langage du projet OBO : Le projet OBO (Open Biomedical Ontologies) est une initiative d'un groupe de développeurs d'ontologies dans le domaine biomédical, qui s'est mis d'accord sur un nombre de principes spécifiant les bonnes pratiques pour le développement d'ontologies biomédicales. Les principes édictés reposent sur l'objectif d'interopérabilité entre les différentes ontologies développées. Un langage formel commun est fourni pour la représentation des ontologies. Il est conçu pour permettre la prise en compte de plusieurs métadonnées, et comprend un mécanisme d'historisation. Parmi les principes édictés par le projet OBO citons :

- chaque ontologie doit avoir un identifiant unique au sein de OBO ;
- chaque ontologie doit inclure des définitions textuelles pour chacun de ses termes ;
- les ontologies au sein de OBO doivent être développées de façon collaborative.

En mode « fichiers composés », l'index se trouve plutôt dans des fichiers avec des extensions .cfs et .cfx.

### WordNet

WordNet<sup>5</sup> est une base de données lexicale développée par des linguistes du laboratoire des sciences cognitives de l'université de Princeton depuis une vingtaine d'année [Miller, 1995]. Son but est de répertorier, classifier et mettre en relation de diverses manières le contenu sémantique et lexical de la langue anglaise. Des versions de WordNet pour d'autres langues existent, mais la version anglaise est cependant la plus complète à ce jour. Elle est distribuée sous une licence libre. WordNet regroupe des termes (noms, verbes, adjectifs et adverbes) en ensembles de synonymes appelés synsets. Un synset regroupe tous les termes dénotant un concept donné. Cette base de données est utilisée grâce à l'API JWNL (Java WorNet Library).

### Maven

Maven<sup>6</sup> est un outil de gestion de projet qui comprend un modèle objet pour définir un projet, un ensemble de standards, un cycle de vie, et un système de gestion des dépendances<sup>7</sup>. Il embarque aussi la logique nécessaire à l'exécution d'actions pour des phases bien définies de ce cycle de vie, par le biais de plugins. Lorsque vous utilisez Maven, vous décrivez votre projet selon un modèle objet de projet (Project Object Model : POM) clair qui contient une description détaillée de votre projet, avec en particulier des informations concernant la version et la gestion des configurations, les dépendances, les ressources de l'application, les tests, les membres de l'équipe, la structure, etc. Les cycles de vie des projets sont un concept central de Maven. Voici quelques-unes des phases les plus utiles du cycle de vie Maven :

- generate-sources : Génère le code source supplémentaire nécessité par l'application, ce qui est généralement accompli par les plug-ins appropriés.
- compile : Compile le code source du projet
- test-compile : Compile les tests unitaires du projet
- test : Exécute les tests unitaires (typiquement avec Junit)
- package : Mets en forme le code compilé dans son format de diffusion (JAR, WAR, etc.)
- integration-test : Réalise et déploie le package si nécessaire dans un environnement dans lequel les tests d'intégration peuvent être effectués.

---

5. <http://wordnet.princeton.edu/wordnet/download/>; <http://wordnet.princeton.edu/wordnet/license/>

6. <http://maven.apache.org>

7. <http://dcabasson.developpez.com/articles/java/maven/introduction-maven2>



- install : Installe les produits dans l'entrepôt local, pour être utilisé comme dépendance des autres projets sur votre machine locale.
- deploy : Réalisé dans un environnement d'intégration ou de production, copie le produit final dans un entrepôt distant pour être partagé avec d'autres développeurs ou projets.