

Université Saâd DAHLAB de Blida



Faculté des sciences

Département d'Informatique

Mémoire présenté par :

M^{lles} NADJI Khadidja et MEZZI Melyara

Pour l'obtention du diplôme de Master

Domaine : Mathématique et Informatique

Filière : Informatique

Spécialité : Ingénierie des logiciels

Sujet :

**Conception et réalisation d'un système de construction
d'Ontologies par fusion des Ontologies existantes**

Soutenu le : 02 Juillet 2011, devant le jury composé de :

M^{me} FAREH

Promotrice

M^{lle} BOUSTIA

Présidente

M^{lle} AMEUR

Examinatrice

M^{lle} BOYERBOU

Examinatrice





Dédicaces

C'est avec un immense plaisir que je dédie ce travail

A mes très chers parents qui sont toute ma vie et tout ce que j'ai de plus cher au monde, en témoignage de ma reconnaissance infinie pour les nombreux sacrifices qu'ils n'ont cessé de déployer pour moi et dont je serais à jamais redevable.

Qu'ils trouvent en ce travail la preuve de mon éternel amour et ma reconnaissance envers eux.

Que dieu les garde et leur procure la santé et le bonheur.

Ainsi qu'à mes frères bien aimés Akram et Heytem qui sont ma fierté, en témoignage de ma grande affection pour eux.

Puisse dieu les protéger.

Aussi à toute ma famille, à mes frère et sœur de cœur Nadir et Samira, aussi à Djaouida, Lamia, Nadia, Asma, Nesrine, Amina, Meriem, Fatima, Imène et Sarah.

Sans oublier mon amie khadidja (mon acolyte dans cette épreuve) ainsi que toute sa famille.

Mlle Mezzi Melyara

Dédicaces

C'est avec un immense plaisir que je dédie ce travail

A mes très chers parents en témoignage de ma reconnaissance infinie pour les nombreux sacrifices qu'ils n'ont cessé de déployer pour moi et dont je serais à jamais redevable.

Qu'ils trouvent en ce travail la preuve de mon éternel amour et ma reconnaissance envers eux.

Que dieu les garde et leur procure la santé et le bonheur.

Ainsi qu'à mon frère Mohamed Raouf et ma sœur Asmaa en témoignage de ma grande affection pour eux.

Aussi à mes amies Ahlem, Samira, Fatima, Chaima, Soumia, Hadjer et Zola

Sans oublier mon amie Melyara (mon acolyte dans cette épreuve) ainsi que toute sa famille.

Mlle Nadji Khadidja

Remerciement

Tout d'abord, nous tenons à rendre grâce à dieu tout puissant pour nous avoir donné le courage et la détermination nécessaire pour finaliser ce travail et le mener à terme.

En second lieu et puisque une seule main ne lie pas un fagot de bois, Nous tenons à remercier chaleureusement et affectueusement tous ceux et toutes celles qui de près ou de loin ont contribué à la réalisation de notre projet.

On ne saurait ne pas remercier encore une fois nos parents respectifs qui, par leur amour et leur affection nous ont permis d'arriver là où nous sommes aujourd'hui.

Nous tenons également à remercier M^{me} Fareh qui a endossé son rôle de promotrice de la meilleure façon qui soit. Nous retiendrons son aide précieuse, ces conseils avisés, ces idées riches mais aussi sa sympathie et ses encouragements. Nous prions Allah de lui rendre grâce pour avoir fait de notre travail avec elle, un réel honneur et grand plaisir.

Par ailleurs ; nous rendons un vibrant hommage à l'ensemble du corps professoral du département d'informatique de l'université Saâd DAHLAB de Blida qui ont contribué activement et vaillamment à notre formation pendant ces cinq dernières années.

Dédicaces spéciale à tous nos camarades de la promotion 2010/2011, plus particulièrement à nos amies Asma, Dahbia, Nawel, Zineb, Rafika, Warda et Wassila. Puisse dieu les aider tous à atteindre tous leurs buts.

Aussi à l'ensemble du personnel de la bibliothèque centrale qui se sont montrés très professionnels et serviables et nous ont fournis des conditions de travaux optimales.

A tous ceux et à toutes celles dont les noms n'apparaissent pas sur cette page, qu'ils demeurent convaincus, que nous ne les avons point oublié et qu'ils soient assurés de notre profonde gratitude.

Merci.

Mlle Mezzi Melyara et Mlle Nadji Khadidja.

Sommaire

Introduction générale

1. Contexte de travail.....	6
2. Problématique et motivations.....	7
3. Objectifs de travail	8
4. Organisation du mémoire.....	8

Partie I: Etat de l'art

Chapitre I: Généralités sur les Ontologies

1. Introduction	10
2. Origine des ontologies.....	10
3. Qu'est-ce qu'une ontologie ?.....	10
3.1. Définitions	10
3.2. Termes relatifs.....	12
4. Composants d'une ontologie	12
4.1. Les concepts	12
4.2. Les relations.....	14
4.3. Les fonctions	14
4.4. Les axiomes.....	15
4.5. Les instances (individus).....	15
5. Classification d'ontologies.....	15
5.1. Niveau de granularité	15
5.2. Niveau de formalisme.....	15
5.3. Objet de conceptualisation	16
5.4. Niveau de complétude.....	16
6. Buts des ontologies	17
7. Les langages d'ontologies	18
7.1. KIF.....	18
7.2. RDF/RDF Schéma.....	18
7.3. DAML + OIL	18
7.4. OWL.....	19
8. Cycle de vie global d'une ontologie	20
9. Les méthodes de construction des ontologies.....	21
10. Conclusion.....	22

Chapitre II: Fusion des Ontologies

1. Introduction	23
2. Mesures de similarité.....	23
2.1. Mesures simples	24
2.2. Mesures combinées	26
3. Fusion des ontologies	27
3.1. Définition.....	27
3.2. Principe de la fusion.....	28
3.3. Les méthodes de fusion	29
4. Analyse et synthèse	41
5. Conclusion.....	42

Partie II: Conception et Implémentation

Chapitre III: Approche proposée

1. Introduction	43
2. Etude comparative des méthodes de fusion.....	43
3. Proposition d'une nouvelle approche de fusion d'Ontologies FOnES	44
4. Conclusion.....	46

Chapitre IV: Analyse des besoins et conception

1. Introduction	47
2. Présentation de la démarche utilisée.....	47
1.1. UP (Unified process)	47
1.2. Les activités.....	48
2. Expression des besoins	49
2.1. Recueil des besoins fonctionnels (diagramme de cas d'utilisation).....	49
2.2. Recueil des besoins techniques	58
3. Analyse du système	59
3.1. Scenarios et diagrammes de séquences	59
4. Conception du système	66
4.1. Architecture du système	66
4.2. Diagramme de Paquetages	72
4.3. Diagramme de classes.....	73
5. Conclusion.....	75

Chapitre V: Implémentation

1. Introduction	76
2. Environnement de developpement	76
2.1. Langages utilises	76
2.2. OUTILS	76
3. Le systeme FOnES.....	79
3.1. Diagramme d'accessiblité du systeme.....	79
3.2. Architecture de deploiement de fones	79
3.3. Modèle de deploiement	81
3.4. Presentation de l'application	81
4. Conclusion.....	87

Chapitre VI: Tests et validation

1. Introduction	88
2. Définition des mesures de performance utilisées	88
3. Jeux de Données	90
4. mesure de performance de FOnES.....	92

Conclusion générale

Conclusion générale	96
Perspectives	98

Annexes

Annexe A : Le langage JAVA

Annexe B : Jena

Annexe C : Table des correspondances

Annexe D : Les ontologies de tests

Glossaire

Références bibliographiques et webographiques

Liste des acronymes

SI	<i>Système d'Information</i>
URI	<i>Uniform Resource Identifier</i>
URL	<i>Uniform Resource Locator</i>
W3C	<i>World Wide Web Consortium</i>
BD/BDD	<i>Base de données</i>
RDF	<i>Ressource Description Framework</i>
RDFS	<i>RDF SCHEMA</i>
XML	<i>Extended Markup Language</i>
OIL	<i>Ontology Interchange Language</i>
DAML	<i>Darpa Agent Markup Language</i>
OWL	<i>Web Ontology Language</i>
FCA-Merge	<i>Formal Conceptual Analysis Merge</i>
MOMIS	<i>Mediator Onvironment for Multiple Information Source</i>
ONIONS	<i>Ontological Integration Of Naive Sources</i>
ODL	<i>Object Description Language</i>

Les figures

Figure 1: Le cycle de vie d'une ontologie.	21
Figure 2: Mesures de calcul de similarités.	24
Figure 3 : Le principe de la fusion d'Ontologies.	28
Figure 4 : Processus progressif de la méthode MOMIS.	33
Figure 5: Processus global de la fusion par FOnES.	45
Figure 6:Processus Générique UP.	47
Figure 7:Diagramme de cas d'utilisation global du système de fusion d'Ontologies.	52
Figure 8:Diagramme de cas d'utilisation détaillé "Visualisation du résultat de fusion".	54
Figure 9:Diagramme de cas d'utilisation détaillé "Alimentation de la base d'Ontologies".	56
Figure 10:Diagramme de cas d'utilisation détaillé "Enrichissement des Ontologies".	58
Figure 11: Diagramme de séquences "Authentification" (cas n°1).	60
Figure 12: Diagramme de séquences "Authentification" (cas n°2).	60
Figure 13: Diagramme de séquences Visualisation du résultat de fusion.	62
Figure 14: Diagramme de séquences Alimentation de la base d'Ontologies (cas n°1).	63
Figure 15: Diagramme de séquences Alimentation de la base d'Ontologies (cas n°2).	63
Figure 16: Diagramme de séquences Alimentation de la bases d'Ontologies (cas n°3).	64
Figure 17: Diagramme de séquences "Enrichissement des Ontologies ".	65
Figure 18:Architecture du système.	66
Figure 19: Paquage "Structure logique de FOnES".	72
Figure 20: Paquage "Atilisateurs de FOnES".	72
Figure 21: Diagramme de paquetage global.	73
Figure 22: Diagramme de classes du système.	73
Figure 23: Diagramme d'accessibilité du système FOnES.	79
Figure 24: Architecture MVC (Modele Controle Vue).	79
Figure 25: Diagramme de déploiement de FOnES.	81
Figure 26: Interface d'authentification.	82
Figure 27: Interface de création d'un nouveau compte.	82
Figure 28: Interface principale de FOnES.	83
Figure 29: L'arbre des Concepts.	83
Figure 30: L'arbre des instances.	84
Figure 31: L'arbre des propriétés.	84
Figure 32: Menu utilisateur tiers.	85
Figure 33: Menu Administrateur.	85
Figure 34: Menu Expert.	85
Figure 35: Interface d'importation.	86
Figure 36: Editeur textuel.	86
Figure 37: Interface d'enrichissement.	87
Figure 38: Taxonomie Ontologie1.	90
Figure 39: Taxonomie Ontologie2.	91
Figure 40:Statistiques de similarité entre Ontologie1 et Ontologie2.	93
Figure 41: Degré de similarité entre Ontologie1 et Ontologie2.	93
Figure 42: Taxonomie de l'Ontologie fusionnée.	94

Les tableaux

<i>Tableau 1: Etude comparative des méthodes de fusion.</i>	43
<i>Tableau 2: Les acteurs du système.</i>	49
<i>Tableau 3: Description des besoins fonctionnels.</i>	50
<i>Tableau 4: Modèle de représentation des descriptions détaillées des cas d'utilisations.</i>	51
<i>Tableau 5: Description détaillée du diagramme de cas d'utilisation global.</i>	54
<i>Tableau 6: Description détaillée de la visualisation du résultat de fusion.</i>	56
<i>Tableau 7: Description détaillée de l'alimentation de la base d'Ontologies.</i>	57
<i>Tableau 8: Description détaillée de l'enrichissement des Ontologies.</i>	58
<i>Tableau 9: Description du diagramme de cas d'utilisation.</i>	75
<i>Tableau 10: Correspondances manuelles / FOnES (test et validation).</i>	92
<i>Tableau 11: Mesures de performances de FOnES.</i>	92

RESUME

Résumé

Nées des besoins de représentation des connaissances, les ontologies sont à l'heure actuelle au cœur des travaux menés en Ingénierie des Connaissances, visant à établir des représentations à travers lesquelles les machines puissent manipuler la sémantique des informations.

La construction des ontologies demande à la fois une étude des connaissances humaines et la définition de langages de représentation, ainsi que la réalisation de systèmes pour les manipuler. Dans ce cadre, la fusion d'ontologies s'offre comme une perspective intéressante dans la conception d'une ontologie, de par ces avantages notoires en gain de temps et ajout de clarté et d'optimalité dans la représentation des domaines.

Notre travail consistera donc, dans la conception et la réalisation d'un système permettant la construction automatique et incrémentale de nouvelles Ontologies, et ce par fusion des Ontologies existantes. Pour ce faire, l'accent sera mis sur l'introduction d'une dimension sémantique dans le processus de fusion via l'exploitation de WordNet (une base de données lexicale) ou par enrichissement sémantique d'un expert. D'autre part, cette dimension sémantique reposera sur la position structurelle des entités à comparer mais aussi sur la combinaison de mesures dites de similarité afin d'obtenir une valeur très fine de degrés de ressemblances entre ces entités.

Mots clés : Ontologie, taxonomie, fusion, sémantique, mesures de similarité (correspondances).

ABSTRACT

Abstract

Born of the needs of knowledge representation, Ontologies constitute nowadays the core of labor in knowledge engineering, to make representations through which the machines can handle the semantic information.

The building of Ontologies requires both the study of human knowledge and the definition of representation languages, together with the implementation of systems to handle them. In this context, the merging of ontologies offers itself as an interesting perspective in the design of ontology, through its significant advantages in saving time and adding clarity and optimality in domain representations.

Our work will, therefore consist in the design and implementation of a system for automatic and incremental construction of new Ontologies, this by the fusion of existing Ontologies. To do this, the focus will be on the introduction of a semantic dimension in the merging process through the exploitation of WordNet (lexical database for the English language) or an expert's semantic enrichment.

Moreover, this semantic dimension also relies on the structural location of the entities to be compared in their initial Ontologies but also the combination of values so called similarity measures to obtain a sterling degree of resemblance between these entities.

Keywords: Ontology, taxonomy, merging, semantic, similarity measures (correspondences).

ملخص

ملخص

إن الانطولوجيا عنصر أساسي في أعمال هندسة المعرفة، حيث تعمل على إقامة تمثيل من خلاله تصبح الآلات قادرة على التعامل مع دلالات المعلومات.

بناء الانطولوجيا يتطلب كل من دراسة المعرفة البشرية، لغة التمثيل و نظام المعالجة في هذا السياق يعتبر دمج الانطولوجيا طريقة جد مهمة في تصميم الانطولوجيا نظرا للايجابيات التي تقدمها كتوفير الوقت و إضافة الوضوح.

عمانا ينص على تصميم و انجاز نظام يسمح بالإنشاء الأتوماتيكي لانطولوجيا جديدة عن طريق عملية الدمج.

حيث سوف نركز على إدخال البعد الدلالي باستخدام **WordNet** أو بالإثراء الدلالي من طرف خبير.

هذا البعد الدلالي يستند أيضا على استعمال تركيبية من مقاييس التشابه للحصول على قيمة درجة التماثل بين مكونات الانطولوجيا.

الكلمات الرئيسية

انطولوجيا، تصنيف، دمج، مقاييس التشابه

Introduction générale

Introduction générale

1. Contexte de travail

Les ontologies ont pour but de saisir la connaissance dans un domaine, d'une façon générale et de fournir une représentation communément acceptée qui pourra être réutilisée et partagée par divers applications et groupes. Les ontologies fournissent un vocabulaire commun dans un domaine et définissent - à différents niveaux de formalisation - la signification de termes et leurs relations.

Les ontologies de domaine sont actuellement au cœur de nombreuses applications de l'Ingénierie des Connaissances, en particulier le Web Sémantique, car elles ont pour objectif de supporter la gestion des connaissances et le raisonnement sur ces connaissances, dans une optique d'interopérabilité sémantique entre agents humains et/ou artificiels. Les ontologies apparaissent ainsi comme des composants logiciels s'insérant dans les systèmes d'informations et leur apportant une dimension sémantique qui leur fait défaut jusqu'ici.

Mais voilà le processus de création d'Ontologie est un processus très délicat et contraignant. La construction automatique d'Ontologies a été au cœur de nombreuses recherches afin de réduire le coût qu'incombe la construction manuelle d'Ontologies.

Au fur et à mesure des expérimentations, des méthodologies de construction d'ontologies et des outils de développement adéquats sont apparus. Émergeant des pratiques artisanales initiales, une véritable ingénierie se constitue autour des ontologies, ayant pour but leur construction mais plus largement leur gestion tout au long d'un cycle de vie. De ces travaux acharnés, trois grandes catégories de construction d'Ontologies ont vu le jour, à savoir : Construction à partir de zéro, la réingénierie d'Ontologies existantes ou encore la construction de nouvelles Ontologies par fusion des Ontologies existantes. Cette dernière, s'offre comme étant l'une des solutions les moins coûteuses et qui offrent des Ontologies de qualité (car riche de par ces sources multiples).

Jusqu'à un passé très récent, le travail de fusion d'ontologies était fait à *la main* et nécessitait par conséquent une assistance constante du spécialiste du domaine de connaissance concerné. Les travaux qui s'étaient intéressés à ce type de problèmes ont abouti pour la plupart à des mécanismes de fusion basés uniquement sur la correspondance syntaxique des concepts des ontologies sources, sans tenir compte de leur sémantique ni de leur structure. Ainsi, la fusion automatique d'ontologies n'était pas encore envisageable. Depuis, les chercheurs ont constaté que plusieurs étapes du processus de fusion pouvaient être automatisées. C'est alors que quelques esquisses d'approches ont commencé à voir le jour mais les outils qui les implémentent sont, jusqu'au jour d'aujourd'hui en voie de développement.

2. Problématique et motivations

La fusion d'ontologies apparaît donc comme un processus délicat, qui suppose au minimum une compatibilité entre les formalismes de représentation et entre les modèles de connaissances utilisés. Pour le moment, la diversité prévaut dans ce domaine, comme le démontre la variété des outils de construction d'ontologies disponibles. Mais concrètement, et si on regarde le problème de plus près, on se rend compte que la construction d'Ontologie par fusion doit faire face à différents facteurs, parmi les quels, on cite :

- **Répétition** (duplication) : des ontologies de même ordre peuvent avoir été créées pour diverses applications. Cette multiplication fait souffrir les principes de réutilisation et de consistance.
- **Consistance** : chaque expert, selon son domaine de prédilection va référer un concept de manière spécifique. D'où la nécessité d'une terminologie consistante pour éviter les raisonnements inconsistants et les conclusions contradictoires.
- **Complexité** : il est en effet très complexe de construire un modèle de couverture optimale pour un domaine donné.
- **Hétérogénéité** : en effet, les ontologies ont été créées pour résoudre le problème de l'hétérogénéité des données sur le web, cependant elles sont devenues elles-mêmes source d'hétérogénéité. L'hétérogénéité se présente au niveau syntaxique, conceptuel ou pragmatique il peut même subsister entre Ontologies de même domaine ce qui rend la compréhension et l'analyse de ces dernière très difficile.

L'hétérogénéité sémantique présente un défi majeur dans le processus de fusion des ontologies. Elle est due aux différentes interprétations des objets du monde réel. En effet, les ontologies existantes sont conçues indépendamment, par des concepteurs différents, ayant des objectifs applicatifs différents. Chacune peut donc avoir un point de vue différent sur le même concept. C'est pourquoi, partant du fait que l'un des buts premiers que doit atteindre une Ontologie est *le partage de la signification* de termes dans un domaine donné, en améliorant la communication entre humains, mais aussi entre humains et ordinateurs et finalement entre ordinateurs, la difficulté principale que rencontre la fusion d'ontologies est sans contrefaçon la prise en charge de la dimension sémantique.

3. Objectifs de travail

L'objectif de ce travail est de concevoir et de réaliser un système de construction d'ontologies par fusion des ontologies existantes, en traitant le problème de l'hétérogénéité sémantique le tout d'une manière automatique pour permettre de décharger l'utilisateur de la résolution des conflits dus à cette hétérogénéité. La fusion des ontologies permet de créer une ontologie unique et cohérente; les différentes ontologies à propos du même sujet sont fusionnées en une seule qui les *unifie* toutes.

L'ontologie construite par le système final doit satisfaire les caractéristiques suivantes :

- non ambiguïté,
- établie sur des distinctions fondamentalement appropriées,
- consistante et cohérente dans l'ensemble (bien que composée de connaissances de différentes ontologies intégrées),
- possédant un niveau suffisant et adéquat de détail (pas de surcharge ou d'exagération),
- Enfin, la granularité de l'ontologie se devra d'être homogène.

4. Organisation du mémoire

Afin d'atteindre les objectifs cité ci-dessus, notre mémoire s'articulera autour de deux parties :

- Etat de l'art qui contiendra deux chapitres :
 - Chapitre I: Généralités sur les Ontologies, qui constitue une porte d'entrée et une mise en contexte global sur le monde des

Ontologies. nous y aborderons l'origine, les définitions, les composants, la classification et les buts des Ontologies.

- Chapitre II : Fusion des Ontologies, dans lequel nous présenterons les mesures de similarité, le principe de la fusion des ontologies et quelques méthodes existantes dans le domaine. Nous conclurons ce chapitre par une synthèse critique de ces dernières.

- Conception et implémentation qui contiendra quatre chapitres :

- Chapitre III: Approche proposée, qui sera consacrée à la méthode que nous avons proposé. Ce chapitre débutera par une petite étude comparative des quelques méthodes existantes pour la fusion, ce qui va nous aidé à dégager les grandes lignes (ou bien des points de départ) pour l'élaboration de notre propre méthode et il se terminera par le processus global de cette dernière.

- Chapitre IV: Analyse des besoins et conception, dans lequel, nous parlerons de la démarche de modélisation utilisée, nous présenterons la conception du système suivant cette démarche. Partant de l'analyse des besoins fonctionnels et techniques, ensuite, l'analyse du système et présentation des scénarios et enfin, la présentation de l'architecture statique.

- Chapitre V: Implémentation du système, dans lequel nous présenterons l'environnement de développement (langages et outils utilisés). Ensuite, le diagramme d'accessibilité, l'architecture de déploiement du système et enfin les impressions d'écran.

- Chapitre VI: Tests et validation, où nous évaluerons les expérimentations du système et la qualité des résultats qu'il fournit comparée aux objectifs initiaux, et ce, à travers des expérimentations et des tests sur deux Ontologies concrètes.

En fin, la conclusion de ce mémoire synthétisera nos principales contributions et donnera quelques perspectives à notre travail.

Partie I :

Etat de l'art

Chapitre I: Généralité sur les Ontologies

1. Introduction

Dans ce premier chapitre nous allons essayer de donner une introduction globale aux *Ontologies*, en mettant le point sur leur origine, leur définition, leur composition, leurs buts, leurs langages de modélisation ainsi que leur cycle de vie. Ce chapitre va donc être le premier maillon d'une longue chaîne qui va suivre et qui va se préciser au fur et à mesure pour coller à notre projet de fin d'étude qui porte sur « La construction d'une Ontologie par fusion des ontologies existantes ».

2. Origine des ontologies

Le terme *Ontologie* a été utilisé pour la première fois par les philosophes grecs dans une discipline qui a plus 2300 ans, qui s'intéresse à l'existence de l'être en tant qu'être et aux catégories fondamentales de l'existant. Etymologiquement parlant, *Onto* est un terme dérivé d'un mot grec et signifie l'Être (ainsi que ses manifestations) *Logie* vient du mot grec *Logos* qui veut dire Science ou Étude. Donc, l'Ontologie est la science qui s'intéresse à l'étude de l'être en tant qu'être.

Dans la philosophie classique, l'Ontologie correspond à ce qu'Aristote appelait la Philosophie première, c'est-à-dire *la science de l'être en tant qu'être*, par opposition aux philosophies secondes qui s'intéressent à l'étude des manifestations de l'être (les *existants*).

Ensuite la notion d'ontologie a été abordée dans le domaine de l'intelligence artificielle (IA) par John McCarthy. Il affirmait déjà en 1980 que les concepteurs des systèmes intelligents fondés sur la logique devraient d'abord énumérer tout ce qui existe. [VAL, 04]

3. Qu'est-ce qu'une ontologie ?

3.1. Définitions

Il est difficile de définir ce qu'est une ontologie d'une façon définitive. Le mot est en effet employé dans des contextes très différents touchant la philosophie, la linguistique ou l'intelligence artificielle. Nous allons nous intéresser à sa définition dans le domaine informatique.

Dans le cadre de l'intelligence artificielle, *Neeches* et ses collègues ont été les premiers à proposer une définition à savoir : «Une ontologie définit les termes et les

relations de base du vocabulaire d'un domaine ainsi que les règles qui indiquent comment combiner les termes et les relations de façon à pouvoir étendre le vocabulaire». [NEE et al, 91]

En 1993, **Gruber** propose la définition suivante : «*spécification explicite d'une conceptualisation*» [GRU, 93] qui est jusqu'à présent la définition la plus citée dans la littérature en intelligence artificielle. Cette définition a été modifiée légèrement par **Borst** comme «*spécification formelle d'une conceptualisation partagée*». [BOR, 97]

Ces deux définitions sont regroupées dans celle de **Studer** comme «*Spécification formelle et explicite d'une conceptualisation partagée*». [STU et al, 98]

- *Formelle* : l'ontologie doit être lisible par une machine, ce qui exclut le langage naturel.
- *Explicite* : la définition explicite des concepts utilisés et des contraintes de leur utilisation.
- *Conceptualisation* : le modèle abstrait d'un phénomène du monde réel par identification des concepts clefs de ce phénomène.
- *Partagée* : l'ontologie n'est pas la propriété d'un individu, mais elle représente un consensus accepté par une communauté d'utilisateurs.

Pour **Guarino & Giaretta** «*Une ontologie est une spécification rendant partiellement compte d'une conceptualisation*». [GUA et al, 95]

Swartout et ses collègues la définissent comme suit : «*une ontologie est un ensemble de termes structurés de façon hiérarchique, conçue afin de décrire un domaine et qui peut servir de charpente à une base de connaissances*». [SWA et al, 97]

La même notion est également développée par **Gomez** comme suit: «*une ontologie fournit les moyens de décrire de façon explicite la conceptualisation des connaissances représentées dans une base de connaissances*.» [GOM, 99]

Pour conclure, nous pouvons donc constater que les définitions, dans leur diversité, offrent des points de vue à la fois différents et complémentaires sur un même concept. En clair, une ontologie est un schéma particulier qui décrit, à l'aide

d'un vocabulaire structuré et un langage formel non ambigu, les concepts et les propriétés pertinents pour un domaine d'application donné.

3.2. Termes relatifs

- **Otologiste** : Personne qui exploite les principes d'ontologies, soit pour les construire ou parce que son travail est en relation avec elles.
- **Thésaurus** : Un ensemble de mots clés (descripteurs) hiérarchisés relatifs à un domaine donné et les relations entre eux (is-a, synonymes, antonymes, ... etc)
- **Les taxonomies**: Une taxonomie est la forme la plus simple (représentation graphique) des ontologies. Elle se présente par une hiérarchie des termes généralement organisés dans le sens de la spécialisation (du général au particulier) bien que d'autres organisations soient possibles mais on ne peut représenter qu'une seule relation à la fois.

4. Composants d'une ontologie

Comme nous l'avons dit, les ontologies fournissent un vocabulaire commun d'un domaine et définissent la signification des termes et des relations entre elles. La connaissance dans les ontologies est principalement formalisée à l'aide de cinq éléments: *Concepts, Relations, Fonctions, Axiomes et Instances*.

4.1. Les concepts

Un concept est un principe, une idée, une notion abstraite, sémantiquement évaluable et communicable. L'ensemble des propriétés d'un concept constitue sa compréhension ou son intension, et l'ensemble des termes qu'il englobe et son extension [NAT, 05].

Un concept est composé de trois parties :

- *Un (ou plusieurs) terme(s)* : c'est la représentation linguistique des concepts, les termes permettent de désigner le concept. Ces termes sont aussi appelés *labels* de concepts.
 - **Synonymie** : plusieurs termes dénotent le même concept
 - **Ambiguïté** : plusieurs concepts dénotés par le même terme

- *Une notion* : elle correspond à la sémantique du concept, elle est définie à travers ses propriétés et ses attributs. Elle est appelée *intention* du concept.

- *Un ensemble d'objets* : il correspond aux objets définis par le concept, il est appelé *extension* du concept. Les objets sont les *instances* du concept.

4.1.1. Les propriétés portant sur un concept [GUA, 97]

- *La généricité* : un concept est générique s'il n'admet pas d'extension

Exemple : la vérité est un concept générique.

- *L'identité* : un concept porte une propriété d'identité si cette propriété permet de conclure quant à l'identité de deux instances de ce concept. Cette propriété peut porter sur des attributs du concept ou sur d'autres concepts.

Exemple : le concept d'étudiant porte une propriété d'identité liée au numéro de l'étudiant, deux étudiants étant identiques s'ils ont le même numéro.

- *La rigidité* : un concept est dit rigide si toute instance de ce concept en reste instance dans tous les mondes possibles.

Exemple : humain est un concept rigide, étudiant est un concept non rigide.

- *L'anti-rigidité* : un concept est anti-rigide si toute instance de ce concept est essentiellement définie par son appartenance à l'extension d'un autre concept.

Autrement dit : un concept est *anti-rigide* s'il peut être une instance pour d'autres concepts.

Exemple : étudiant est un concept anti-rigide car l'étudiant est avant tout un humain.

4.1.2. Les propriétés portant sur deux concepts [GUA, 97]

- *L'équivalence* : deux concepts sont équivalents s'ils ont la même extension.

Exemple : étoile du matin et étoile du soir.

- *La disjonction* : (on parle aussi d'incompatibilité) deux concepts sont disjoints si leurs extensions sont disjointes.

Exemple: homme et femme.

- *La dépendance* : Un concept C1 est dépendant d'un concept C2 si pour toute instance de C1 il existe une instance de C2 qui ne soit ni partie ni constituant de l'instance de C1.

Exemple : *parent* est un concept dépendant de *enfant* (et vice-versa).

4.2. Les relations

Elles représentent des interactions entre concepts permettant de construire des représentations complexes de la connaissance du domaine. Elles établissent des liens sémantiques binaires, organisables hiérarchiquement [CHA et al, 04].

4.2.1. Les propriétés fondamentales à une relation [CHA et al, 04]

- *Les propriétés algébriques* : symétrie, réflexivité, transitivité
- *La cardinalité* : nombre possible de relations de ce type entre les mêmes concepts (ou instances de concept). Les relations portant une cardinalité représentent souvent des attributs.

Exemple: une pièce a au moins une porte.

4.2.2. Les propriétés liant deux relations [CHA et al, 04]

- *L'incompatibilité* : on dit que deux relations R1 et R2 sont incompatibles si elles ne peuvent lier les mêmes instances de concepts.

Exemple : « être rouge » et « être vert » sont deux relations incompatibles.

- *L'inverse* : on dit que deux relations binaires R1 et R2 sont inverses l'une de l'autre si, quand R1 lie deux instances I1 et I2, alors R2 lie I2 et I1.

Exemple : « a pour père » et « a pour enfant » sont deux relations inverses l'une de l'autre.

- *L'exclusivité* : deux relations R1 et R2 sont exclusives si, quand R1 lie des instances de concepts, R2 ne lient pas ces instances, et vice-versa. L'exclusivité entraîne l'incompatibilité.

Exemple : l'appartenance et la non appartenance sont exclusives.

4.3. Les fonctions

Elles constituent des cas particuliers de relation, dans laquelle le nième élément de la relation, est défini en fonction des n-1 éléments précédents.

4.4. Les axiomes

Les axiomes sont des expressions qui sont toujours vraies. Ils ont pour but de définir dans un langage logique la description des concepts et des relations. Leur inclusion dans une ontologie peut avoir plusieurs objectifs [STA et al, 00]:

- Définir la signification des composants.
- Définir des restrictions sur la valeur des attributs.
- Définir les arguments d'une relation.
- Vérifier la validité des informations spécifiées ou en déduire de nouvelles.

4.5. Les instances (individus)

Elles constituent la définition extensionnelle de l'ontologie ; elles sont utilisées pour représenter des éléments dans un domaine.

Exemple: les individus *Melyara* et *Khadidja* sont des instances du concept *Personne*.

5. Classification d'ontologies

Les ontologies peuvent être classifiées selon quatre dimensions:

5.1. Niveau de granularité

Les chercheurs ont divisées les ontologies par rapport au niveau de détails en deux catégories [GUA, 97] :

- **Granularité fine** : correspondant à des ontologies très détaillées, possédant un vocabulaire plus riche.
- **Granularité large** : correspondant à un vocabulaire moins détaillé.

5.2. Niveau de formalisme

On peut distinguer trois sortes d'ontologies [GRF et al, 95] :

- **Informelle** : L'ontologie est exprimée en langage naturel
- **Semi formelle**: l'ontologie est exprimée en utilisant un langage naturel structuré
- **formelle**: l'ontologie est exprimée dans un langage artificiel disposant d'une sémantique formelle.

5.3. Objet de conceptualisation

Dépendamment de leur objet de conceptualisation, les ontologies sont classifiées de la manière suivante:

- *Ontologie de représentation des connaissances:* Ce type d'ontologies regroupe les concepts (primitifs de représentation) impliqués dans la formalisation des connaissances
- *Ontologie de haut niveau:* Elle exprime des conceptualisations valables dans différents domaines. Elle décrit des concepts très généraux, Ces concepts ne dépendent pas d'un problème ou d'un domaine particulier.
- *Ontologie Générique:* Egalement appelée noyau ontologique modélise des connaissances moins abstraites que celles véhiculées par l'ontologie de haut niveau mais assez générales néanmoins pour être réutilisées à travers différents domaines.
- *Ontologie du Domaine:* Cette ontologie exprime des conceptualisations spécifiques à un domaine, elle est réutilisable pour plusieurs applications de ce domaine.
- *Ontologie de Tâches:* L'ontologie de tâches fournit un vocabulaire systématisé des termes employés pour résoudre des problèmes liés aux tâches qui peuvent être ou non du même domaine. Elle fournit un ensemble de termes au moyen desquelles nous pouvons décrire généralement comment résoudre un type de problèmes.
- *Ontologie d'Application:* Cette ontologie est la plus spécifique. Les concepts dans l'ontologie d'application correspondent souvent aux rôles joués par les entités du domaine.

5.4. Niveau de complétude

Le niveau de complétude a été abordé par *Mizoguchi* et *Bachimont*. À titre d'exemple, nous décrivons la typologie de *Bachimont*. Ce dernier propose la classification sur trois niveaux [BAC, 00]:

- **Niveau 1 - Sémantique:** Deux concepts sémantiques sont identiques si l'interprétation du terme de point de vue Communauté (similarité ou

différence avec l'ancêtre, similarité ou différence avec les frères) est la même.

- **Niveau 2 - Conceptuel:** Il est une collection structurée de termes.
- **Niveau 3 - Formel:** Il est formalisé au moyen d'un langage de représentation interprétable par une machine.

6. Buts des ontologies

Les ontologies sont développées pour [ELI, 07]:

- *Un besoin existe de partager la signification de termes dans un domaine donné :* fondamentalement, le rôle des ontologies est d'améliorer la communication entre humains, mais aussi entre humains et ordinateurs et finalement entre ordinateurs.
- *Une aide à la communication entre agents humains et aussi entre organisations :* l'existence de vocabulaires différents au sein d'une entreprise ou d'une industrie constitue un frein à la collaboration et aux partenariats.

Dans ce cas l'ontologie sert à :

- améliorer la *compréhension* entre les employés,
- favoriser la *diffusion* des informations et leur *exploitation*,
- *Une aide à la conception et à l'utilisation des systèmes d'information*
 - *Spécification ; Acquisition des connaissances :* une ontologie peut aider à l'analyse des besoins et à définir les spécifications d'un SI.
 - *Réutilisation ; Partage :* une ontologie peut être, ou peut devenir suite à une traduction, un composant réutilisable et/ou partagé par plusieurs logiciels.
 - *Fiabilité ; Maintenance :* une ontologie peut servir à améliorer la documentation d'un logiciel et/ou à automatiser des vérifications de cohérence, réduisant les coûts de maintenance.
- *Une meilleure exploitation de sources d'information*
 - *Recherche :* une ontologie peut jouer le rôle de métadonnée servant d'index dans un répertoire d'information.

d'exprimer une sémantique à travers le modèle des frames tout en utilisant la syntaxe de RDF-S.

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

Le langage OIL a été fusionné avec le langage DAML pour former le DAML+OIL.

DAML (Darpa Agent Markup Language) est conçu pour permettre l'expression d'ontologies dans une extension du langage RDF. Il offre les primitives usuelles d'une représentation à base de frames et utilise la syntaxe RDF.

L'intégration de OIL rend possible les inférences compatibles avec les logiques de description, essentiellement les calculs de liens de subsomption.

7.4. OWL

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

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

OWL vise également à rendre les ressources sur le Web aisément accessibles aux processus automatisés, d'une part en les structurant d'une façon compréhensible et standardisée, et d'autre part en leur ajoutant des méta-informations.

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

De plus, OWL tient compte de l'aspect diffus des sources de connaissances et permet à l'information d'être recueillie à partir de sources distribuées, notamment en permettant la mise en relation des ontologies et l'importation des informations provenant explicitement d'autres ontologies.

8. Cycle de vie global d'une ontologie

Les ontologies étant destinées à être utilisées comme des composants logiciels dans des systèmes répondant à des objectifs opérationnels différents, leur développement doit s'appuyer sur les mêmes principes que ceux appliqués en génie logiciel. En particulier, les ontologies doivent être considérées comme des objets techniques évolutifs et possédants un cycle de vie qui nécessite d'être spécifié [BLA et al, 98].

Les activités liées à une ontologie peuvent être regroupées en trois catégories:

- Des activités de gestion de projet: planification, contrôle, assurance qualité.
- Des activités de développement: spécification, conceptualisation, formalisation.
- Des activités de support: évaluation, documentation, gestion de la configuration.

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

- Évaluation des besoins ; construction, diffusion, et utilisation. Après chaque utilisation significative, l'ontologie et les besoins sont réétudiés et l'ontologie peut être étendue.
- La phase de construction peut être décomposée en 3 étapes: conceptualisation, ontologisation, opérationnalisation. L'étape d'ontologisation peut être complétée d'une étape d'intégration au cours de laquelle une ou plusieurs ontologies vont être importées dans l'ontologie à construire. [FRE, 02]

On peut schématiser tout ça par:

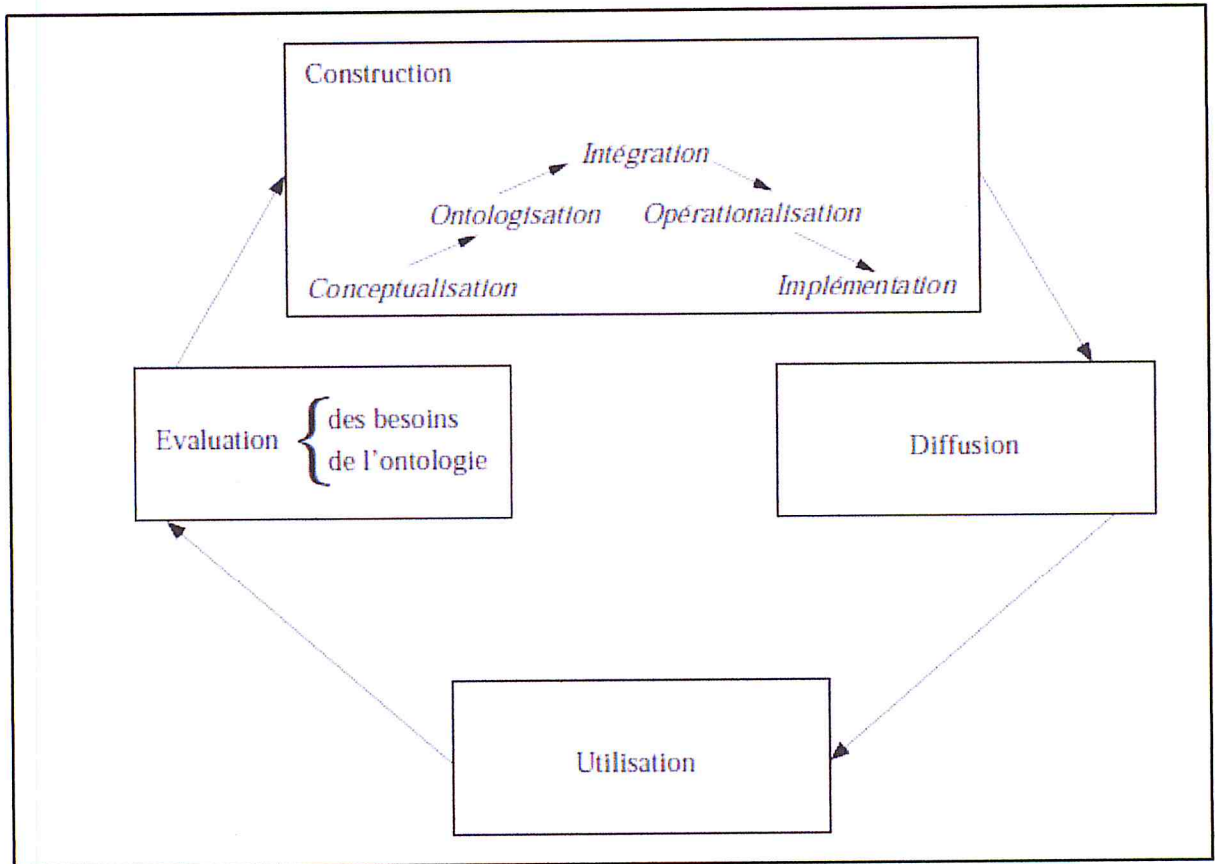


Figure 1: Le cycle de vie d'une ontologie. [FRE, 02]

9. Les méthodes de construction des ontologies

Le processus de construction d'une ontologie est un processus complexe et délicat, la difficulté se trouve au niveau de la modélisation c'est-à-dire l'extraction des termes existants et les termes pertinents pour un domaine. Les méthodologies recensées sont réparties en cinq catégories, dont les trois plus grandes sont :

- La construction d'ontologie à partir de début (création d'une nouvelle ontologie): la plupart des méthodes existantes appartiennent à cette catégorie par exemple la méthode Cyc, SENSUS ou KACTUS, la méthode d'Uschold et King ou celle de Grüninger et Fox, les méthodologies METHONTOLOGIE et On-To-Knowledge.
- L'intégration ou la fusion avec d'autres ontologies existantes: La fusion d'ontologies représente la création d'une nouvelle ontologie à partir de deux ontologies ou plus. L'ontologie résultante unifie et remplace les ontologies d'origine.

- La réingénierie d'ontologies existantes.

10. Conclusion

Dans ce premier chapitre, nous avons essayé de donner une vision générale de ce que peuvent être « *Les ontologies* », les chapitres suivants promettent d'être plus techniques et plus spécifiques quant aux différentes méthodes de construction d'ontologies et plus particulièrement à celles basées sur la fusion d'ontologies existantes.

Chapitre II: Fusion des Ontologies

1. Introduction

Dans cette section du mémoire, on introduira des définitions génériques mais non moins pertinentes sur les mesures de similarité qui nous permettent de calculer le degré de similarité entre deux chaînes de caractères (dans notre cas, deux concepts issus de deux Ontologies différentes). Ensuite, nous aborderons la notion de fusion des Ontologie en mettant le point sur : ses principes ainsi qu'une analyse et synthèse critique des méthodes et outils qui existent dans ce domaine.

2. Mesures de similarité

L'identification de la similarité dans les ontologies est un concept fondamental qui est adopté par plusieurs techniques telles que le regroupement, la fouille de données (datamining), le Web sémantique et en particulier, le domaine de la recherche de l'information qui repose largement sur des mesures pour l'identification de la similarité entre les documents. La majorité des approches de la recherche de l'information ne prennent en compte que des mots simples et/ou des fragments des mots pour la recherche des documents pertinents et ignorent l'idée essentielle qui prend en compte les rapports ontologiques des mots.

Les rapports ontologiques entre les mots peuvent être détectés par un processus de calcul de similarité entre des paires d'objets contenus dans l'ontologie [ELB, 09].

Les différentes méthodes de calcul de similarité utilisées dans le domaine Ontologique et qui sont organisées selon la classification ci-dessous :

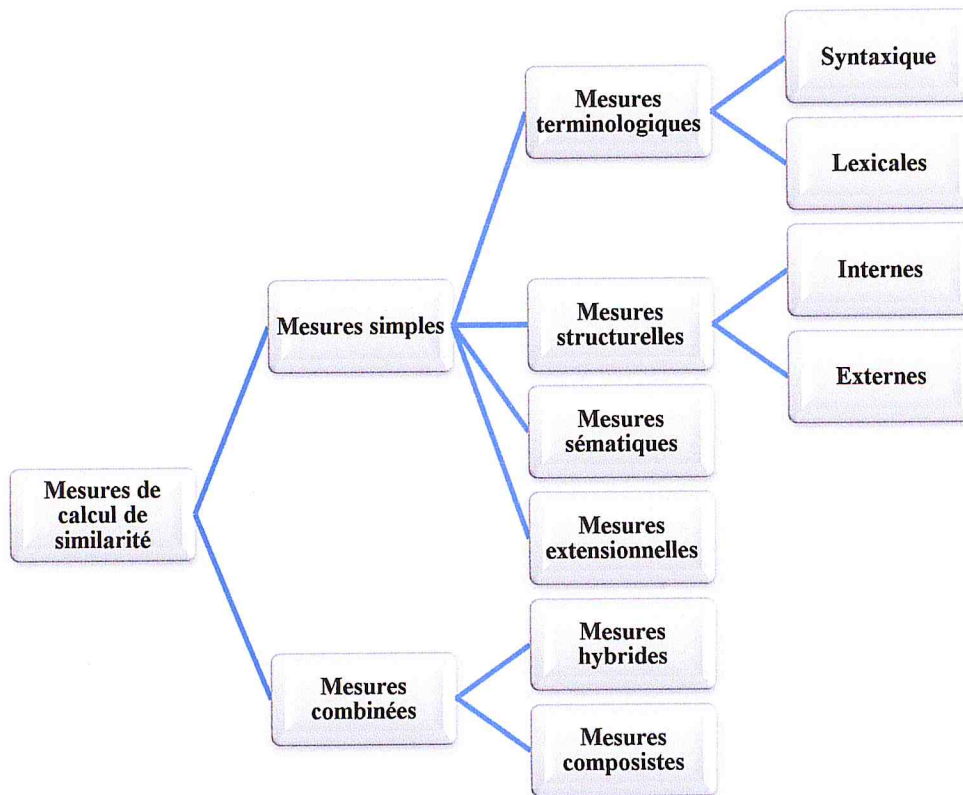


Figure 2: Mesures de calcul de similarités.

2.1. Mesures simples

2.1.1. Mesures terminologiques

Le calcul de la similarité terminologique est effectué entre les descripteurs d'entités comme les noms, les commentaires, les étiquettes et le sujet des entités.

Nous trouvons dans cette méthode deux approches : l'approche syntaxique et l'approche lexicale, appelée aussi linguistique [ELB, 09]:

- L'approche syntaxique : effectue la correspondance à travers les mesures de dis-similarité des chaînes de caractères (par exemple, la distance de Hemming qui permet de calculer le nombre de positions dans lesquelles deux chaînes de caractères se différencient).
- Dans l'approche lexicale ou linguistique, les informations exploitées peuvent être celles intrinsèques (des propriétés linguistiques internes des termes telles que des propriétés morphologiques ou syntaxiques) ou celles extrinsèques en effectuant la correspondance à

travers les relations lexicales (par exemple, synonymie, hyponymie, etc.) en faisant appel à des ressources auxiliaires telles que les dictionnaires de synonymes et d'hyponymes, mais aussi à des thésaurus et des ressources sémantiques ainsi qu'à des dictionnaires spécifiques aux domaines étudiés.

2.1.2. Mesures structurelles

Ce sont des méthodes qui 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 [GHO, 08].

Nous appelons méthodes structurelles internes les méthodes qui calculent la similarité entre deux concepts en exploitant les informations relatives à leur structure interne (restrictions et cardinalités sur les attributs, valeurs des instances,...) [GHO, 08]. et méthodes structurelles externes, les méthodes qui se servent de la structure hiérarchique de l'ontologie et se basent sur des techniques de comptage d'arcs pour déterminer la similarité sémantique entre deux entités.

2.1.3. Mesures extensionnelles

Elles déduisent la similarité entre deux entités qui sont notamment des concepts ou des classes en analysant leurs extensions (leurs ensembles d'instances).

Chaque instance peut être représentée par un vecteur de noms et/ou de valeurs. Des calculs de similarités entre vecteurs permettent de comparer les instances [STA et al, 00].

On distingue Deux approches pour comparer les ontologies à partir des instances associées aux concepts d'ontologies :

- soit les deux ontologies à comparer référencent les mêmes instances et dans ce cas on génère une similarité entre les concepts qui partagent les mêmes instances ;
- soit les deux ontologies à comparer ne référencent pas les mêmes instances et dans ce cas on fait des recherches par mots-clés dans les instances (souvent des documents ou autres fichiers). La similarité est ensuite calculée entre les instances à l'aide de ces mots-clés.

2.1.4. Mesures sémantiques

L'évaluation du lien sémantique entre deux concepts dans une ontologie est un problème de longue date dans le domaine de l'intelligence artificielle et de la psychologie. La similarité sémantique est une évaluation du lien sémantique entre deux concepts dont le but est d'estimer le degré par lequel les concepts sont proches dans leur sens [MEL, 07].

La définition donnée par Lin de la similarité sémantique repose sur trois suppositions. La similarité entre deux concepts est liée aux caractéristiques qu'ils ont en commun (plus ils ont de caractéristiques communes, plus les concepts sont similaires) et à leurs différences (plus deux concepts sont différents, moins ils sont similaires). La similarité maximale est obtenue lorsque deux concepts sont identiques [MEL, 07].

La majorité des travaux portant sur le calcul de similarité dans une ontologie considèrent que la similarité peut être évaluée uniquement à partir des liens taxonomiques (ou lien « est un ») dans cette approche différentes mesures ont été définies. Ces mesures sont classées par rapport aux caractéristiques des concepts pertinents de l'ontologie et reposent soit sur la distance entre les concepts à travers leurs liens dans l'ontologie, soit sur l'information contenue par les concepts, soit sur les deux.

D'autres, au contraire, estiment que ce calcul doit intégrer les autres types de liens comme le lien « partie de » [MEL, 07].

2.2. Mesures combinées

Les différentes techniques citées auparavant peuvent ensuite être utilisées ensemble dans un algorithme « hybride » (deux ou plusieurs techniques dans un même algorithme) ou en un paramétrage d'algorithmes exécutés en parallèle (« composite ») [ELB, 09].

2.2.1. La combinaison séquentielle (hybride)

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

Par exemple, nous choisissons de lancer une mesure terminologique avant de lancer une autre mesure structurelle ou sémantique [ELB, 09].

2.2.2. La combinaison parallèle (composite)

Une autre manière de combiner les résultats des différentes mesures (c.-à-d. les valeurs de similarité) consiste tout d'abord à lancer parallèlement plusieurs mesures, puis par la suite à combiner leurs résultats [ELB, 09].

3. Fusion des ontologies

Dans cette section, nous allons aborder la fusion d'Ontologie. En commençant par, une définition, ensuite le principe de la fusion et on présentera enfin quelques méthodes de fusion d'Ontologies.

3.1. Définition

La fusion d'ontologies est le processus de création d'une seule ontologie rassemblant les connaissances de deux ou plusieurs ontologies existantes et différentes qui décrivent le même sujet ou appartiennent au même domaine d'application. C'est donc la création d'une nouvelle ontologie, appelée *l'ontologie fusionnée* capturant les connaissances des ontologies d'origines. Le défi est alors d'assurer que toutes les correspondances et les différences entre les ontologies soient correctement prises en compte dans l'ontologie résultante [ELB, 09].

L'ontologie fusionnée unifie et remplace les ontologies d'origine. Cette définition ne précise pas comment l'ontologie résultante est reliée aux ontologies originales pour laisser ouvert le problème du choix de la méthode de fusion [MOS, 05].

Les approches les plus courantes utilisent l'union ou l'intersection. Dans l'approche par union, l'ontologie résultante contient l'union des entités provenant des ontologies originales et suppose résolues les différences de représentation d'un même concept. Dans l'approche de type intersection, l'ontologie résultante ne contient que les parties communes des ontologies originales [MEL, 07].

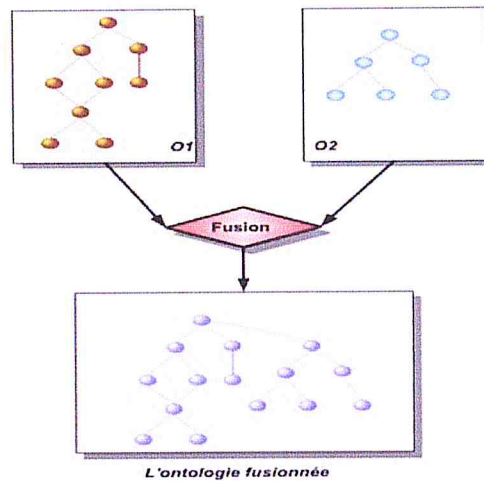


Figure 3 : Le principe de la fusion d'Ontologies. [FRE, 02]

3.2. Principe de la fusion

L'utilisation conjointe de deux (ou plus de deux) ontologies peut nécessiter soit un simple alignement dans le cas où aucune partie n'est commune aux ontologies, soit une véritable fusion. L'alignement suffit dans le cas de l'utilisation d'ontologies portant sur des domaines de connaissance complémentaires, ou sur des domaines de niveaux sémantiques différents. Par exemple, l'utilisation, dans un même système, d'une ontologie de haut niveau et d'une ontologie de domaine ne va nécessiter qu'une compatibilité entre les deux. La compatibilité de deux ontologies est assurée par l'utilisation des mêmes formalismes de représentation, ou l'utilisation de formalismes compatibles, mais également par la compatibilité des modèles de connaissance utilisés.

Préalablement à la fusion, il convient de déterminer quelle est l'ontologie la plus générale, ou celle qui est la plus étendue, c'est-à-dire celle qui ne sera pas modifiée. Les autres devront être alignées sémantiquement et syntaxiquement sur l'ontologie la plus générale. Le problème se ramène alors à l'intégration d'une ontologie dans une autre.

La fusion de deux ontologies suppose la présence dans ces deux ontologies d'entités conceptuelles (concepts ou relations) communes. Une fois les deux ontologies exprimées dans le même formalisme et à travers le même modèle cognitif, les entités communes aux deux ontologies doivent être identifiées. L'uniformisation des modèles et formalismes de représentation est également

nécessaire à la fusion d'ontologies. Les différents critères qui peuvent alors être appliqués pour repérer les similarités entre entités conceptuelles sont [MAE, 02] :

- La similarité des termes désignant deux entités ;
- La similarité des propriétés portées par deux entités ;
- La similarité des entités subsumant ou étant subsumées par deux entités.

Les correspondances ainsi établies entre entités conceptuelles ne sont pas forcément bijectives. Des conflits peuvent naître lors de cette « traduction au niveau sémantique », qui ne peuvent être résolus automatiquement. Si le degré de similarité ne permet pas de trancher entre deux correspondances possibles, l'intervention humaine est indispensable. D'autre part, si certaines entités de l'ontologie à intégrer n'offrent de similarité avec aucune entité de l'ontologie cible, il est tout de même nécessaire de leur trouver une entité subsumante dans l'ontologie cible. La différence de granularité entre les deux ontologies peut de plus entraîner la suppression de certaines entités, ou plus précisément leur agrégation au sein d'une même entité cible.

La fusion d'ontologies apparaît donc comme un processus délicat, qui suppose au minimum une compatibilité entre les formalismes de représentation et entre les modèles de connaissance utilisés. Pour le moment, la diversité prévaut dans ce domaine, comme le démontre la variété des outils de construction d'ontologies disponibles [MAE, 02].

3.3. Les méthodes de fusion

Dans cette section, nous allons essayer de définir une partie des méthodes existantes dans le spectre de la fusion des Ontologies afin d'en faire une brève analyse. En effet, il existe quelques approches (bien que peu nombreuses contrairement à l'alignement des ontologies où les méthodes sont nombreuses) et parmi les plus connues, on retient :

3.3.1. FCA-Merge

a) Définition

Stumme et Maedche ont proposé FCA-Merge (*Formal Conceptual Analysis Merge*) [STA et al, 00] dans le but de fusionner des ontologies locales qui partagent

le même ensemble d'instances. Pour cela, les auteurs exploitent l'analyse formelle des concepts.

Dans FCA-Merge, les auteurs définissent une méthode formelle et ascendante de fusion des ontologies en se basant sur un ensemble de documents.

Ils appliquent des techniques de traitement du langage naturel et d'analyse formel de concepts pour dériver le treillis des concepts. Ce dernier est exploré et transformé en une ontologie par l'intervention de l'être humain.

Le processus de fusion nécessite trois étapes, nommées :

- L'extraction des instances à partir de documents de type texte.
- Application de l'algorithme FCA pour la génération du treillis de concepts en appliquant l'analyse formelle des concepts aux instances.
- La génération interactive de l'ontologie fusionnée est l'étape finale de l'analyse du treillis qui construit l'ontologie globale.

b) Processus

Le processus de fusion des ontologies selon FCA-MERGE se présente comme suit :

- *Analyse linguistique et génération du contexte:* La méthode prend en entrée, les deux ontologies et un ensemble de documents D . Ces documents sont des documents descriptifs des ontologies initiales. Ils contiennent tous les concepts rencontrés dans ces dernières ainsi que leurs instances qui vont être extraites par la suite.

Le but de cette première étape est de générer pour chaque ontologie O_i , $i \in \{1, 2\}$, un contexte formel $K_i (G_i, M_i, I_i)$. L'ensemble des documents est considéré comme des objets ($G_i := D$) et celui des concepts comme des attributs ($M_i = C_i$). La tâche la plus difficile est la génération des relations I_i telle qu'une relation $(g, m) \in I_i$ doit exprimer le fait que le document g contient une instance de m .

Cette étape se termine, par la génération des deux contextes K_1 et K_2 dits formels parce qu'ils associent à chaque concept tous les documents aux quels il appartient.

• *Génération du treillis de concepts:* Cette étape se fonde sur la construction d'un treillis de concepts (appelé aussi treillis de Galois) qui représente le résultat de la fusion des deux ontologies. C'est là où l'Algorithme FCA (Formal Concepts Analysis) proprement dit, est appliqué. L'utilisation de treillis de Galois pour l'analyse formelle de concepts est une approche théorique de regroupement conceptuel des données permettant d'identifier des concepts. Elle a été initiée par R. Wille au début des années 80. Le treillis de Galois est constitué de couples d'ensembles fermés d'instances et d'attributs. Plus précisément, le contexte formel d'un treillis de Galois est défini par un triplet $(G; M; I)$ où G est un ensemble d'instances, M un ensemble d'attributs, et I une relation binaire telle que $I \subseteq G \times M$. Le couple $(g; m)$ signifie que l'instance g possède l'attribut m .

Soit $A \subseteq G$, l'intension de A est l'ensemble A' tel que

$$A' = \{m \in M \mid \forall g \in A; (g; m) \in I\}$$

Soit $B \subseteq M$, l'extension de B est l'ensemble B' tel que

$$B' = \{g \in G \mid \forall m \in B; (g; m) \in I\}$$

Un concept formel de $(G; M; I)$ est une paire $(A; B)$ de $G \times M$ telle que

$$A' = B \text{ et } B' = A$$

Les concepts sont organisés en hiérarchie grâce à une relation de subsomption: $(A; B1) \leq (A2; B2)$ ssi $A1 \subseteq A2$ (ou $B2 \subseteq B1$).

Dans le cadre de FCA-Merge, G est un ensemble de documents et M est l'ensemble des concepts de l'ontologie. Pour chaque ontologie O_i , une étape d'analyse linguistique des documents de G permet de générer un contexte formel $(G; M_i; I_i)$ tel que l'on a $(g; m) \in I_i$ si le document g contient des instances du concept m .

Ensuite, un treillis de concepts est construit à partir de l'union de ces deux contextes. Seuls les concepts formels équivalents ou généralisant au moins l'un des concepts des ontologies originales seront pris en considération. Si le même concept est rencontré dans les deux Ontologies, ce dernier est considéré comme traité différemment et apparaîtra dans le treillis final avec

des identifiants différents (représentant les deux Ontologies initiales) afin d'éviter tout conflit.

Cette approche semi-automatique permet donc de découvrir des appariements entre des concepts de deux ontologies, en se fondant sur leurs instances communes. Elle permet également de découvrir de nouveaux concepts ou de nouvelles relations dites règles d'association. Pour déterminer ces règles et calculer le treillis final, des algorithmes, tels que : CLOSE, A-CLOSE ou TITANIC sont utilisés.

- *Génération de la nouvelle ontologie:* La validation de l'ontologie fusionnée ne peut se faire que par l'intervention d'un expert, ce dernier doit résoudre les concepts conflictuels présents dans le treillis de Gallois. Dans les cas où les concepts formels sont constitués de plusieurs concepts, l'utilisateur doit décider si un nouveau concept doit être créé ou si le concept formel représente une relation.

c) Constat

Le principal problème de la méthode FCA-Merge est son côté semi-automatique du fait qu'elle ait besoin de l'intervention d'un humain pour la validation du processus de construction. Aussi la difficulté qu'incombe la procuration des documents initiaux (pertinent pour la génération des contextes) et l'absence de l'aspect sémantique, en font des points faibles de cette méthode.

3.3.2. MOMIS

a) Définition

MOMIS (Mediator Onvironment for Multiple Information Source) [GAY, 06][FRE, 02] est fondée sur une approche sémantique basée sur un schéma conceptuel (appelé médiateur/adaptateur) et sur l'utilisation d'une logique de description très riche des schémas sources à intégrer. Le processus d'intégration de cette méthode consiste à la création d'un schéma virtuel qui regroupe toutes les sources de données.

b) Processus

La méthode MOMIS s'axe sur trois étapes pour parvenir au schéma global :

- *Construction du thésaurus commun:* Le thésaurus commun est un ensemble de relations terminologiques intentionnelles et extensionnelles (synonymes, hyperonymes ...), décrivant les connaissances inter et intra-schémas sur les classes et attributs du schéma source. La méthode fait appel à Wordnet¹ pour extraire les spécifications lexicales qui dérivent de ces relations. La construction d'un thésaurus commun se fait en suivant un processus progressif (figure 4) qui permet d'ajouter des relations selon un ordre bien précis :

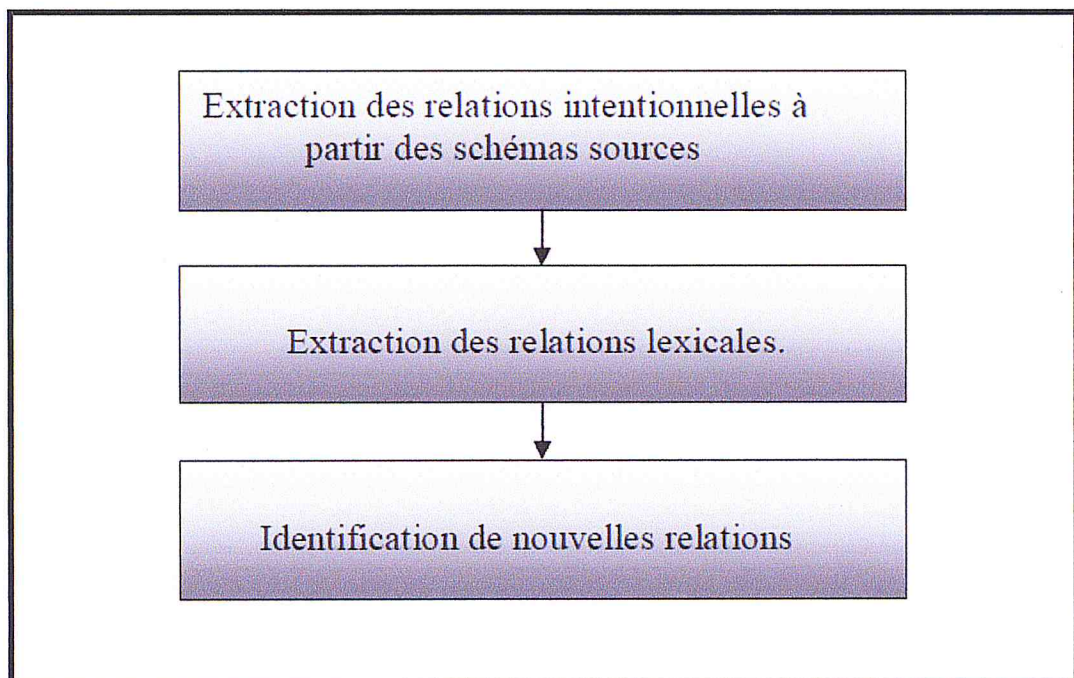


Figure 4 : Processus progressif de la méthode MOMIS. [GAY, 06]

Le thésaurus commun représente une référence pour la détermination des classes candidates lors de l'intégration. La méthode s'intéresse plus particulièrement à la détermination des relations intentionnelles et extensionnelles et ce à travers trois étapes :

1. Extraction des relations intentionnelles à partir des schémas sources : ces relations sont extraites en analysant les ODL (Object Description Language) des schémas sources séparément. Les relations

¹ Wordnet est une ressource lexicale de langue anglaise, disponible sur internet, qui regroupe des termes (noms, verbes, adjectifs et adverbes) en ensembles de synonymes appelés *synsets*

intra-schéma sont extraites à partir des spécifications de clés étrangères dans les schémas relationnels

2. Extraction des relations lexicales : les relations lexicales sont extraites grâce au module SLIM qui analyse les différents schémas ODL en correspondance avec Wordnet (pour l'analyse lexicale).

3. Relations ajoutées par l'expert : ces dernières sont ajoutées directement par l'utilisateur afin de capturer les spécifications du domaine de connaissances à propos du schéma source.

4. Identification de nouvelles relations : les nouvelles relations intentionnelles et extensionnelle qui se trouvent au sein des schémas internes sont identifiées par l'application de règles d'inférences.

Toutes ces relations sont ajoutées au thésaurus commun et vont servir durant les autres phases permettant la génération du schéma global.

Par ailleurs, l'utilisateur peut intervenir afin d'ajouter des relations d'ordre terminologique entre deux classes et donc introduire des relations d'ordre extensionnel.

- *Analyse de similarité des classes et Clustering* : Les relations dans le thésaurus commun sont utilisées pour évaluer le niveau de similarité entre les classes intra et inter sources (par classification hiérarchique). Le concept de similarité est introduit pour formaliser le type de relations qui peuvent avoir lieu entre les classes du point de vue de l'intégration. MOMIS utilise des techniques de regroupement hiérarchique basées sur le concept d'affinité (coefficient calculé à partir de chaque paire de classe en prenant en considération le nom des classes, structure des classes et les relations dans le thésaurus commun) qui permet de déterminer le niveau de la sémantique et de localiser les classes qui décrivent la même information. L'outil ARTEMIS (Analysis and Reconciliation Tool Environnement for Multiple Information Sources) permet d'évaluer le niveau d'affinité entre toutes les paires de classes possibles (inter-schéma) en se basant sur les relations contenue dans le thésaurus commun, ARTEMIS procède par analyse sémantique.

Le résultat de cette étape est un arbre de concepts qui est représentatif de toutes les classes locales. Dans ce dernier ; chaque nœud représente un niveau de regroupement, les classes adjacentes sont considérées comme ayant un haut niveau d'affinité et de même les classes qui se trouvent loin les unes des autres ont un bas niveau d'affinité.

- *Génération des classes globales:* Partant des clusters générés, il faut définir pour chaque cluster une classe globale qui représente toutes les classes du cluster. Pour chaque classe globale, le SI-Designer (Source Integrator Designer qui est un outil pour l'intégration semi-automatique de sources hétérogènes) crée un ensemble d'attributs globaux et pour chacun de ces attributs, il détermine la correspondance avec les attributs locaux (ceux qui appartiennent aux clusters des classes globales). L'outil est non seulement capable de trouver les correspondances mais également de résoudre les ambiguïtés. Les attributs globaux sont construits en deux étapes :

- Union des attributs de toutes les classes d'un cluster.
- Fusion des attributs similaires (éliminer les redondances en prenant en considération les relations contenues dans le thésaurus commun). Le processus de fusion est entièrement automatique. Il délègue dans de très rares cas conflictuels (dus à une ambiguïté des relations) à l'expert la tâche de fusion.

Ensuite, pour chaque classe globale, une table de correspondances intentionnelles est générée. Les colonnes de cette dernière représentent l'ensemble de classes locales qui appartiennent au cluster et les lignes représentent l'ensemble des attributs globaux. Un élément M [ag] [cl] représente la correspondance entre, une classe locale « cl », et un attribut global « ag ».

Avec la construction de l'ontologie fusionnée, plusieurs composants sont aussi construits, parmi les quels :

- *Ontologie locale (source):* qui est une description formelle et explicite de concepts dans une source d'information (classe), propriétés de chaque concept décrivant différentes caractéristiques des concepts (attributs) et restriction des instances de classes (contraintes d'intégrité). Une fois cette

description exprimée en langage formel, l'ontologie est appelée « Schéma local »

- Annotation des sources locales : l'ontologie fusionnée ayant différentes sources, il est nécessaire de distinguer le sens des termes dans les différentes ontologies locales (via Wordnet par exemple).
 - Thésaurus commun : un ensemble de relation terminologiques intentionnelles et extensionnelles décrivant les connaissances inter et intra-schéma (classes et attributs des schémas sources).
 - Ontologie globale : qui est la vue globale résultante de l'intégration des ontologies locales. Exprimé dans le même langage formel et avec le même niveau de granularité des données utilisées pour décrire les ontologies locales. L'ontologie globale est aussi appelée « schéma global ».

c) Constat

MOMIS a été mise en place pour pallier en premier lieu aux problèmes d'hétérogénéité des sources d'informations présentes dans le web (facilitant le partage et l'évolution de ces dernières). C'est pour ces raisons, que MOMIS s'est inscrite comme étant un médiateur pour l'extraction et l'intégration d'informations structurées et semi-structurées. Sa force réside dans le fait qu'elle se base sur une approche sémantique et aussi descriptive des données.

MOMIS est une méthode qui se base dans son processus de fusion sur un ensemble d'outils pour faciliter la création de l'ontologie fusionnée, comme :

- WordNet : pour la découverte des relations lexicales qui peuvent exister entre les différents termes extraits.
- ARTEMIS : qui est l'outil qui permet le regroupement effectif des classes.

L'un des plus grands défauts de MOMIS est son aspect statique, en effet destinées à être exploitées dans le WEB, les ontologies construites avec MOMIS doivent être en mesure de fournir une grande souplesse pour les utilisateurs d'un point de vue changement et évolution (les informations sur le web étant de sources hétérogène et qui évoluent rapidement) MOMIS doit être en mesure de suivre la cadence hors cette dernière ne le fait pas, elle ne permet ni l'ajout, ni la modification ou la suppression dans l'ontologie construite ce qui représente un grand point faible

dans un environnement aussi dynamique que le web où, une ontologie rencontre des changements de domaine de spécification et de conceptualisation à cause de :

- Les ontologies rencontrent parfois des erreurs et ne peuvent pas immédiatement répondre aux besoins des utilisateurs
- L'environnement dans lequel opère l'ontologie peut changer sans prévenir. C'est pourquoi ça peut fausser toutes les bases sur les quels l'ontologie a été bâtie
- Les besoins des utilisateurs, peuvent changer par rapport à leurs besoins avant la construction de l'ontologie

3.3.3. PROMPT

a) Définition

Avec PROMPT, les auteurs ont proposé une méthode de comparaison des ontologies de domaine par la définition des correspondances entre leurs concepts. Ils choisissent deux paires de concepts équivalents comme référence. Chaque paire appartient à une ontologie. Ensuite, ils sélectionnent tous les concepts intermédiaires deux à deux qui occupent les mêmes positions dans deux chemins de même longueur, reliant les deux concepts de la même paire. Cela permet aux auteurs de juger si ces paires de concepts sont équivalents ou pas. Selon les auteurs, deux concepts se trouvant dans la même position entre deux concepts équivalents, sont également équivalents. Prompt suppose que les deux ontologies sont construites de la même façon. Ce n'est pas le cas dans la réalité [MAI et al, 07].

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

b) Processus

PROMPT est un algorithme de fusion/alignement d'ontologies basé sur une approche semi-automatique. PROMPT effectue certaines tâches automatiquement et guide l'utilisateur pour l'accomplissement d'autres tâches pour lesquelles son intervention est nécessaire. Les caractéristiques inhérentes de PROMPT peuvent se résumer à:

- PROMPT détermine des inconsistances possibles dans l'état de l'ontologie résultant des actions de l'utilisateur et suggère des solutions à ces inconsistances. C'est cette propriété qui fait la force de PROMPT par rapport à d'autres algorithmes de fusion qui détectent des inconsistances mais ne font aucune suggestion quant à la façon de les résoudre.

- PROMPT est basé sur un modèle de connaissance extrêmement général et donc multi plateformes. Il se caractérise par les notions de base que sont les classes, les propriétés et les instances. Du fait que ce modèle de connaissances est très général, plusieurs systèmes de représentation des connaissances lui sont compatibles, rendant ainsi la fusion/alignement d'ontologies basées sur PROMPT largement applicable à une grande variété de systèmes de représentation des connaissances.

- PROMPT a pour entrée deux ontologies et guide l'utilisateur pour la production en sortie d'une ontologie représentant la fusion des ontologies initiales. Au début, PROMPT crée une liste initiale de correspondances (matches) basée sur les noms des classes, puis le cycle suivant a lieu :

1. L'utilisateur déclenche une opération en sélectionnant une suggestion de PROMPT de la liste (TO DO) ou par le biais d'un éditeur d'ontologie pour spécifier l'opération désirée ;

2. PROMPT effectue l'opération et réalise automatiquement des changements supplémentaires basés sur le type d'opération choisi. Il génère ensuite une liste de suggestions basées sur la structure de l'ontologie et sur la dernière opération exécutée, détermine les conflits induits par cette dernière opération et propose des solutions possibles à ces conflits. L'utilisateur est ensuite invité à valider tout ou une partie des suggestions selon ses besoins.

Ce cycle continue ainsi jusqu'à ce que l'utilisateur soit satisfait du résultat de la fusion ou qu'il n'y-ait plus de conflits détectés [MOS, 05].

c) Constat

Prompt est plus considéré comme étant un outil qu'une méthode, son but est d'aider à la construction d'une ontologie par fusion ou alignement des ontologies existantes, il est semi-automatique dans la mesure où il ne fait que proposer des alternatives qui ne pourront être validées que par l'intervention humaine.

3.3.4. ONIONS

a) *Définition*

ONIONS (*Ontological Integration Of Naive Sources*) [STE et al, 97] fournit des directives pour analyser et fusionner des ontologies existantes et souligne la réutilisation des terminologies de domaine. Elle a été développée depuis le début des années 90 pour tenir compte du problème de l'hétérogénéité conceptuelle.

b) *Processus*

Le processus de fusion par la méthode ONIONS se passe en 6 phases :

- *Création d'un corpus de domaine valide à partir de sources textuelles:* Cette phase vise à recueillir un corpus terminologique validé pour un domaine. Une telle phase a des crochets pour les techniques de formation des corpus textuels, de la définition des types et de l'acquisition.
- *Analyse taxonomique:* Cette phase vise à découvrir la structure taxonomique principale dans les listes de termes. Au début, il faut choisir les ensembles d'expressions les plus pertinents (expressions de source) puis, la taxonomie des expressions contenues dans chaque source est déduite à partir du corpus de la 1^{ère} phase.

La taxonomie est exploitée pour identifier les concepts de haut niveau dans la source, puis les concepts de haut niveau sont utilisés pour choisir une limite de profondeur.

- *Analyse de définitions locales:* Une fois un ensemble pertinent de concepts provenant de chaque source est disponible, le processus se concentre sur les critères de classification, à savoir sur les définitions locales des concepts, afin de créer un champ interprétant qualifié. Et ce, en répondant à la question suivante :

Quel est la différence entre deux concepts frères dans un groupe de concepts homogènes de la même source ?

Lorsque les définitions font défaut, il est nécessaire de créer une définition explicite saine, exploitant tous les conseils qu'un système terminologique peut fournir (hiérarchie, groupement, définitions de texte libre, combinaisons

booléennes, axiomes, modificateurs métalinguistiques, etc.) ainsi que d'autres sources de définition (dictionnaires, glossaires, encyclopédies ou des experts).

- *Analyse de définition multi locale*: Cette phase consiste en :
 - la schématisation des éléments produits au cours de la description de la phase 3, afin de fournir des « faibles » contraintes, Cette schématisation peut être réalisée par la construction d'ontologies de domaine.
 - le déclenchement des principes conceptuels (paradigmes) qui motivent la description.
- *Construction d'une bibliothèque intégrée d'ontologie* : Pour chaque source, les définitions locales ont des paradigmes fondamentaux, le but est l'enrichissement des définitions locales en construisant les ontologies de domaine qui incluent explicitement d'autres ontologies. Un tel enrichissement n'est pas un choix arbitraire : il est fait afin de relier des définitions locales hétérogènes par l'application des ontologies génériques.

On peut résumer le déroulement de cette phase en 3 étapes :

- la construction (ou la réutilisation, si disponible) d'une bibliothèque des ontologies génériques pour expliquer les conditions mémorisées pendant la phase précédente.
 - l'enrichissement des ontologies de domaine de la phase précédente, ceci exige l'inclusion des ontologies génériques en ontologies de domaine.
 - l'attribution des catégories saines de méta-niveau aux classes et des relations dans la bibliothèque.
- *Classification de la bibliothèque*: Cette phase consiste à la mise en œuvre d'une ontologie de domaine dans un système qui permet la classification automatique.

c) Constat

L'objectif d'ONIONS est :

- Développer un ensemble bien-accordé d'ontologies génériques pour soutenir l'intégration conceptuelle des ontologies de domaine appropriés dans

la médecine. La plupart des ontologies médicales manquent d'une base sémantique, de l'axiomatisation, ou de la profondeur ontologique.

- Fournir un cadre explicite de traçage de la procédure de construction d'une ontologie, afin de faciliter son entretien (évaluation, extensions et / ou la mise à jour, et le consensus intersubjectif).

4. Analyse et synthèse

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

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

- La plupart de ces approches, sont limitées à l'utilisation d'algorithmes semi-automatiques pour la fusion d'ontologies.
- La prolifération des concepts introduits dans les ontologies engendre souvent une explosion combinatoire.
- Pour lier les concepts de différentes ontologies, ces approches s'appuient sur des similarités syntaxiques entre ces concepts. Toutefois, deux concepts peuvent avoir une même syntaxe alors que leur sémantique est différente car placée dans des contextes différents (ambiguïté). A l'inverse, deux concepts peuvent avoir la même sémantique alors qu'ils sont décrits par différentes syntaxes (synonymie).
- Enfin, ces méthodologies sont difficilement compréhensibles par les utilisateurs à cause de leur complexité.

5. Conclusion

Dans ce chapitre, nous avons parlé des différentes méthodes de calcul de similarité qui sont généralement utilisées lors du processus de fusion des ontologies.

Nous avons aussi parlé des différentes méthodes de fusion existantes et ce, dans le but de nous en inspirer pour la construction de notre approche.

Cette partie, nous a permis de nous situer dans le contexte général de notre projet de fin d'étude. Les détails spécifiques quant à l'élaboration de notre système seront abordés dans la prochaine partie.

Partie II :
Conception et
Implémentation

Chapitre III: Approche proposée

1. Introduction

Après avoir donné un aperçu général sur le domaine de la fusion des Ontologies nous allons maintenant, présenter une petite étude comparative des différentes méthodes citées précédemment et le processus général qu'on a mis au point après constats des points fort et points faibles de chacune de ces dernières.

2. Etude comparative des méthodes de fusion

Dans le chapitre précédent, nous avons vu les approches de calcul de similarité ainsi que quatre des méthodes les plus connues pour la fusion d'Ontologies. Ce qui nous a permis de faire le constat suivant :

Caractéristique		Approche	MOMIS	PROMPT	FCA-Merge	ONIONS
Terminologique	Syntaxique		Oui	Oui	Non	Oui
	Lexicale		Oui	Non	Non	Oui
Structurelle	Interne		Oui	Oui	Non	Oui
	Externe		Oui	Oui	Oui	Non
Sémantique			Oui	Non	Non	Oui
Extensionnelle			Oui	Oui	Oui	Non
Mode d'agrégation	Alignement		Non	Oui	Non	Non
	Fusion		Oui	Oui	Oui	Oui
Nature	Automatique		Non	Non	Non	Non
	Semi-automatique		Oui	Oui	Oui	Oui

Tableau 1: Etude comparative des méthodes de fusion.

Cette étude nous a permis, de faire le point sur ces différentes méthodes de fusion avec ce qu'elles offrent comme avantages et inconvénients. Aussi, elle nous a permis de dégager le processus global de fusion des Ontologies. Partant de là, notre système se vaudra être le plus optimal que possible.

3. Proposition d'une nouvelle approche de fusion d'Ontologies FOnES

D'après l'étude comparative que nous avons fait, nous avons constaté que les différentes méthodes de fusion existantes ne traitent pas vraiment de la sémantique de plus les outils qui les implémentent sont semi-automatiques. C'est-à-dire il implique l'intervention d'un expert du domaine pour la validation du processus de fusion.

Pour pallier à ces contraintes non négligeables quant à la pertinence et la souplesse dans la construction des ontologies par fusion, nous avons imaginé et mis au point le système FOnES (Fusion d'Ontologies par Enrichissement Sémantique) qui se concentre sur :

- l'ajout d'une dimension sémantique par enrichissement préalable des ontologies,
- l'automatisation de l'algorithme de fusion des ontologies initiales et de création de l'ontologie finale afin de décharger le plus possible l'utilisateur.
- la méthode est basée sur le calcul d'une similarité sémantique entre les concepts des Ontologies. Elle s'appuie sur une combinaison pondérée des méthodes de calcul de similarité existantes

Par ailleurs, notre système:

- permet à un administrateur de créer (ou d'importer) les ontologies qui vont servir lors de la fusion en alimentant une base d'ontologies,
- et ne prend en charge que les ontologies au format OWL.

Avant qu'une ontologie ne soit apte à être fusionnée via FOnES, il faut que cette dernière fasse l'objet d'un enrichissement par un expert. Au cours de cet enrichissement, les concepts de l'Ontologies seront situés dans leur contexte. C'est-à-dire que l'expert va les doter d'une définition et des synonymes qui collent le plus possible au domaine auquel se rapporte leur Ontologie et ce en s'aidant d'une base

lexicale (dictionnaire) WordNet ou en se référant à ces connaissances d'expert. Cet enrichissement aura un grand impacte sur le calcul de similarité entre les concepts issus de diverses Ontologies. On reparlera de l'enrichissement dans le 5^{ième} chapitre de ce mémoire. Le processus de fusion selon FOnES (Figure 5), se présente comme suit:

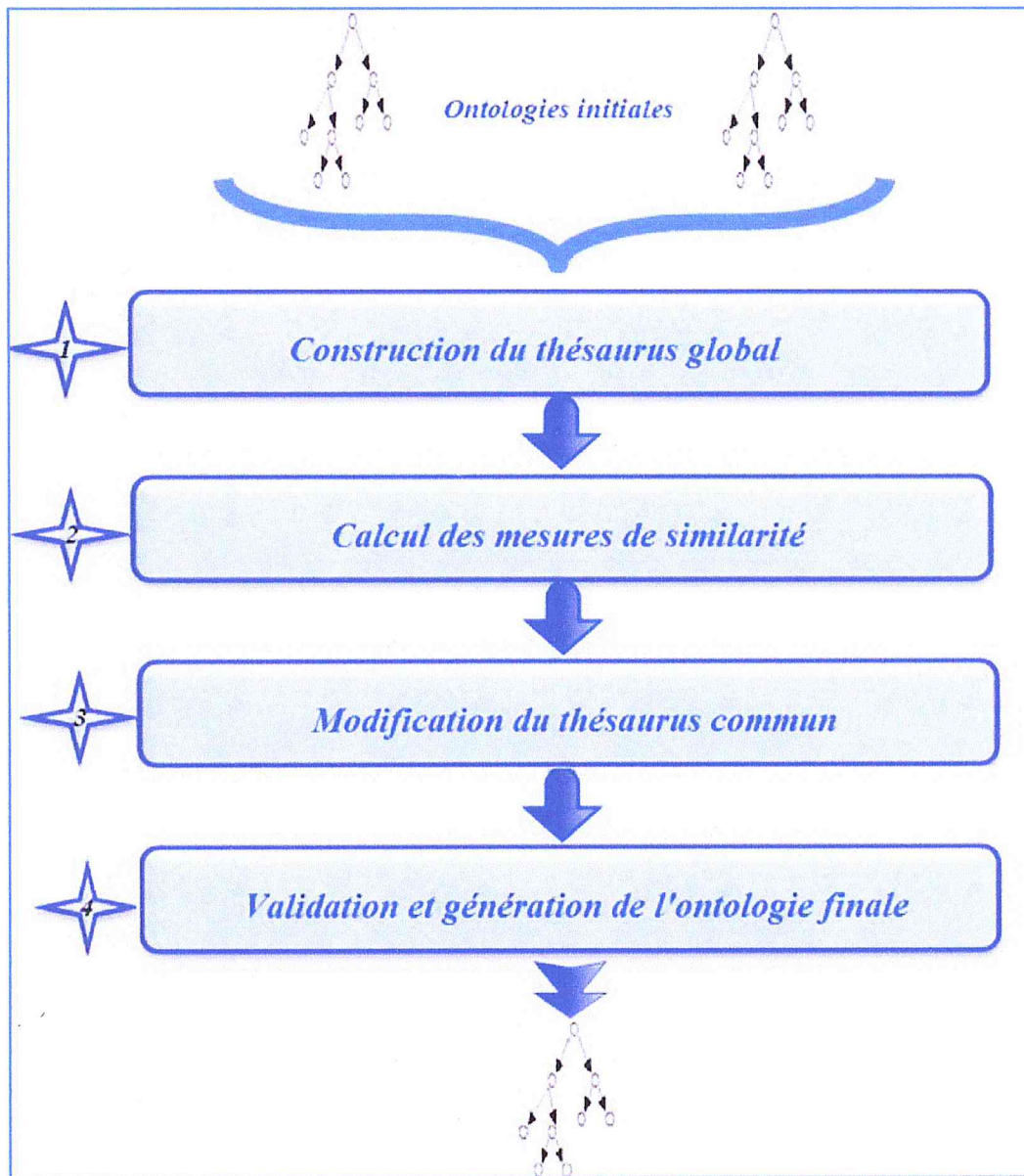


Figure 5: Processus global de la fusion par FOnES.

Etape 1:

La première étape du processus consiste à construire à partir des ontologies initiales un thésaurus global qui regroupe les différents concepts ainsi que leurs attributs, instances et relations afin d'effectuer sur ces derniers les différents traitements conduisant à la fusion.

Etape 2:

En second lieu, vient le calcul de similarité qui s'effectue entre les concepts des deux ontologies afin de mesurer leur degré de correspondance et faciliter le regroupement et la fusion de ces dernières.

Etape 3:

Après calcul des similarités entre ontologies, on modifie le thésaurus global afin qu'il représente une vision d'ensemble sur la nouvelle ontologie et ce en instaurant une nouvelle hiérarchie entre les différents concepts.

Etape 4 :

La dernière étape consiste à construire l'ontologie finale à partir du thésaurus modifié.

4. Conclusion

Au cours de ce chapitre nous avons essayé de donner le processus global de notre approche. Nous consacrerons la section qui suit à la conception du système FOnES.

Chapitre IV: Analyse des besoins et conception

1. Introduction

Après avoir donné un aperçu sur FOnES, nous allons à présent entamer la partie conception de ce dernier. En passant par une brève définition de la démarche de conception adoptée. Ensuite, on étudiera les besoins techniques et fonctionnels du système FOnES, on en fera l'analyse et on conclura le chapitre, en parlant de l'architecture statique du système.

2. Présentation de la démarche utilisée

UML est un langage qui permet de représenter des modèles, mais il ne définit pas le processus d'élaboration des modèles. Cependant, dans le cadre de la modélisation d'une application informatique, les auteurs d'UML préconisent d'utiliser une démarche.

Pour standardiser les démarches, plusieurs modèles de démarches ont été décrits et parfois formalisés, parmi ces derniers, UP.

1.1. UP (*Unified process*)

Le processus unifié est un processus de développement logiciel itératif, centré sur l'architecture, piloté par des cas d'utilisation et orienté vers la diminution des risques.

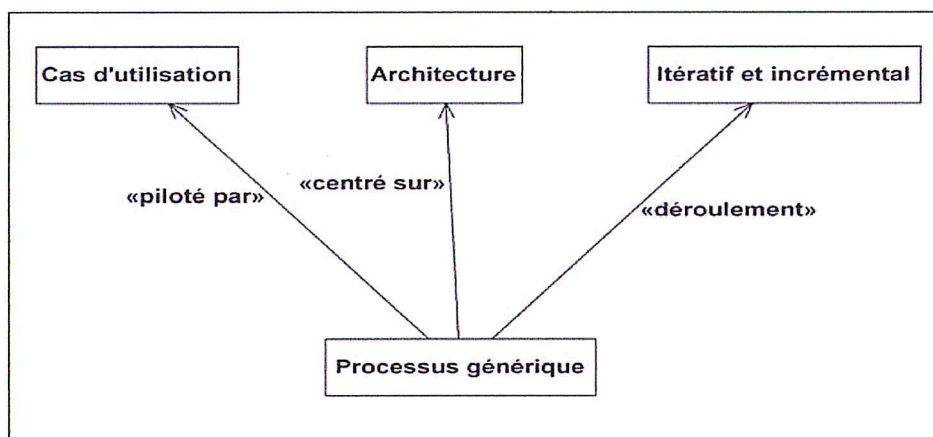


Figure 6: Processus Générique UP.

L'objectif d'un processus unifié est de maîtriser la complexité des projets informatiques en diminuant les risques.

1.2. Les activités

1.2.1. EXPSSION DES BESOINS

L'expression des besoins comme son nom l'indique, permet de définir les différents besoins :

- inventorer les *besoins* principaux et fournir une liste de leurs fonctions.
- recenser les *besoins fonctionnels* (du point de vue de l'utilisateur) qui conduisent à l'élaboration des modèles de cas d'utilisation.
- appréhender les *besoins non fonctionnels* (techniques) et livrer une liste des exigences.

Le modèle de cas d'utilisation présente le système du point de vue de l'utilisateur et représente sous forme de cas d'utilisation et d'acteur, les besoins du client.

1.2.2. Analyse

L'objectif de l'analyse est d'accéder à une compréhension des besoins et des exigences du client. Il s'agit de livrer des spécifications pour permettre de choisir la conception de la solution.

Un modèle d'analyse livre une spécification complète des besoins issus des cas d'utilisation et les structure sous une forme qui facilite la compréhension (scénarios), la préparation (définition de l'architecture), la modification et la maintenance du futur système.

Il s'écrit dans le langage des développeurs et peut être considéré comme une première ébauche du modèle de conception.

1.2.3. Conception

La conception permet d'acquérir une compréhension approfondie des contraintes liées au langage de programmation, à l'utilisation des composants et au système d'exploitation.

Elle détermine les principales interfaces et les transcrit à l'aide d'une notation commune.

Elle constitue un point de départ à l'implémentation :

- elle décompose le travail d'implémentation en sous-système

- elle crée une abstraction transparente de l'implémentation

1.2.4. Implémentation

L'implémentation est le résultat de la conception pour implémenter le système sous formes de composants, c'est-à-dire, de code source, de scripts, de binaires, d'exécutables et d'autres éléments du même type.

Les objectifs principaux de l'implémentation sont de planifier les intégrations des composants pour chaque itération, et de produire les classes et les sous-systèmes sous formes de codes sources.

1.2.5. Test

Les tests permettent de vérifier des résultats de l'implémentation en testant la construction. Pour mener à bien ces tests, il faut les planifier pour chaque itération, les implémenter en créant des cas de tests, effectuer ces tests et prendre en compte le résultat de chacun.

2. Expression des besoins

2.1. Recueil des besoins fonctionnels (diagramme de cas d'utilisation)

2.1.1. Identification des acteurs

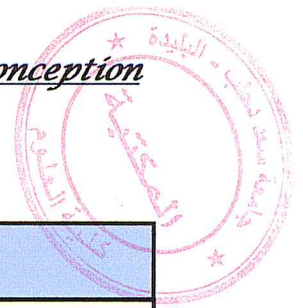
La première étape de cette phase est d'énumérer les Acteurs susceptibles d'interagir avec le système. Un **Acteur** représente l'abstraction d'un rôle joué par des entités externes (utilisateur, dispositif matériel ou autre système), qui interagissent directement avec le système étudié.

Le tableau ci-dessous identifie les acteurs de notre système et décrit la mission de chacun d'eux.

- Les Acteurs

Acteur	Désignation
<i>Utilisateur</i>	Les utilisateurs tiers sont des utilisateurs non privilégiés.
<i>Administrateur</i>	L'administrateur est un utilisateur privilégié qui a un rôle principal dans notre système.
<i>Expert</i>	L'expert est un utilisateur privilégié qui joue un rôle primordial pour la fusion des Ontologies.

Tableau 2: Les acteurs du système.



- Description des besoins fonctionnels

Acteur	Description des besoins fonctionnels
<i>Utilisateur tiers</i>	L'application doit permettre aux utilisateurs tiers de : <ul style="list-style-type: none"> • Sélectionner des Ontologies à partir de la base d'Ontologies • Lancer le processus de fusion des Ontologies • Visualiser le résultat de fusion
<i>Administrateur</i>	L'application doit permettre à l'administrateur de : <ul style="list-style-type: none"> • Alimenter la base d'Ontologies soit : <ul style="list-style-type: none"> - En important des Ontologies existantes - En construisant de nouvelles Ontologies • Sélectionner des Ontologies à partir de la base d'Ontologies • Lancer le processus de fusion des Ontologies • Visualiser le résultat de la fusion
<i>Expert</i>	L'application doit permettre aux experts de : <ul style="list-style-type: none"> • Sélectionner des Ontologies à partir de la base d'Ontologies • Enrichir les Ontologies • Lancer le processus de fusion des Ontologies • Visualiser le résultat de la fusion

Tableau 3:Description des besoins fonctionnels.

2.1.2. Identification des cas d'utilisation

L'identification des cas d'utilisation, donne un aperçu des fonctionnalités futures que doit implémenter le système. Cependant, il nous faut plusieurs itérations pour ainsi arriver à constituer des cas d'utilisation complets. D'autres cas d'utilisation vont apparaître au fur à mesure de la description de ceux là, et l'avancement dans le « recueil des besoins fonctionnels ».

Dans notre cas d'étude nous avons défini les diagrammes de cas d'utilisation généraux et détaillés. Pour les cas globaux, on distingue:

- Authentification (pour les utilisateurs privilégiés).
- Alimentation de la base d'Ontologies.
- Enrichissement des Ontologies.
- Visualisation des résultats de fusion.

Nous détaillerons chaque cas d'utilisation qui doit faire l'objet d'une définition a priori qui décrit l'intention de l'acteur lorsqu'il utilise le système, et les séquences d'actions principales qu'il est susceptible d'effectuer. Ces définitions servent à fixer les idées et n'ont pas pour but de spécifier un fonctionnement complet et irréversible.

Les descriptions détaillées vont être organisées dans des tableaux de la façon suivante :

Elément	Signification
<i>Cas d'utilisation</i>	Le nom du cas d'utilisation
<i>Acteur</i>	L'acteur qui réalise le cas d'utilisation
<i>But</i>	Le but du cas d'utilisation
<i>Description</i>	Une explication du cas d'utilisation
<i>Pré condition</i>	Les conditions qui doivent être vérifiées afin d'effectuer le cas d'utilisation
<i>Post condition</i>	Les résultats du cas d'utilisation
<i>Exception</i>	Les informations entrées par l'acteur

Tableau 4:Modèle de représentation des descriptions détaillées des cas d'utilisations.

2.1.3. Diagrammes de cas d'utilisation

- Diagramme de cas d'utilisation global

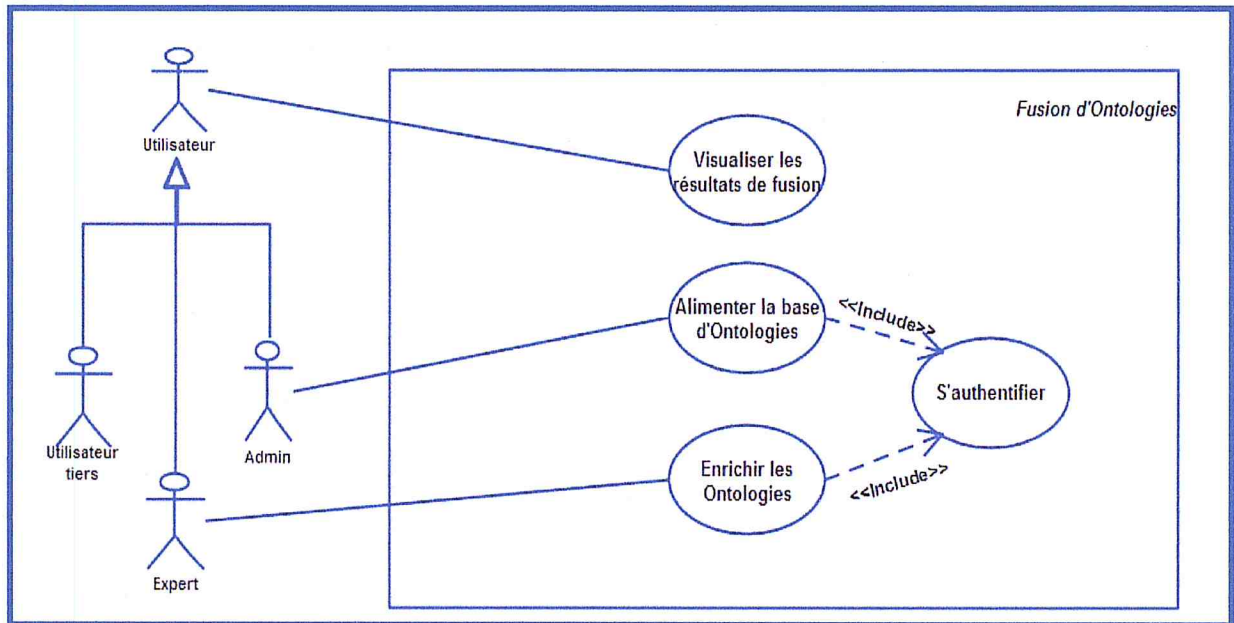


Figure 7: Diagramme de cas d'utilisation global du système de fusion d'Ontologies.

Authentification	
Acteur	Utilisateur privilégié
But	Accéder à l'interface de traitements privilégiés.
Description	L'accès à l'espace privilégié doit passer par l'interface d'authentification qui contient les renseignements adéquats.
Pré condition	L'utilisateur privilégié doit fournir les informations correctes (le nom d'utilisateur et le mot de passe).
Post condition	L'utilisateur privilégié peut effectuer les différentes tâches qui lui sont permises (par rapport à sa fonction).
Exception	Annulation ; si l'utilisateur privilégié saisit un nom d'utilisateur ou un mot de passe incorrecte, le système affiche un message d'erreur.
Visualisation des résultats de fusion	
Acteur	Utilisateurs
But	Obtention d'une nouvelle Ontologie par fusion des ontologies existantes.

Description	La visualisation du résultat de fusion doit passer par la sélection des Ontologies qui vont être fusionnées et du lancement du processus.
Pré condition	L'utilisateur doit sélectionner deux Ontologies parmi les Ontologies qui se trouvent dans la base d'Ontologies.
Post condition	L'utilisateur peut visualiser la nouvelle Ontologie (taxonomie, concepts, propriétés et instances) avec possibilité d'impression ou d'exportation de la taxonomie sous forme d'image et de sauvegarde de l'Ontologie finale.
Alimentation de la base d'Ontologies	
Acteur	Administrateur
But	Ajouter de nouvelle Ontologies à la base d'Ontologies pour offrir une large palette de choix de fusion pour les différents utilisateurs du système
Description	L'alimentation de la base d'Ontologie ne peut être effectuée que par l'administrateur dans un cadre privilégié.
Pré condition	L'administrateur, après s'être authentifié peut soit : <ul style="list-style-type: none"> - Sélectionner des Ontologies déjà créées - Construire de nouvelles Ontologies (par éditeur graphique ou textuel)
Post condition	L'administrateur sauvegarde les ontologies importées (respectivement, construite)
Enrichissement d'une Ontologie	
Acteur	Expert
But	Renforcer la dimension sémantique ce qui sera très utile lors de la fusion des Ontologies
Description	L'enrichissement consiste à donner la définition d'un concept qui se rapproche le plus au domaine auquel il appartient.
Pré condition	Avant d'enrichir une Ontologie il faudra d'abord la sélectionner à partir de la base d'Ontologies.

Post condition	Le système va générer automatiquement pour chaque Ontologie enrichie un thésaurus local.
-----------------------	--

Tableau 5: Description détaillée du diagramme de cas d'utilisation global.

- Diagramme de cas d'utilisation « Visualisation du résultat de fusion »

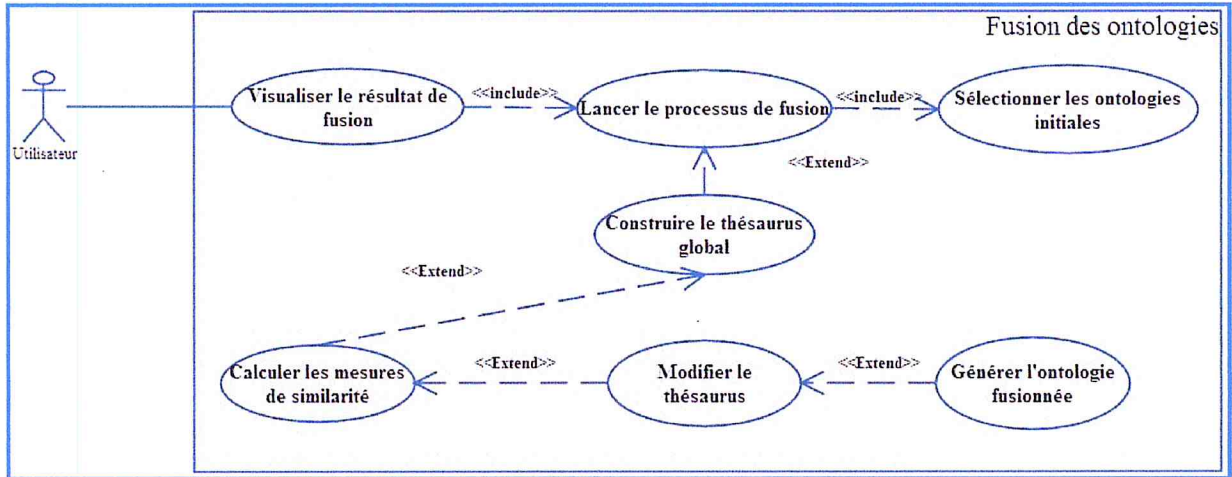


Figure 8: Diagramme de cas d'utilisation détaillé "Visualisation du résultat de fusion".

Lancement du processus de fusion	
Acteur	Utilisateur
But	Obtenir une nouvelle Ontologie de domaine.
Description	L'obtention de la nouvelle Ontologie est réalisée par fusion des Ontologies existantes.
Pré condition	L'utilisateur sélectionne les deux Ontologies qu'il veut fusionner
Post condition	L'utilisateur peut lancer le processus de fusion afin de visualiser les résultats de fusion.
Exception	Annulation, si l'utilisateur sélectionne une seule Ontologie (ou pas d'ontologie) et lance le processus, le système lui affiche un message d'erreurs.
Sélection des Ontologies	
Acteur	Utilisateurs
But	Lancer le processus de fusion

<i>Description</i>	Le lancement du processus de fusion doit être précédé de la sélection des deux Ontologies de domaine qui vont constituer la base de la nouvelle Ontologie.
<i>Pré condition</i>	La base d'Ontologie ne doit pas être vide
<i>Post condition</i>	L'utilisateur lance le processus de fusion
<i>Exception</i>	Annulation, si la base d'Ontologies est vide le système affiche un message d'erreur
Construction du thesaurus	
<i>Réalisé par</i>	Système
<i>But</i>	Conserver toutes les informations relatives aux ontologies initiales
<i>Description</i>	La première étape du processus de fusion est la création du thésaurus global qui va être primordial tout au long de la fusion
<i>Pré condition</i>	Pour que le thésaurus puisse être construit il faut que les informations des ontologies initiales soient sauvegarder dans la base de données
Calculer les mesures de similarité	
<i>Réalisé par</i>	Système
<i>But</i>	Voir le degré de correspondances entre les concepts des deux ontologies
<i>Description</i>	Le calcul des mesures de similarité va nous aider pour connaître les concepts qui seront fusionnés
<i>Pré condition</i>	Il faut que le thésaurus soit construit d'une manière méticuleuse
Modifier le thésaurus global	
<i>Réalisé par</i>	Système
<i>But</i>	Construire l'ontologie fusionnée
<i>Description</i>	Obtenir des propriétés globales (qui n'appartiennent plus aux ontologies locales) et qui constitue les composants de la nouvelle ontologie.

Pré condition	Pour fusionner deux concepts il faut que leur similarité soit supérieure à un certain seuil fixé préalablement.
Générer l'ontologie finale	
Réalisé par	Système
But	Obtention de notre nouvelle ontologie
Description	La génération de l'ontologie finale se fait à partir des différentes propriétés contenues dans le thésaurus modifié.
Pré condition	Il faut disposer de tous les composants nécessaires à la construction d'une nouvelle ontologie. En n'omettant aucun détail.
Post condition	Le système affiche l'ontologie finale à l'utilisateur pour qu'il puisse la visualiser.

Tableau 6:Description détaillée de la visualisation du résultat de fusion.

- Diagramme de cas d'utilisation détaillé « Alimentation de la base d'Ontologies »

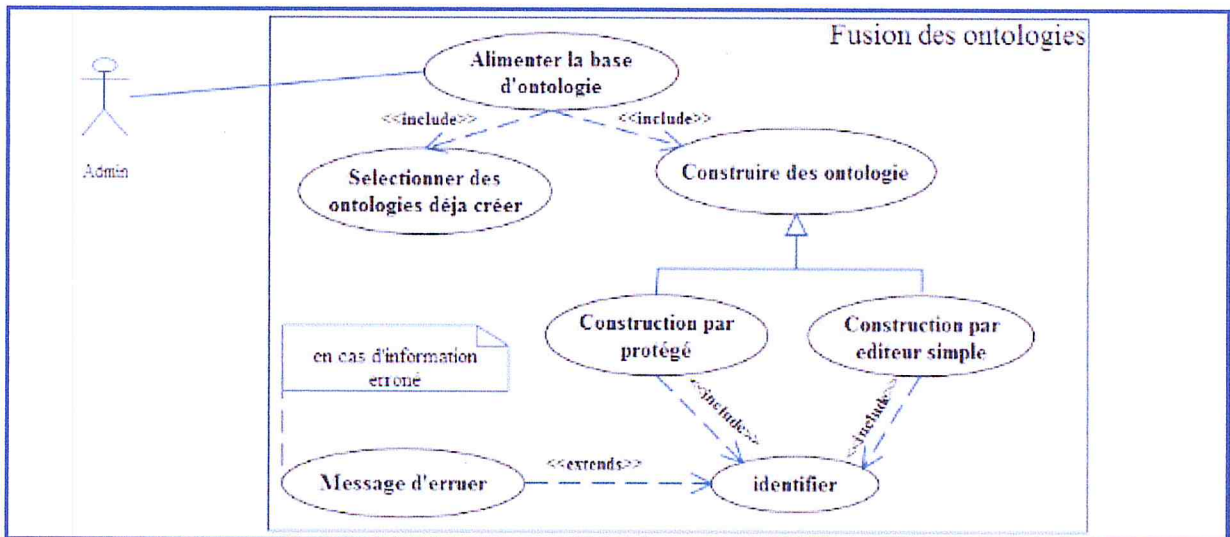


Figure 9:Diagramme de cas d'utilisation détaillé "Alimentation de la base d'Ontologies".

Sélection des Ontologies déjà créés	
Acteur	Administrateur
But	Donner aux utilisateurs du système des Ontologies de départ qu'ils pourront fusionner pour en créer de nouvelles.

Description	L'administrateur peut choisir n'importe quel Ontologie déjà construite (qui se trouve n'importe où dans le système) et la mettre dans la base d'Ontologies
Pré condition	L'existence des Ontologies.
Post condition	Visualisation de l'Ontologie.
Exception	Annulation, s'il n'existe pas d'Ontologies déjà créés l'administrateur se devra de les créer par lui-même.
Construction par Protégé	
Acteur	Administrateur
But	Donner aux utilisateurs du système des Ontologies de départ qu'ils pourront fusionner pour en créer de nouvelles.
Description	L'administrateur peut utiliser Protégé (qui est un éditeur graphique de construction d'Ontologies) pour construire ses Ontologies. Protégé lui permettra d'effectuer pas mal de tâche sur l'Ontologie de manière souple.
Pré condition	Le logiciel Protégé doit être installé dans le système
Post condition	Une fois que l'administrateur construit l'Ontologie, il pourra l'enregistrer dans la base d'Ontologies.
Construction par Editeur textuel	
Acteur	Administrateur
But	Donner aux utilisateurs du système des Ontologies de départ qu'ils pourront fusionner pour en créer de nouvelles.
Description	L'administrateur peut utiliser un éditeur textuel que met à sa disposition le système afin de construire une Ontologie (l'administrateur se devra alors, avoir des notions sur la syntaxe OWL).
Pré condition	Informations sur l'Ontologie à construire
Post condition	Une fois que l'administrateur construit l'Ontologie, il pourra l'enregistrer dans la base d'Ontologies.

Tableau 7: Description détaillée de l'alimentation de la base d'Ontologies.

- Diagramme de cas d'utilisation détaillé « enrichissement d'Ontologies »

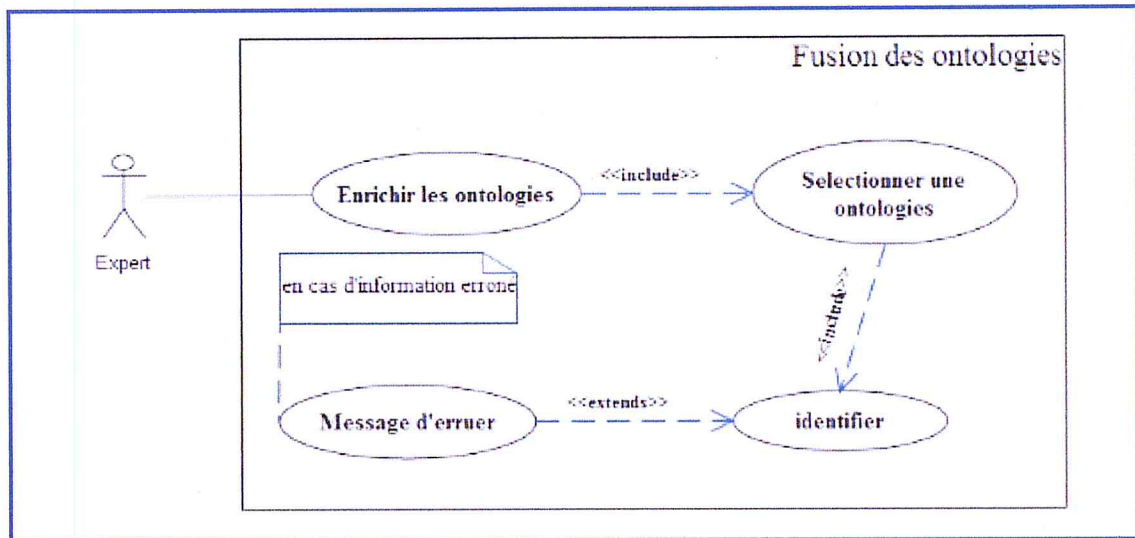


Figure 10: Diagramme de cas d'utilisation détaillé "Enrichissement des Ontologies".

Sélection d'une Ontologie	
Acteur	Expert
But	Renforcer la dimension sémantique et du coup la précision de l'Ontologie créée par fusion.
Description	L'expert peut enrichir une Ontologie, en sélectionnant parmi une liste de proposition, la définition qui se rapproche le plus du domaine pour chaque concept de l'Ontologie.
Pré condition	La base d'Ontologie ne doit pas être vide.
Post condition	Débuter l'enrichissement de l'Ontologie.
Exception	Annulation, si la base d'Ontologies est vide, l'enrichissement ne pourra pas se faire et donc le système affiche un message d'erreur.

Tableau 8: Description détaillée de l'enrichissement des Ontologies.

2.2. Recueil des besoins techniques

La capture des besoins techniques recense toutes les contraintes et les choix dimensionnant la conception du système.

Parmi les besoins techniques principaux, on recense :

- L'application doit être réalisée dans un environnement Open source, ce qui offre des outils de développement gratuits et avec la possibilité de modifier le code source de ces derniers.
- Les Ontologies sont écrites au format OWL.
- Pour gérer les Ontologies on va utiliser différentes APIs :
 - **Jena** pour l'exploitation,
 - **JWS** pour le calcul de similarité structurelle,
 - **JAWS** pour la spécification des différentes définitions (et synonymes) d'un concept donné,
 - **JFreeChart** pour la réalisation de graphes statistiques
 - **OWL2Perfuse** pour la génération de la taxonomie des Ontologies
 - L'utilisation de ces APIs sera expliquée en détail dans le chapitre « Implémentation ».

3. Analyse du système

3.1. Scénarios et diagrammes de séquences

3.1.1. Authentification

- *Scénario et diagramme de séquences « Authentification » (le cas normal)*
 1. L'utilisateur se connecte au système.
 2. Le système affiche la boîte d'authentification.
 3. L'utilisateur saisi ses informations personnelles (nom d'utilisateur et mot e passe).
 4. Le système vérifie l'identité de l'utilisateur.
 5. Le système confirme l'identité de l'utilisateur.
 6. Le système Attribue l'accès privilégié à l'utilisateur.

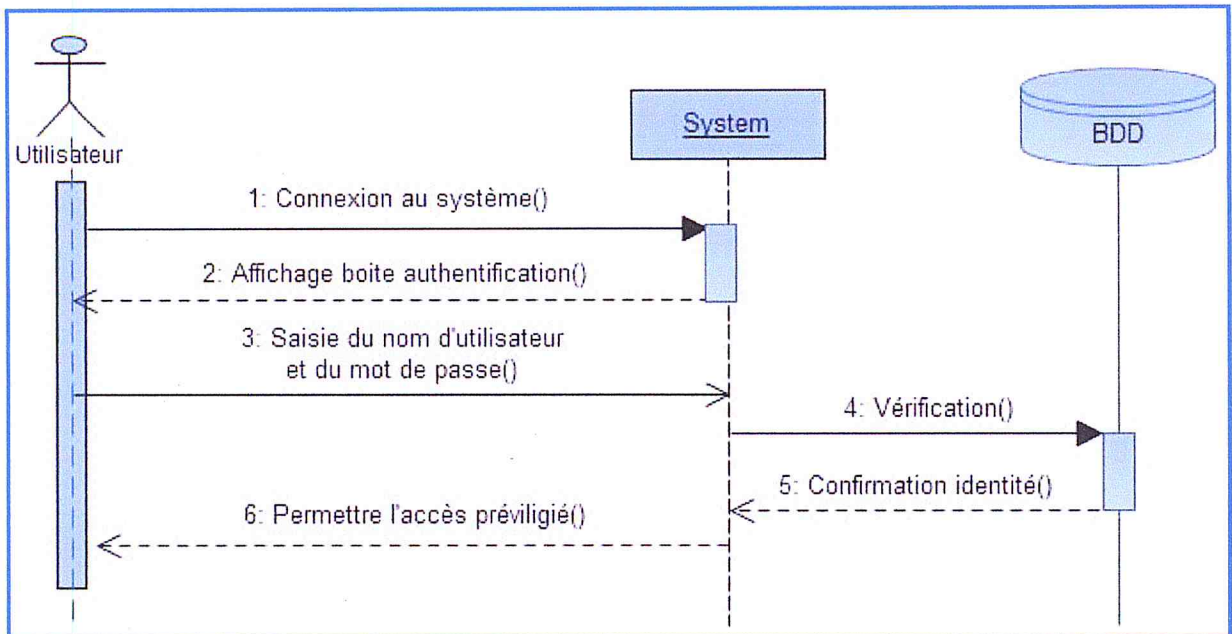


Figure 11: Diagramme de séquences "Authentification" (cas n°1).

- Scénario et diagramme de séquences « Authentification » (le cas d'erreur)

1. L'utilisateur se connecte au système.
2. Le système affiche la boîte d'authentification.
3. L'utilisateur saisi ses informations personnelles (nom d'utilisateur et mot e passe).
4. Le système vérifie l'identité de l'utilisateur.
5. Déni de l'identité de l'utilisateur.
6. Afficher un message d'erreur.

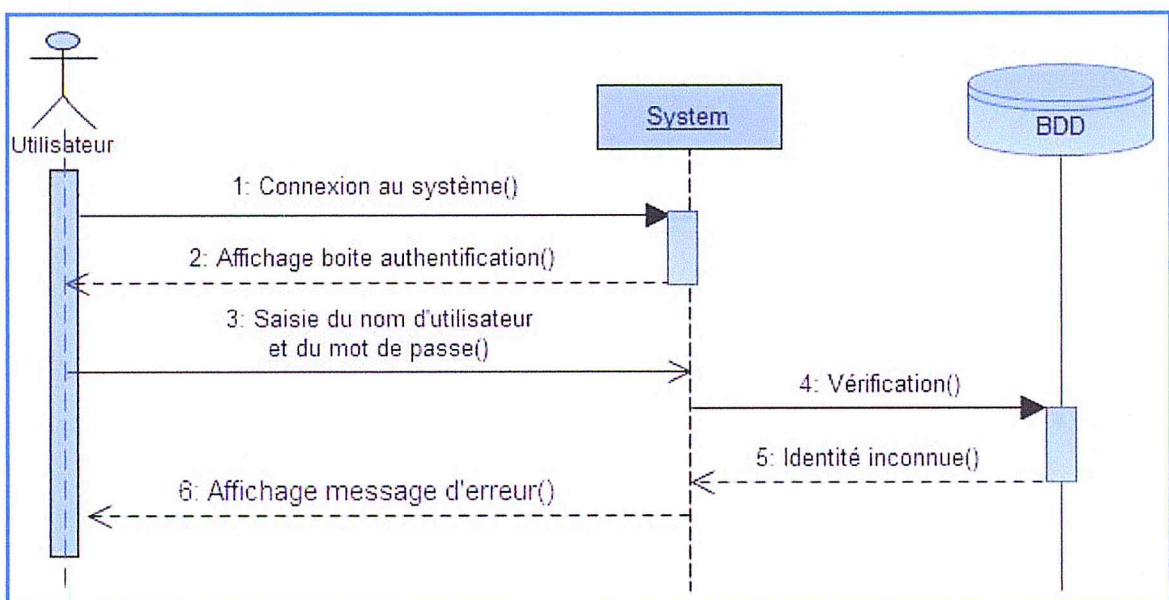


Figure 12: Diagramme de séquences "Authentification" (cas n°2).

3.1.2. Visualisation des résultats de fusion

- *Scénario et diagramme de séquences « visualisation de la fusion d'Ontologies »*
 1. Demande de la liste d'Ontologie
 2. Le système cherche la liste des ontologies dans la base d'ontologies.
 3. La base d'ontologies renvoi le résultat de recherche.
 4. Le système affiche la liste des ontologies.
 5. L'utilisateur sélectionne les ontologies à fusionner.
 6. Le système affiche les ontologies.
 7. L'utilisateur lance le processus de fusion.
 8. Le système accède à la base de données pour récupérer les thesaurus locaux.
 9. La base de données renvoi le résultat.
 10. Le système construit le thesaurus global.
 11. Le système calcule la similarité
 12. Le système modifie le thesaurus.
 13. Le système construit la nouvelle ontologie.
 14. Le système affiche le résultat
 15. Sauvegarde de l'ontologie résultat dans la base d'ontologies.

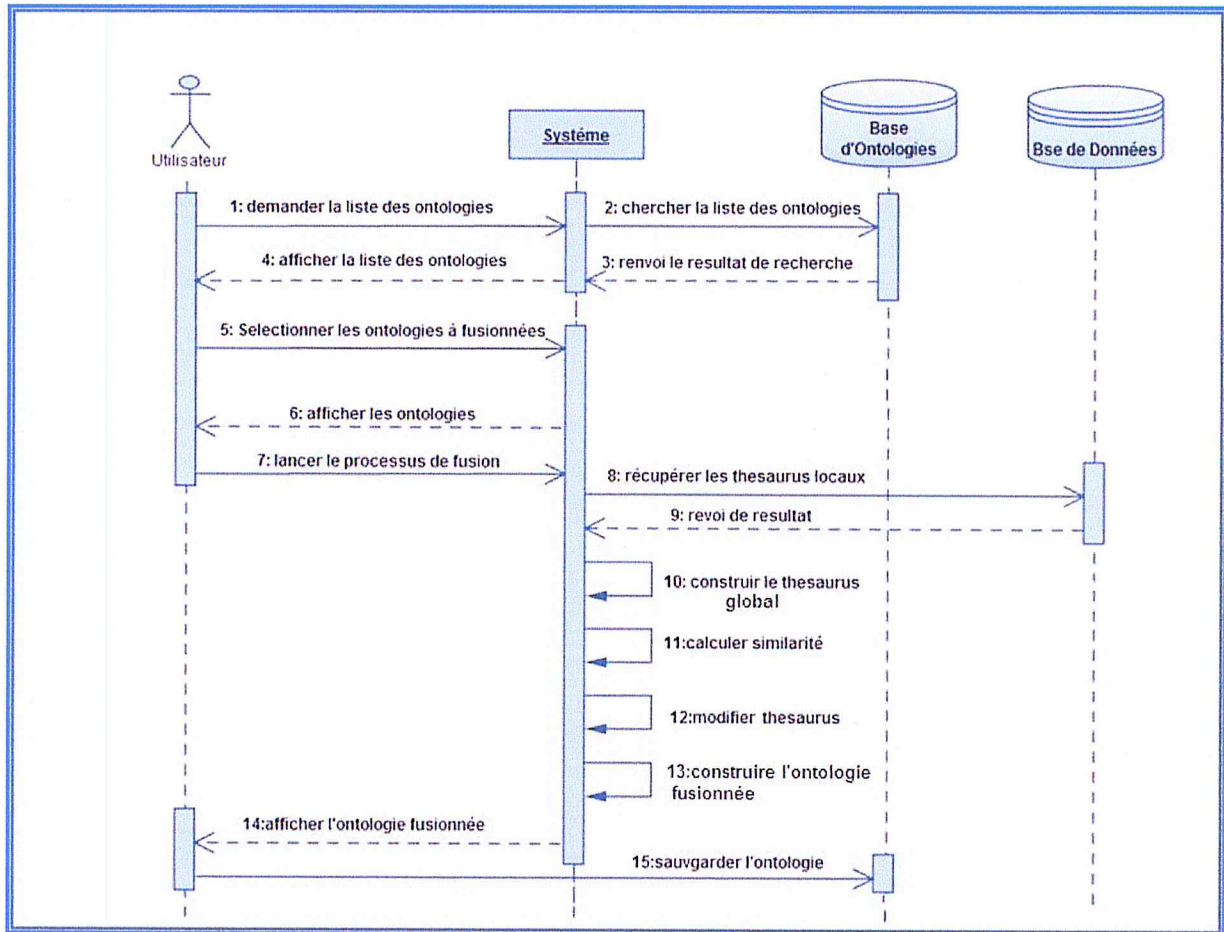


Figure 13: Diagramme de séquences Visualisation du résultat de fusion.

3.1.3. Alimentation de la base d'ontologie

- Scénario et diagramme de séquences « Alimentation de la base d'Ontologies » (cas de la sélection des Ontologies existantes)

1. L'administrateur sélectionne l'Ontologie.
2. Le système charge l'ontologie.
3. Le système affiche l'Ontologie pour que l'utilisateur la visualise.
4. L'administrateur clique sur enregistrer.
5. Le système sauvegarde l'ontologie dans la base d'Ontologies.

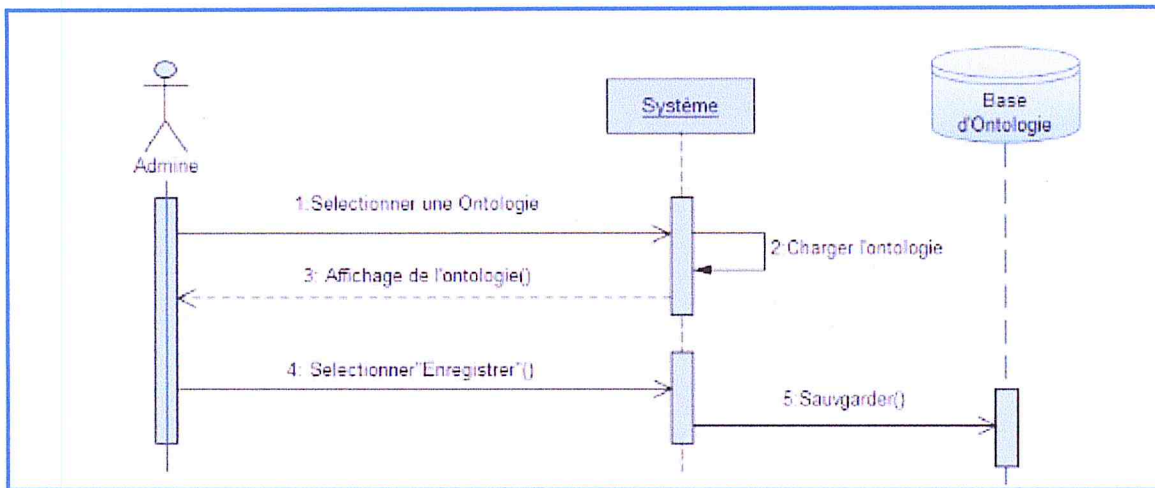


Figure 14: Diagramme de séquences Alimentation de la base d'Ontologies (cas n°1.)

• Scénario et diagramme de séquences « Alimentation de la base d'Ontologies » (cas de la construction par protégé)

1. L'administrateur demande l'accès à protéger.
2. Le système charge la fenêtre de protégé.
3. Le système affiche la fenêtre de Protégé
4. L'administrateur crée l'ontologie et clique sur Enregistrer.
5. Le système sauvegarde l'Ontologie dans la base d'Ontologies.

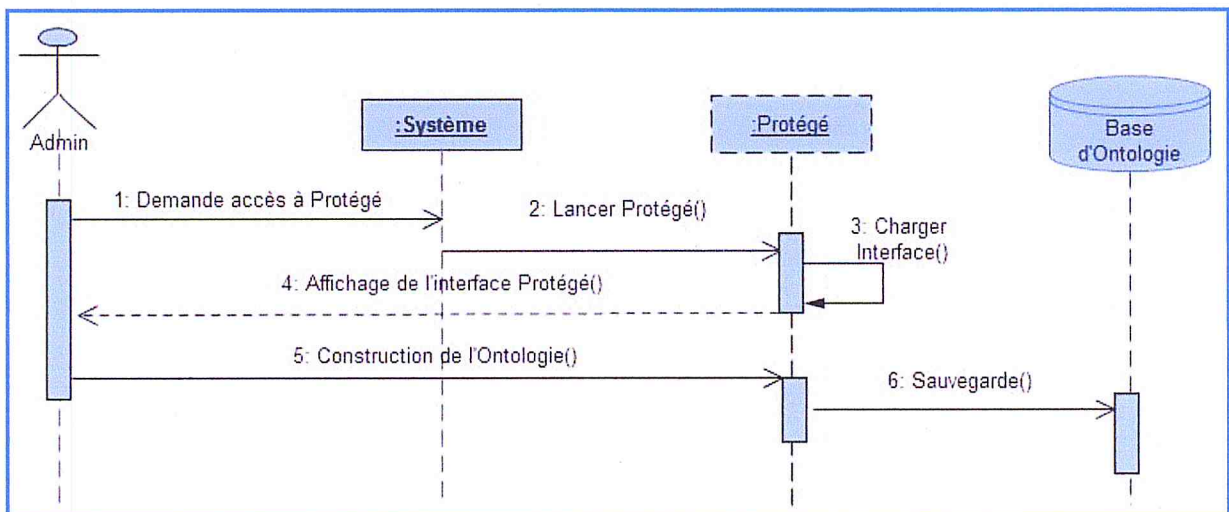


Figure 15: Diagramme de séquences Alimentation de la base d'Ontologies (cas n°2).

• Scénario et diagramme de séquences « Alimentation de la base d'Ontologies » (cas de la construction par éditeur textuel)

1. L'administrateur demande l'accès à l'éditeur du système.
2. Le système affiche l'éditeur.
3. L'administrateur crée l'ontologie et décide de l'enregistrer.

4. Le système sauvegarde l'ontologie dans la base d'ontologie.

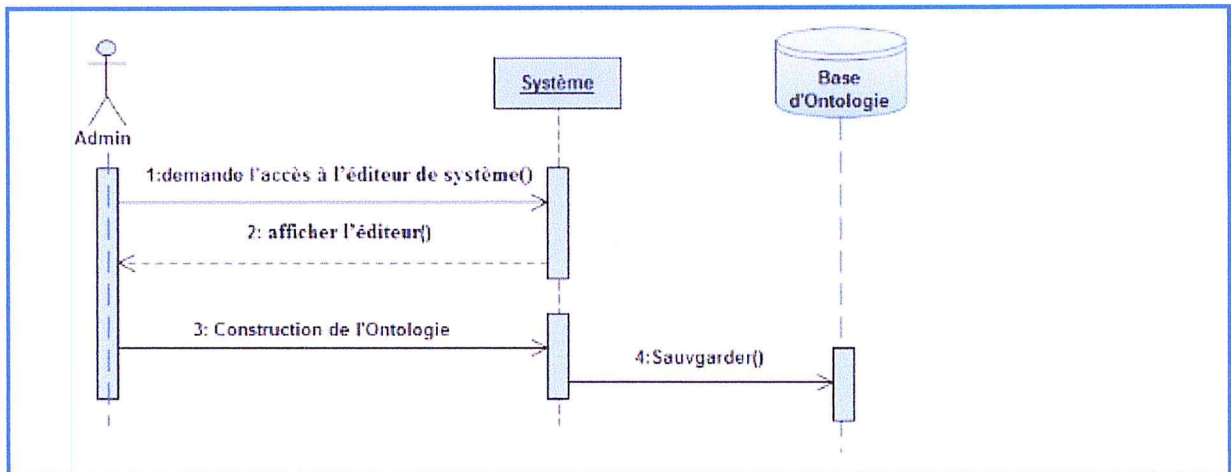


Figure 16: Diagramme de séquences Alimentation de la bases d'Ontologies (cas n°3).

4.1.4. Enrichissement d'une ontologie

- Scénario et diagramme de séquences « Enrichissement d'une Ontologie »
 1. L'expert sélectionne une ontologie.
 2. Le système demande l'Ontologie de la base d'Ontologies.
 3. L'ontologie est chargée puis affichée à l'Utilisateur.
 4. L'expert lance le processus d'enrichissement.
 5. Le système extrait les concepts et propose pour chaque concept, des définitions et synonyme.
 6. L'expert choisi la définition la plus proche du domaine ou propose sa propre définition.
 7. Le système génère un thésaurus pour l'Ontologie et le sauvegarde.

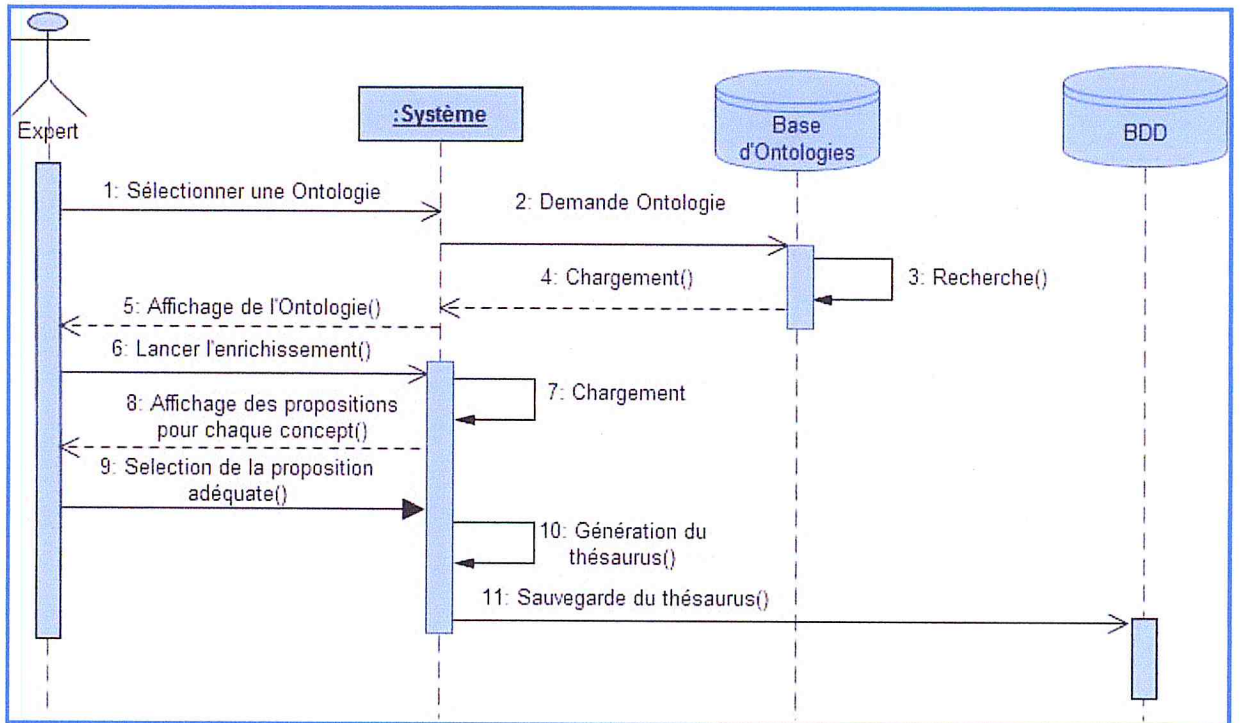


Figure 17: Diagramme de séquences "Enrichissement des Ontologies".

4. Conception du système

4.1. Architecture du système

L'architecture globale du système FOnES (Figure 18), se présente comme suit :

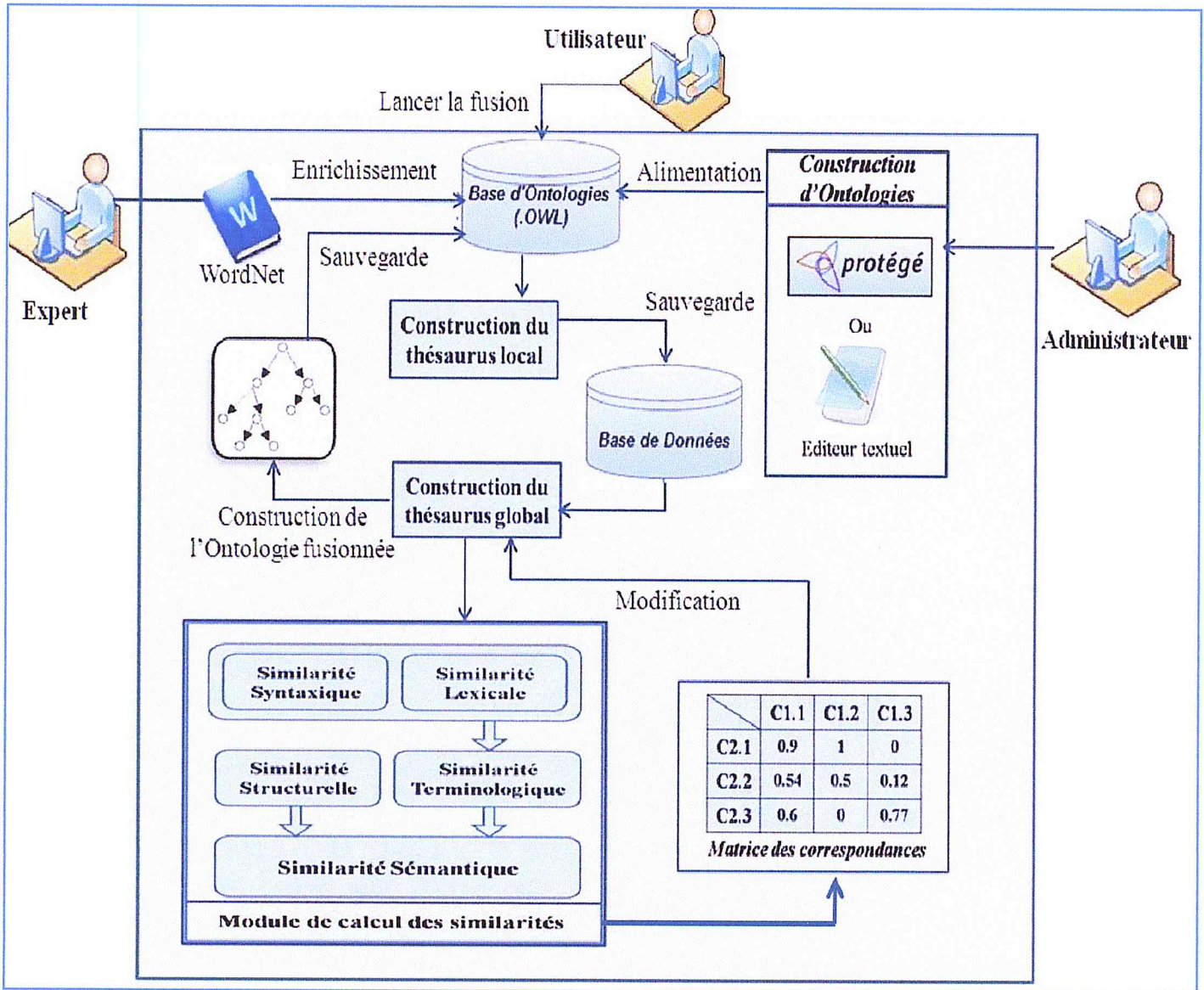


Figure 18: Architecture du système.

Dans les sections suivantes de ce chapitre, nous allons nous intéresser à chaque module du système individuellement afin d'apporter plus amples explications.

Module1 : « Construction des ontologies et alimentation de la base d'Ontologies »

L'administrateur construit les ontologies qui vont servir pour la création des nouvelles Ontologies par fusion, la construction pourra être faite soit par un éditeur graphique « Protégé »⁽²⁾, soit par un éditeur textuel (intégré dans notre système).

Après leur construction, les Ontologies seront stockées (par l'administrateur) dans un emplacement spécifique dans le disque dur et qui va constituer « la base d'ontologies ». Cette opération est dite *Alimentation de la base d'ontologies*. Cette dernière, peut par ailleurs, se faire par l'importation d'Ontologies toutes faites.

Module2 : « Enrichissement de l'ontologie et Génération du Thésaurus local »

Après l'étape de l'alimentation vient l'étape d'enrichissement. Cette dernière est effectuée par un expert qui possède des connaissances du domaine auquel se rapporte l'Ontologie.

L'expert accède à la base d'ontologies, il sélectionne une Ontologie, et il lance le processus d'enrichissement. Le système parcourt l'Ontologie afin d'extraire tous ses concepts et propose à l'expert, pour chaque concept, toutes les définitions et les synonymes qui lui sont relatifs et ce, à l'aide de WordNet et enfin, l'expert choisit la définition la plus proche du domaine.

Si le système se trouve face à un concept composé ou complexe et auquel Wordnet ne propose aucune définition, l'expert pourra donner sa propre définition (toujours en relation avec le domaine).

Une fois l'ontologie enrichie, le système lui génère un thésaurus appelé « *Thésaurus local* » qui répertorie pour chaque Ontologie, l'ensemble de ses concepts $TH = \{concept_1, concept_2, \dots, concept_n\}$. Avec pour chaque concept: sa définition, son concept père, son terme T , ses attributs Att_i , ses relations R_i avec les autres concepts. Le thésaurus local sera stocké dans une base de données.

Module 3 : « Construction de thesaurus global »

Lors de la fusion, l'utilisateur (administrateur, expert et utilisateur tiers) sélectionne les ontologies initiales, le système accède alors à la base de données,

² Protégé est un éditeur d'ontologies distribué en open source qui permet la manipulation de divers formats d'ontologies d'une manière graphique.

pour récupérer les thésaurus locaux et génère un thésaurus global par concaténation de ces derniers.

Module4 : « Calcul de similarité »

Lors de cette étape, nous allons calculer les différentes mesures de similarité qui vont nous permettre de faire les correspondances entre les deux ontologies initiales. Pour ce faire, on a choisit une approche combinée de calcul des mesures.

Ceci s'effectuera en trois étapes

- **Calcul de similarités terminologiques**

Les mesures à base de comparaison syntaxique et lexicale sont appliquées sur des couples de concepts afin de mesurer leur similarité terminologique.

Dans notre système on va utiliser la distance de JARO pour la similarité syntaxique et WordNet pour la similarité lexicale.

- *Similarité syntaxique*: La **distance de Jaro** mesure la similarité entre deux chaînes de caractères. Plus la distance de Jaro entre deux chaînes est élevée, plus elles sont similaires.

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

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

$$\text{Sim}_{\text{syn}} = \frac{1}{3} \left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m} \right)^{(3)}$$

Où:

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

Deux caractères identiques de s_1 et de s_2 sont considérés comme *correspondants* si leur éloignement (i.e. la différence entre leurs positions dans leurs chaînes respectives) ne dépasse pas :

³ La formule de Jaro a été proposée en 1989 par Matthew A. Jaro et qui se base principalement utilisée sur la détection de doublons.

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

Le nombre de transpositions est obtenu en comparant le $i^{\text{ème}}$ caractère *correspondant* de s_1 avec le $i^{\text{ème}}$ caractère *correspondant* de s_2 . Le nombre de fois où ces caractères sont différents, divisé par deux, donne le nombre de *transpositions*.

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

WordNet est une ressource lexicale de langue anglaise, disponible sur internet, qui regroupe des termes (noms, verbes, adjectifs et adverbes) en ensembles de synonymes appelés *synsets*. Un *synset* regroupe tous les termes dénotant un concept donné.

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

Pour le calcul de la similarité linguistique la fonction $Syn(c)$ calcule l'ensemble des *Synsets* de WordNet du concept c retenus après l'enrichissement de l'ontologie; soit $S = Syn(c_1) \cap Syn(c_2)$ l'ensemble des sens communs entre c_1 et c_2 à comparer, la cardinalité de S est :

$$\lambda(S) = |Syn(c_1) \cap Syn(c_2)|;$$

Soit $\min(|Syn(c_1)|, |Syn(c_2)|)$ le minimum entre les cardinalité des deux ensembles $Syn(c_1)$ et $Syn(c_2)$ alors nous avons défini la similarité entre deux concepts c_1 et c_2 comme suit :

$$Simlex(c_1, c_2) = \lambda(S) / \min(|Syn(c_1)|, |Syn(c_2)|);$$

- Nous avons ensuite, combiné ces deux mesures. Et le résultat est formule :

$$Sim_{term}(c_1, c_2) = \frac{(Simlex(c_1, c_2) * Coefflex) + (Simsyn(c_1, c_2) * Coeffsyn)}{Coefflex + Coeffsyn}$$

Où :

Coeff_x : est un coefficient numérique donné pour chaque mesure. nous le calculons de la manière suivante :

$$\text{Coeff} = \text{Exp}^{\text{Sim}}$$

- **Calcul de similarité Structurale :**

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

$$\text{Sim}_{wp}(c1, c2) = \frac{\text{profondeur}(\text{LCA}(c1, c2))}{\text{profondeur}(c1) + \text{profondeur}(c2)} \quad (4)$$

Où :

LCA(c₁, c₂) est le petit ancêtre commun (Lowest Common Ancestors de c₁ et c₂ et profondeur(LCA(c₁, c₂)) est le nombre d'arcs séparant LCA(c₁, c₂) de la racine et profondeur(c₁) est la longueur du chemin séparant c₁ de la racine en passant par LCA(c₁, c₂). On se base pour le calcul sur l'hierarchie de Wordnet.

- **Calcul de similarité sémantique (dite aussi globale):**

La similarité sémantique sera calculée par la combinaison des similarités terminologique et structurelle. Nous avons proposé à titre de combinaison, la formule suivante :

$$\begin{aligned} \text{Simsem}(c1, c2) \\ = \frac{\sum_{(rc1, rc2) \in VR(c1, c2)} \text{simter}(rc1, rc2) * \text{simwp}(c1, c2) * (1 - |d1 - d2|^2)}{|VR(c1, c2)|} \end{aligned}$$

Où :

rc_i : c'est le voisinage relatif (père ou fils) du concept c_i

$$VR(c1, c2) = \{(rc1, rc2) | rc1 \in V(c1) \text{ et } rc2 \in V(c2) \text{ et } \text{Sim}(rc1, rc2) \geq \text{seuil}\}$$

Tels qu'il faut prendre les couple rc₁ et rc₂ (père-père, fils-fils).

$$d_i = \text{Sim}_{wp}(c_i, rc_i)$$

⁴ Formule pour le calcul de similarité structurelle proposée par Wu et Palmer en 1994 et qui se base sur l'hierarchie des mots contenus dans l'Ontologie Wordnet

Si un concept n'a pas un voisinage relatif qui vérifie la condition précédente, nous proposons la formule suivante pour le calcul de la similarité sémantique:

$$sim_{sem}(c1,c2)= \begin{cases} \frac{(simTer(c1,c2) * coeffTer) + (simStr * coeffStr)}{coeffTer+coeffStr} & \text{si } simTer \geq \text{seuil et} \\ & simStr \neq 0 \\ simTer & \text{si } simTer \geq \text{seuil et} \\ & simStr = 0 \\ 0 & \text{sinon} \end{cases}$$

coeffTer et coeffStr se calculent de la même manière que coeffLex et coeffSyn vus précédemment.

A l'issue de ces étapes on obtiendra une matrice des mesures de similarité « matrice de correspondance ».

Module 5 : « Modification de thésaurus et construction de l'ontologie fusionnée »

On parcourt la matrice de correspondances à la recherche des concepts qui ont un degré de similarité élevé fixé à un seuil ≥ 0.70 . Pour chaque couple de concepts qui vérifie cette condition cela impliquera une fusion des deux concepts.

Pour fusionner les deux concepts, il faut d'abord tester si les deux concepts ont le même terme. Dans ce cas il sera conservé, dans le cas contraire, on remplace les deux concepts par leur premier synSets commun et remplace toutes les occurrences des concepts initiaux par le nouveau terme.

Des fois des conflits d'hierarchie peuvent apparaître. Dans le cas où les deux concepts détecté comme étant aptes à être fusionné possèdent deux pères différents – l'héritage multiple étant interdit dans le formalisme OWL- . Dans ce cas, il faut identifier parmi les deux pères celui qui est père du deuxième et du coup la fusion des deux concepts induira à un changement nécessaire de l'hierarchie.

Une fois les deux concepts fusionnés, on doit regrouper les propriétés (attributs et relations), leurs restrictions ainsi que les instances pour les attribuer au nouveau concept. Après regroupement, on procède par l'élimination des doublons.

Le résultat de cette étape est le thésaurus qui va nous aidé à connaître la nouvelle hiérarchie. Afin de construire l'Ontologie fusionnée. Enfin, le système donne la main à l'utilisateur de sauvegarder cette dernière.

4.2. Diagramme de Paquetages

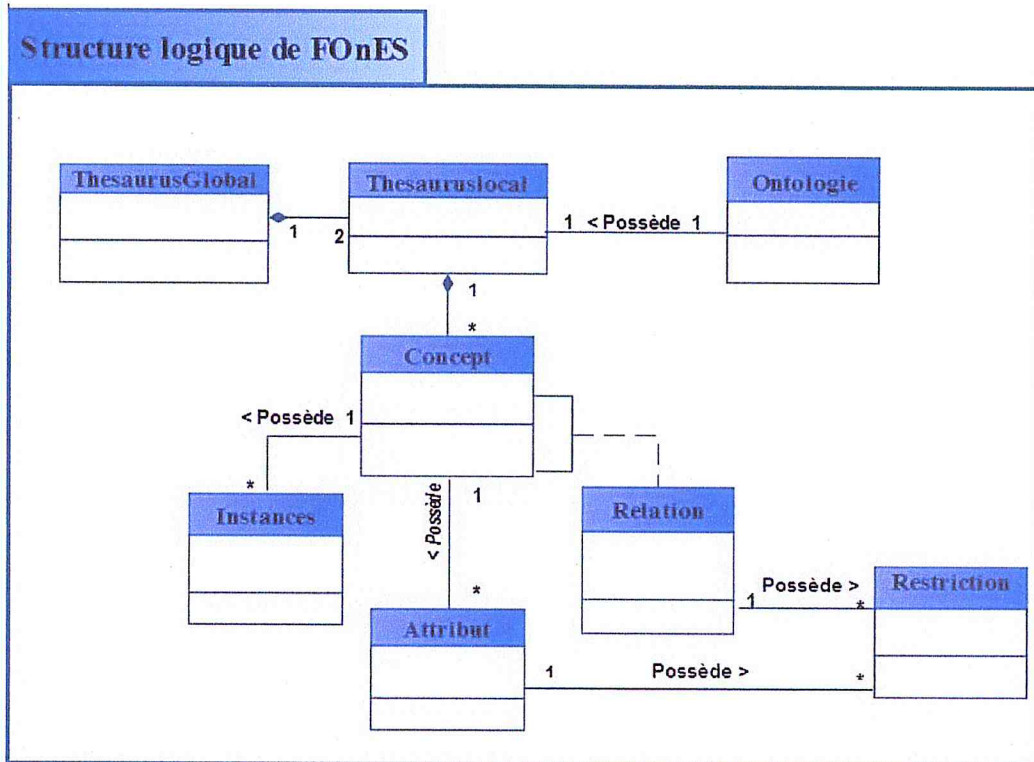


Figure 19: Paquage "Structure logique de FOnES".

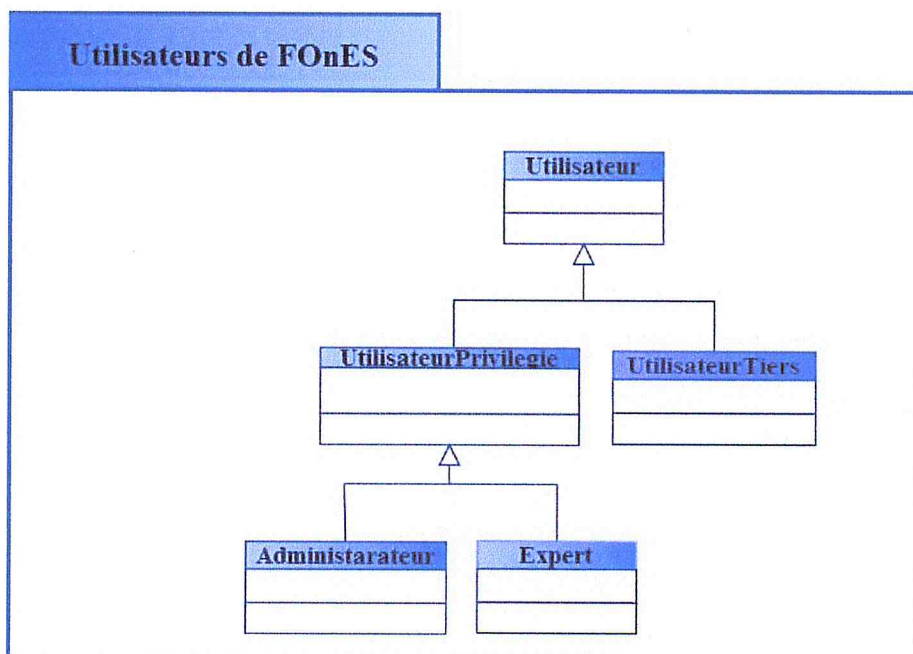


Figure 20: Paquage "Atilisateurs de FOnES".

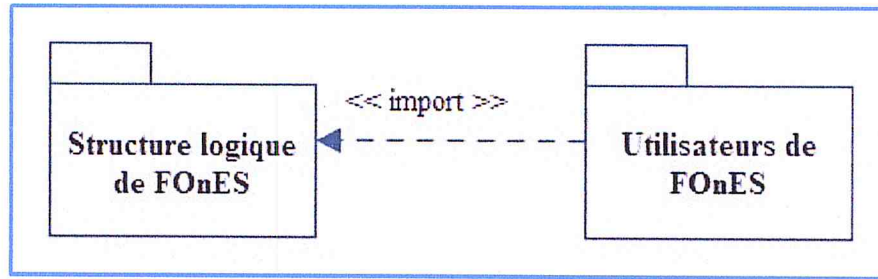


Figure 21: Diagramme de paquetage global.

4.3. Diagramme de classes

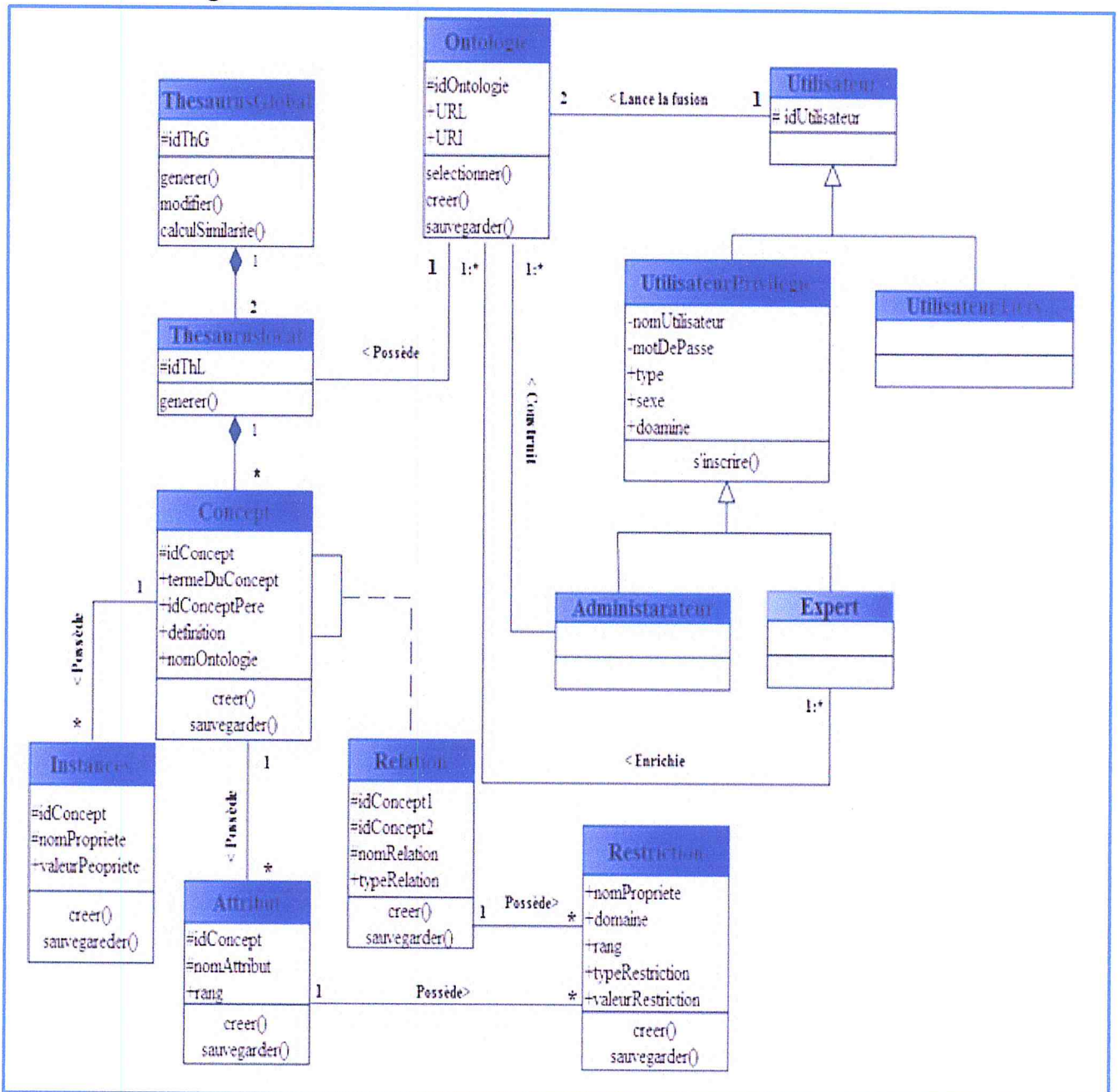


Figure 22: Diagramme de classes du système.

1.1.1. Description du diagramme de classes

Nom Classe	Rôle	Attributs
Utilisateur	C'est la classe mère des utilisateurs privilégiés et utilisateurs tiers. Elle regroupe donc tous les types d'utilisateurs.	<ul style="list-style-type: none"> • idUtilisateur
UtilisateurPrivilege	C'est la classe mère de l'Administrateur et des experts. Elle regroupe donc, tous les utilisateurs qui ont des privileges	<ul style="list-style-type: none"> • idUtilisateur • nomUtilisateur • motDePasse • type
Administrateur	Cette classe regroupe les propriétés et fonction de l'administrateur	<ul style="list-style-type: none"> • idUtilisateur • nomUtilisateur • motDePasse • type
Expert	Cette classe regroupe les caractéristiques et fonctions de l'expert.	<ul style="list-style-type: none"> • idUtilisateur • nomUtilisateur • motDePasse • type
UtilisateurTiers	Cette classe regroupe tous les autres utilisateurs du système (chercheurs, étudiants ... etc)	<ul style="list-style-type: none"> • idUtilisateur
Ontologie	Cette classe est une représentation générique des propriétés et fonctions appliquées sur une Ontologie.	<ul style="list-style-type: none"> • idOntologie • URL • URI
ThesaurusGlobal	Cette classe représente les propriétés du thésaurus global et les différents traitements qui peuvent être effectués sur ce dernier.	<ul style="list-style-type: none"> • idThG
ThesaurusLocal	Cette classe représente le thésaurus local d'une Ontologie qui contient tous les concepts de cette dernière.	<ul style="list-style-type: none"> • idthL
Concept	Cette classe représente le concept qui est l'unité élémentaire de notre système. C'est sur le concept que vont se faire la majorité des traitements induisant à la	<ul style="list-style-type: none"> • idConcept • termeDuConcept • idConceptPere • definition • nomOntologie

	fusion d'ontologies.	
Relation	Cette classe recense toutes les relations qui peuvent exister entre deux concepts donnés.	<ul style="list-style-type: none"> • idConcept1 • idConcept2 • nomRelation • typeRelation
Attribut	Cette classe recense tous les attributs relatifs à un concept donné.	<ul style="list-style-type: none"> • idConcept • nomAttribut • rang
Instance	Cette classe recense toutes les instances relatives à un concept en précisant pour chaque attribut de concept la valeur qu'il prend	<ul style="list-style-type: none"> • idConcept • nomInstance • nomPropriete • valeurPropriete
Restriction	Cette classe recense toutes les restrictions appliquées aux propriétés.	<ul style="list-style-type: none"> • nomPropriete • domaine • rang • typeRestriction • valeurRestriction

Tableau 9: Description du diagramme de cas d'utilisation.

5. Conclusion

Dans ce chapitre, nous avons présenté les étapes détaillées que nous avons suivi pour la construction de FOnES. L'étude conceptuelle nous a permis de mettre en évidence les étapes nécessaires pour la création de l'ontologie fusionnée par le système FOnES. Cette étude nous a permis aussi de mettre en évidence les différentes classes du système en distinguant : la structure métier (logique) et les acteurs interagissant avec le système. Nous avons par ailleurs, mis le point sur l'architecture globale de FOnES ainsi que ces différents modules, à savoir : module de construction, module d'enrichissement, module de calcul des similarités et enfin le module de fusion. Dans le chapitre suivant, nous allons implémenter et mettre en œuvre ce que nous avons proposé dans l'étude conceptuelle, en d'autres termes, le déploiement et l'implémentation de notre approche.

Chapitre V: Implémentation

1. Introduction

Après avoir effectué la conception de FOnES, nous allons à présent entamer la réalisation de ce dernier. Nous présenterons alors, dans un premier lieu, l'environnement de développement (langages et outils) ensuite, le diagramme d'accessibilité du système, son architecture de déploiement et enfin, nous présenterons quelques captures d'écran.

2. Environnement de développement

2.1. Langages utilisés

2.1.1. Java

Java est un langage de programmation informatique orienté objet créé par James Gosling et Patrick Naughton de Sun Microsystems.

Il permet de créer des logiciels compatibles avec de nombreux systèmes d'exploitation (Windows, Linux, Macintosh, Solaris).

Le langage Java donne aussi la possibilité de développer des programmes pour téléphones portables et assistants personnels. Enfin, ce langage peut-être utilisé sur internet pour des petites applications intégrées aux pages web (applet) ou encore comme langage serveur (JSP) [1].

2.1.2. MYSQL

MYSQL est un Système de gestion de bases de données relationnelles (SGBDR) sous licence GNU très utilisé pour mettre en ligne des bases de données.

Il permet d'entreposer des données de manière structurée (Base, Table, Champs, Enregistrements). Le noyau de ce système permet d'accéder à l'information enregistrée via un langage spécifique le SQL.

2.2. OUTILS

2.2.1. Eclipse

Eclipse IDE est un environnement de développement intégré libre, le terme *Eclipse* désigne également le projet correspondant, lancé par IBM.

Il est extensible, universel et polyvalent, permettant potentiellement de créer des projets de développement mettant en œuvre n'importe quel langage de programmation.

Eclipse IDE est principalement écrit en Java à l'aide de la bibliothèque graphique SWT, d'IBM. [2][3]

2.2.2. Protégé

Protégé est un système graphique pour la création d'ontologies. Il a été créé par le SMI (Stanford Medical Informatics) à l'université Stanford. Il est très populaire dans le domaine du Web Sémantique et au niveau de la recherche en informatique.

Protégé est un éditeur d'Ontologies distribué en open source développé en Java. C'est un éditeur hautement extensible, capable de manipuler des formats très divers, tels que: RDF, RDFS, OWL, etc grâce aux plugins dédiés.

Dans le modèle des connaissances de PROTEGE, les ontologies consistent en une hiérarchie de classes ou de concepts représentés sous forme générique qui ont des attributs, qui peuvent eux-mêmes avoir certaines propriétés. L'édition des listes de ces trois types d'objets se fait par l'intermédiaire de l'interface graphique, sans avoir besoin d'exprimer ce que l'on a à spécifier dans un langage formel : il suffit juste de remplir les différents formulaires correspondant à ce que l'on veut spécifier.

Aujourd'hui, il regroupe une large communauté d'utilisateurs et bénéficie des toutes dernières avancées en matière de recherche ontologique : compatibilité OWL de référence, services inférentiels, gestion de bases de connaissances, visualisation d'ontologies, etc. [4]

2.2.3. LES APIS

- **Jena:** JENA [5] est un ensemble d'outils (une API) permettant de lire et de manipuler des ontologies décrites en OWL et d'y appliquer certains mécanismes d'inférences. Au cours de notre développement, on a utilisé la version Jena2.6 qui était la version la plus avancée. Et qui permettait entre autres : la création et l'extraction de concepts, de propriétés (relations et attributs) sur les concepts ainsi que les restrictions sur les propriétés mais aussi des instances et de leurs insertions dans notre ontologie qui n'est autre qu'un fichier owl. Et ce à travers la classe OntModel.

- **JFreeChart:** JFreeChart est une librairie graphique java distribuée en open source. David Gilbert a développé « le JFreeChart project » en Février 2000. Elle est utilisée pour la génération de graphes, tels que : diagrammes en rectangles et en bâtons, diagrammes en camembère, histogrammes, diagramme de Gantt, polygone de fréquences cumulées, nuages de points.

Les développeurs peuvent également avoir une large palette de choix afin de rendre les graphes meilleurs ergonomiquement parlant. [6]

- **OWL2Prefuse :** OWL2Prefuse est un paquetage java auquel on donne en entrée un fichier OWL et qui fournit en sortie un graphe ou un arbre afin de permettre la visualisation de la taxonomie de l'Ontologie initiale qui sera plus compréhensible au format visuel.

Prefuse est un paquetage java qui permet aux développeurs la visualisation de graphes. Il facilite l'interaction entre l'utilisateur et ces graphes. Prefuse est largement utilisé dans les applications de web sémantique. [7]

- **JTattoo :** JTattoo est une librairie java distribuée en open source qui permet aux développeurs d'enjoliver leurs applications en leur offrant d'excellentes interfaces graphiques et ce à travers des thèmes visuels dits « Look and Feels » qui change du thème standard de Java. [8]

- **JAWS :** Comme son nom l'indique, JAWS (Java API for WordNet Searching) est une librairie Java qui permet de chercher et de trouver à travers la base dictionnaire WordNet (aussi bien la version 2.1 que la version 3.0) des définitions, des synonymes et antonymes ... etc

JAWS a été créée par Brett Spell, adjoint membre du département CSE (computer Science and Engineering) à l'université américaine Southern Methodist. [9]

- **JWS (Java WordNet Smilarity) :** C'est une librairie Java distribué en open source qui permet de mesurer la similarité entre deux chaînes de caractère à l'aide de différentes formules, dans notre application nous avons utilisé la distance de Wu & Palmer qui se basant sur l'hierarchie de WordNet.

3. Le système FOnES

3.1. Diagramme d'accessibilité du système

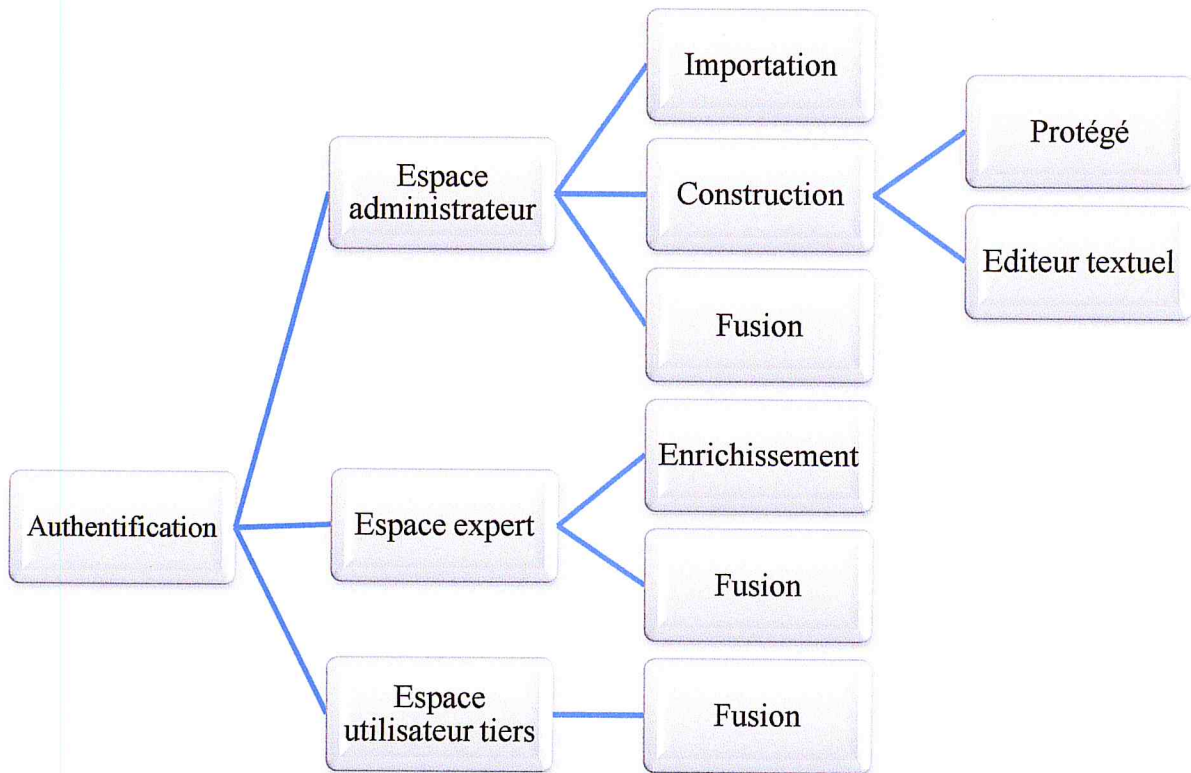


Figure 23: Diagramme d'accessibilité du système FOnES.

3.2. Architecture de déploiement de FOnES

Pour la construction du système FOnES, on a utilisé un type d'architecture dit MVC (Modèle Vue Contrôleur) (Figure 24). Une architecture MVC [10] propose le découpage d'une application en architecture 3-tiers selon le schéma suivant :

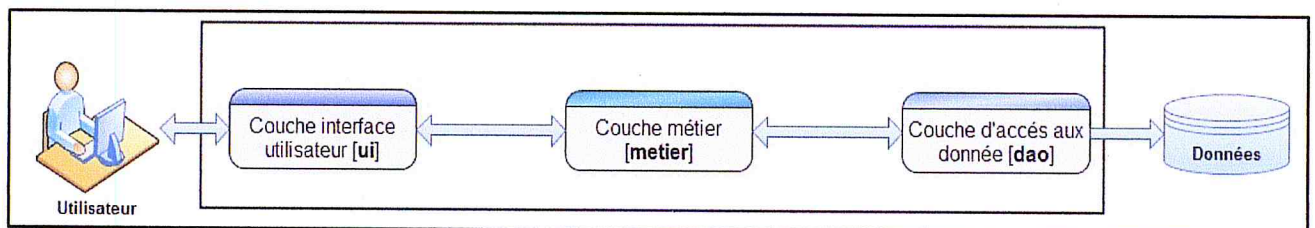


Figure 24: Architecture MVC (Modele Controle Vue).

- La couche [dao] s'occupe de l'accès aux données, le plus souvent des données persistantes au sein d'un SGBD. Mais cela peut être aussi des données qui proviennent de capteurs, du réseau, ... etc

- La couche [métier] implémente les algorithmes " métier " de l'application. Cette couche est indépendante de toute forme d'interface avec l'utilisateur. Ainsi elle doit être utilisable aussi bien avec une interface console, une interface web ou une interface de client riche. Elle doit ainsi pouvoir être testée en-dehors de l'interface graphique et notamment avec une interface console. C'est généralement la couche la plus stable de l'architecture. Elle ne change pas si on change l'interface utilisateur ou la façon d'accéder aux données nécessaires au fonctionnement de l'application.

- La couche [interface utilisateur] qui est l'interface (graphique souvent) qui permet à l'utilisateur de piloter l'application et d'en recevoir des informations.

La communication va de la gauche vers la droite :

- L'utilisateur fait une demande à la couche [interface utilisateur]
- Cette demande est mise en forme par la couche [interface utilisateur] et transmise à la couche [métier]
- Si pour traiter cette demande, la couche [métier] a besoin des données, elle les demande à la couche [dao]
- Chaque couche interrogée rend sa réponse à la couche de gauche jusqu'à la réponse finale à l'utilisateur.

Les couches [métier] et [dao] sont normalement utilisées via des interfaces Java. Ainsi la couche [métier] ne connaît de la couche [dao] que son ou ses interfaces et ne connaît pas les classes les implémentant. C'est ce qui assure l'indépendance des couches entre-elles : changer l'implémentation de la couche [dao] n'a aucune incidence sur la couche [métier] tant qu'on ne touche pas à la définition de l'interface de la couche [dao]. Il en est de même entre les couches [interface utilisateur] et [métier].

3.3. Modèle de déploiement

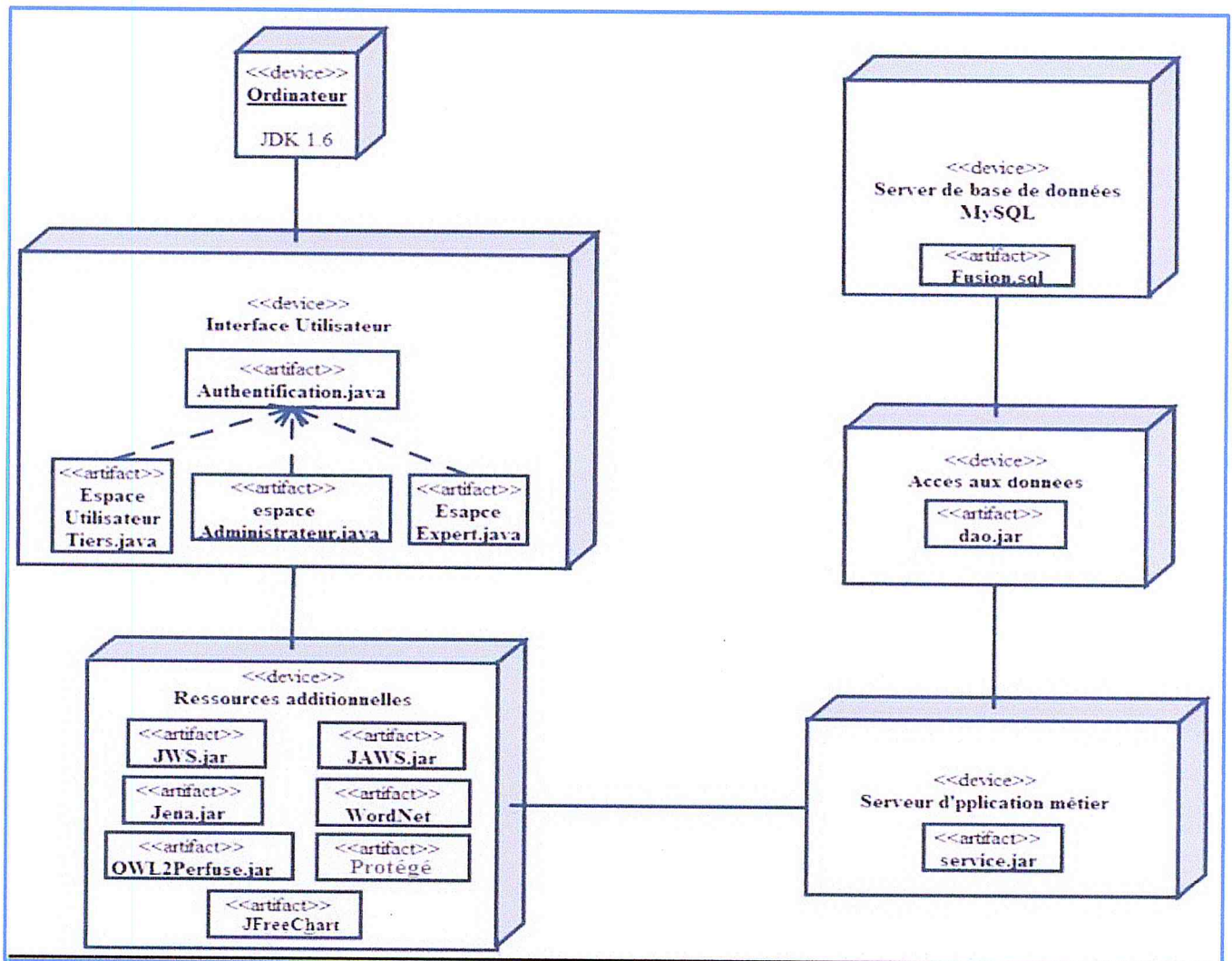


Figure 25: Diagramme de déploiement de FOnES.

3.4. Présentation de l'application

3.4.1. Authentification et Inscription

Dès qu'on lance FOnES, une boîte d'authentification (Figure 26) s'ouvre. Cette dernière permet à l'utilisateur de lancer la fenêtre principale de l'application. Si l'utilisateur n'est pas un membre c'est-à-dire ne possède pas un compte cette boîte lui donne la possibilité d'en créer un, en cliquant sur le lien «S'inscrire».



Figure 26: Interface d'authentification.

Un clic sur « S’inscrire » fait apparaître la fenêtre suivante :



Figure 27: Interface de création d'un nouveau compte.

Cette dernière permet à l'utilisateur de saisir ses informations personnelles et de valider l'inscription.

3.4.2. L'interface principale

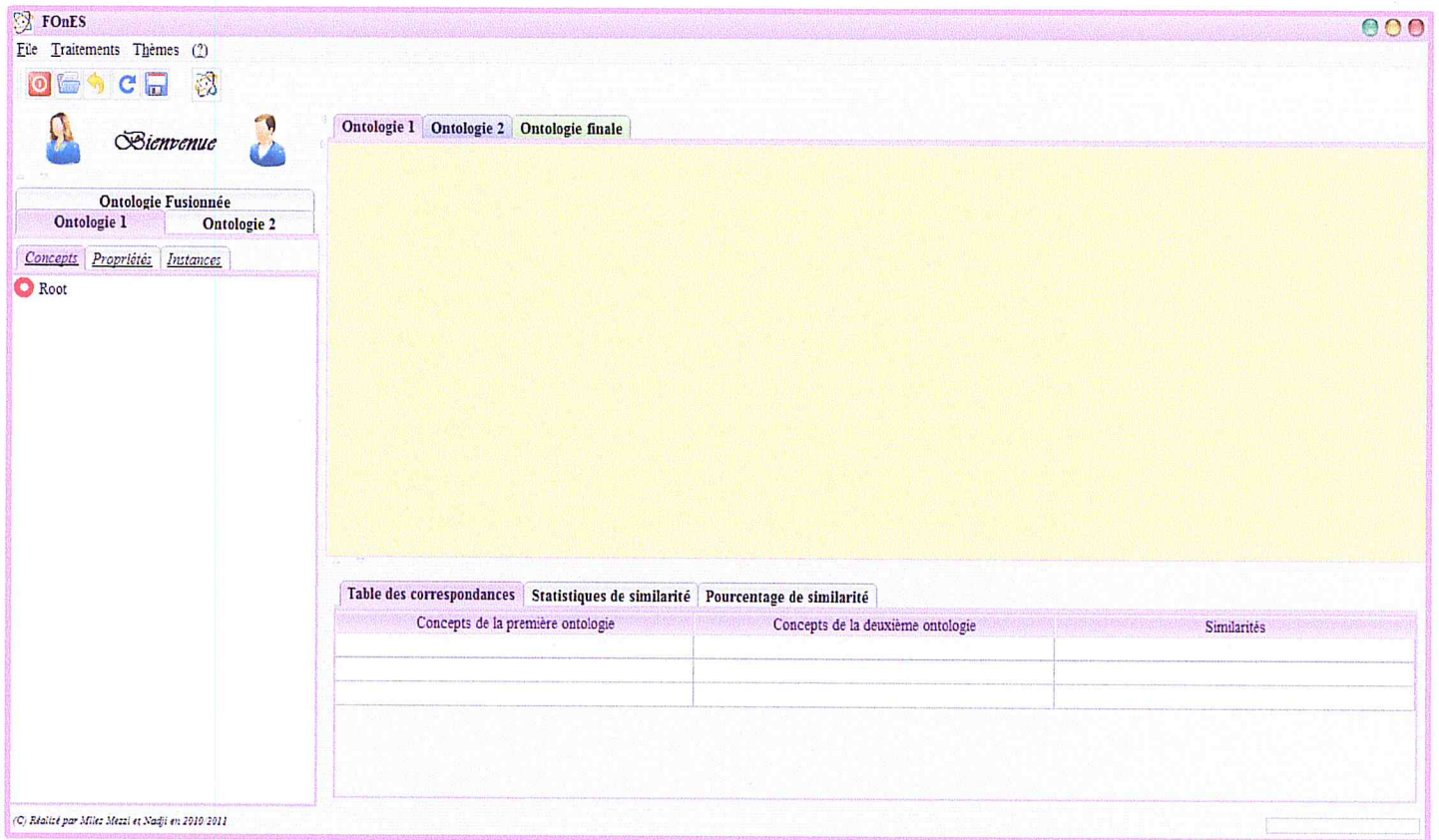


Figure 28: Interface principale de FOnES.

Ceci est l'interface principale de FOnES (Figure 28). Pour le Panel de gauche il sert à développer l'hierarchie de l'Ontologie en présentant :

- Ces concepts :

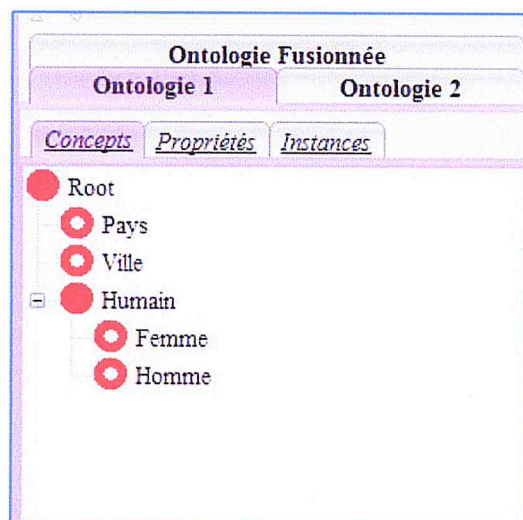


Figure 29: L'arbre des Concepts.

- Ces instances et propriétés



Figure 30: L'arbre des instances.

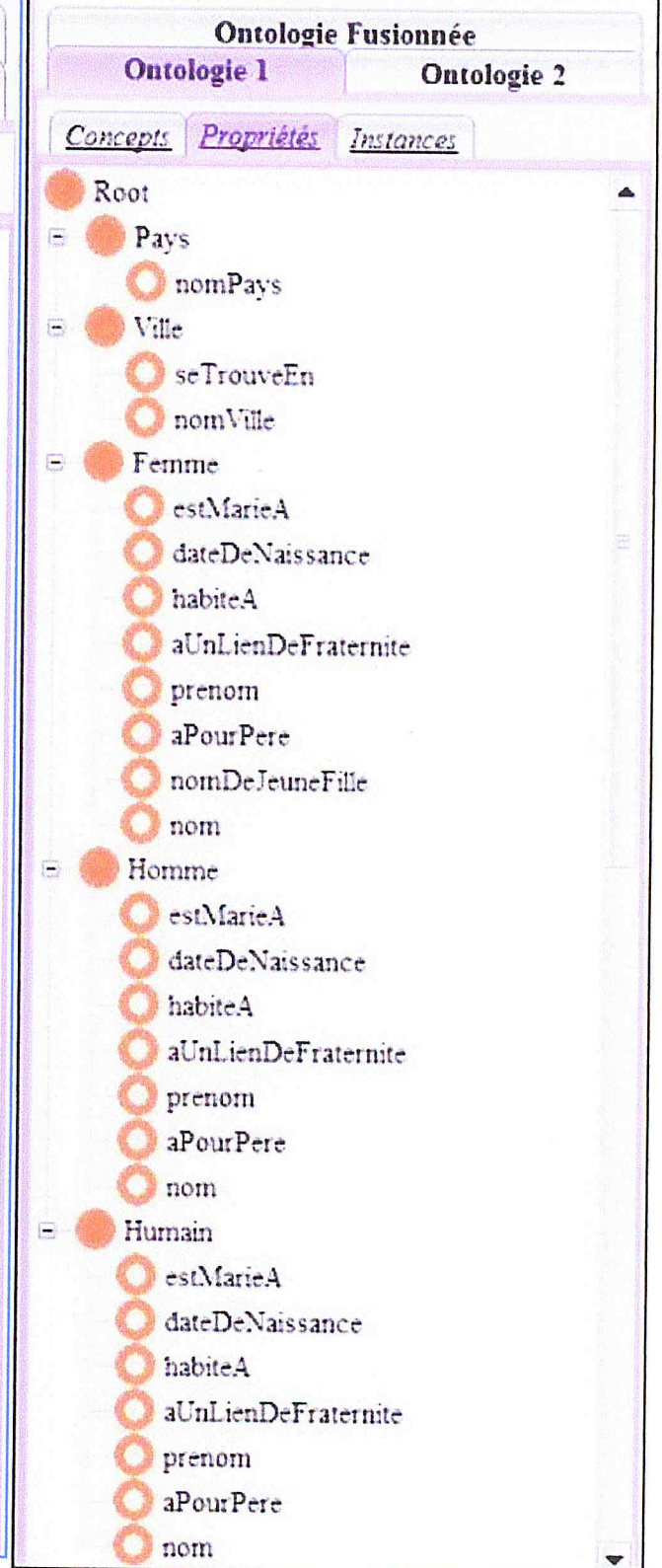


Figure 31: L'arbre des propriétés.

Dans l'interface de FOnES, les fonctionnalités générales diffèrent d'un utilisateur à un autre. Cette différence réside essentiellement dans les traitements. Les captures suivantes montrent les fonctionnalités de chaque acteur.

- Un simple utilisateur n'a le droit que de fusionner

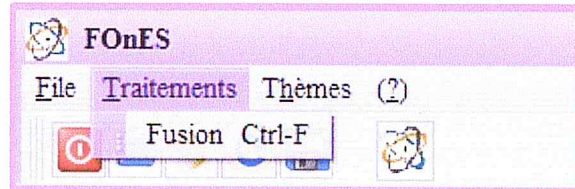


Figure 32: Menu utilisateur tiers.

- Un administrateur a en plus du droit de fusion, le droit d'importer ou de construire des ontologies (via Protégé ou l'éditeur textuel du système).

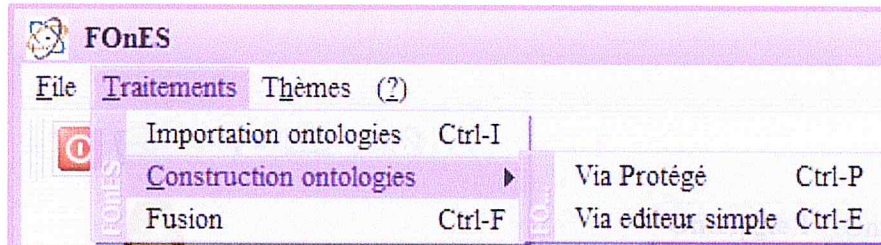


Figure 33: Menu Administrateur.

- Un expert possède le droit de fusion ainsi que celui de l'enrichissement des Ontologies.

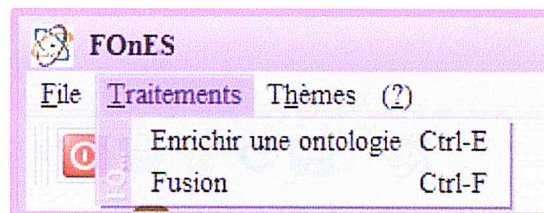


Figure 34: Menu Expert.

3.4.3. Importation

Un clic sur *Importation* du sous menu *Traitements* (administrateur) permet d'afficher la fenêtre d'importation (Figure 35). Cette dernière permet à l'administrateur de visualiser une ontologie (concepts, propriétés, instances plus taxonomie) et puis de la déplacer vers la base d'Ontologies si elle lui convient.

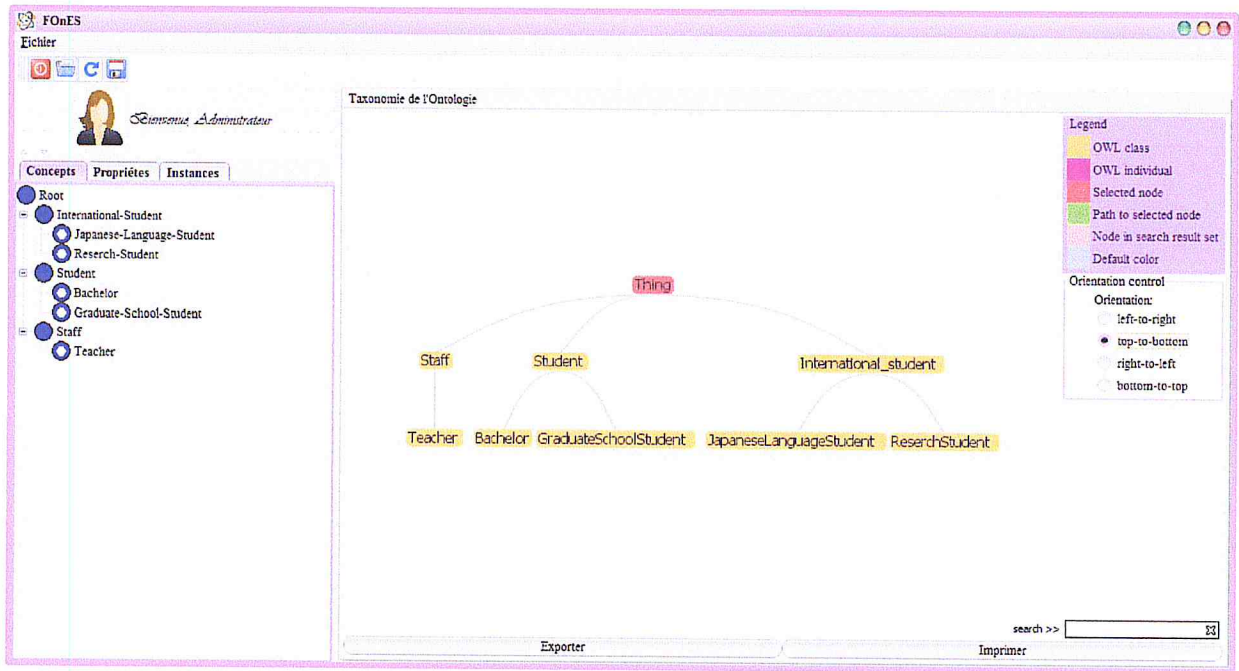


Figure 35: Interface d'importation.

3.4.4. Editeur textuel

Un clic sur *Via éditeur simple* du sous menu *Construction Ontologie* (contenu dans le sous menu *Traitements*) permet d'afficher un éditeur textuel (Figure 36) qui permet à l'administrateur de créer une ontologie en l'écrivant dans une syntaxe OWL .

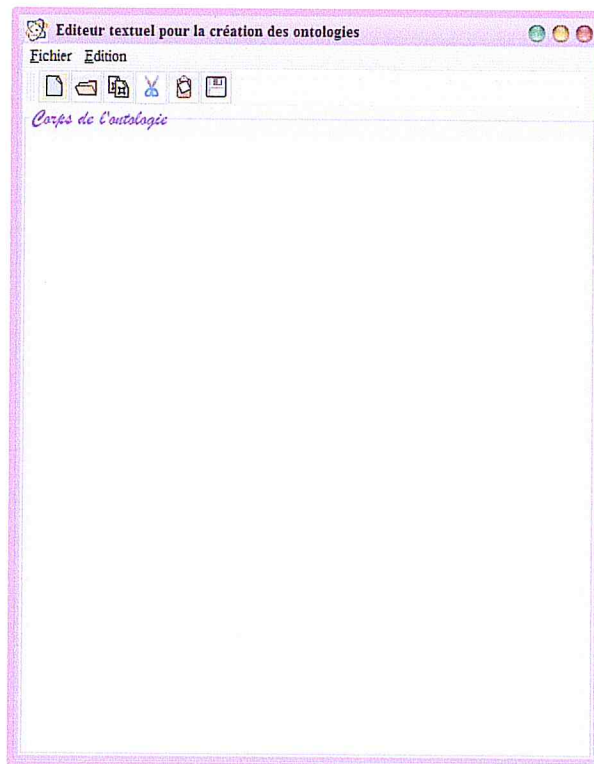


Figure 36: Editeur textuel.

3.4.5. Enrichissement

Un clic sur *Enrichir une Ontologie* du sous menu *Traitements* permet d'afficher la fenêtre d'enrichissement (Figure 37). Cette dernière permet à un expert d'enrichir une ontologie, elle se divise en deux parties (gauche et droite). Lors de la sélection d'un fichier OWL(Ontologie) ses concepts vont être affichés dans la partie gauche de la fenêtre. Lorsque l'expert sélectionne un concept ses définitions s'afficheront dans la partie droite. C'est à lors que l'expert choisi la définition la plus proche au domaine ou propose sa propre définition (en l'écrivant dans définition additionnelle) dans le cas où le système ne lui propose rien.

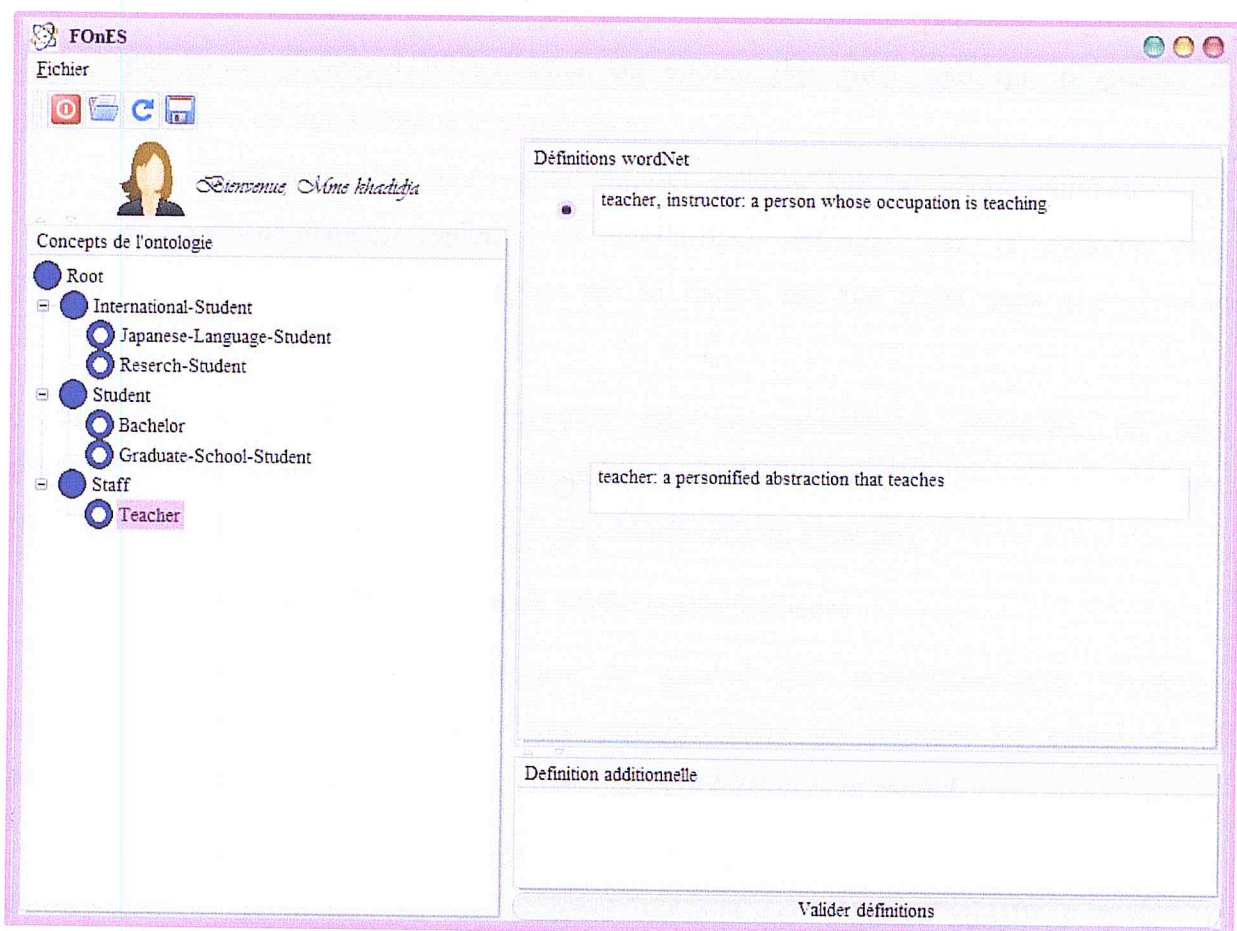


Figure 37: Interface d'enrichissement.

4. Conclusion

Dans ce chapitre nous avons présenté l'implémentation et le déploiement de notre système de fusion FOnES. Cette étape nous a par ailleurs, permis de nous familiariser avec les différentes fonctionnalités qu'offre notre système.

$$FP = S - S \cap H$$

- Les correspondances *correctes omises* par un système sont appelées « the false négatives (FN) » et sont calculées ainsi :

$$FN = H - S \cap H$$

- La *précision* est une mesure d'exactitude, elle varie entre [0,1] elle est calculée de la manière suivante :

$$Precision = \frac{|TP|}{|TP + FP|} = \frac{S \cap H}{S}$$

- Le *rappel* est une mesure de perfection, elle varie entre [0,1], elle est calculée de la manière suivante :

$$Rappel = \frac{|TP|}{|TP + FN|} = \frac{S \cap H}{H}$$

Les valeurs obtenues par le calcul des correspondances entre les Ontologies ne sont comparables par le biais de la précision et du rappel, en fait, *le rappel peut prendre des valeurs importantes aux dépens de la précision*, en retournant toutes les correspondances possibles.

En même temps, *la précision peut prendre des valeurs importantes aux dépens du rappel*, en retournant que les correspondances correctes cependant peu nombreuses.

- C'est pour ces raisons qu'il est préférable de prendre en considération les deux mesures simultanément via une mesure qui combine le rappel et la précision telles que : la *F-mesure* qui se calcule de la manière suivante :

$$F - Mesure = \frac{2 * (Rappel * Précision)}{Rappel + Précision}$$

La *F-mesure* est une mesure globale de la qualité des correspondances produites, elle varie entre [0,1], cette mesure alloue la même importance à la précision et au rappel.

- Une autre mesure de la qualité de l'alignement et qui combine le rappel et la précision : *l'overall* qui se calcule de la manière suivante :

$$OVERALL = Rappel \left(2 - \left(\frac{1}{Précision} \right) \right)$$

L'overall peut prendre des valeurs négatives si le nombre de fausses correspondances (FP) trouvées par le système dépasse le nombre de correspondances correctes (TP) trouvées par le système.

Chapitre VI : Tests et validation

1. Introduction

Conformément à tout ce qui a été dit dans les chapitres précédents, on sait que la construction de l'Ontologie fusionnée par FOnES se fait après l'étape du calcul de similarité et de modification du thésaurus.

Toute fois pour nous assurer de l'optimalité des résultats fournis par FOnES, nous devons valider l'ontologie obtenue. Pour se faire, différentes approches existent :

- La validation automatique de l'Ontologie finale par des logiciels tels que Racer, Pellet, OWLValidator ou encore des sites web qui proposent la validation en ligne comme le wonderWeb.
- Pour notre cas, nous nous sommes appuyés sur la théorie suivante : « Si les correspondances (valeurs de similarités) obtenues par le système sont valides alors l'ontologie obtenue par la fusion est elle aussi dans une certaine mesure valide ».

Pour évaluer le degré de pertinence des correspondances obtenues, on doit procéder à leur évaluation. Et ce, en utilisant quelques mesures comme : précision, rappel, overall et f-mesure (dites mesures de performance du système).

2. Définition des mesures de performance utilisées

Les mesures utilisées pour évaluer la qualité des correspondances produites entre les concepts des ontologies sont principalement les mesures de calcul de la pertinence en recherche d'information, telles que la *précision* et le *rappel*.

Le calcul de ces mesures [GHO, 08], est basé sur la comparaison entre les correspondances produites par un système automatique qu'on appellera S et un ensemble de correspondances de référence produit par un humain qu'on notera H .

- Les correspondances *correctes trouvées* par un système sont appelées « *the true positives (TP)* » et sont calculées ainsi :

$$TP = S \cap H$$

- Les correspondances *incorrectes trouvées* par un système sont appelées « *the false positives (FP)* » et sont calculées ainsi :

Dans la plupart des cas *l'overall* est plus petit que le rappel et la précision, ce qui rend difficile d'atteindre un *overall* supérieur a 0.5.

- Une mesure pour évaluer le *pourcentage d'erreurs* du système automatique est le *Fallout* qui se calcule de la manière suivante :

$$Fallout = \frac{FP}{FP + TP}$$

3. Jeux de Données

Afin de valider notre système, nous avons choisit un jeu de données qui consiste à prendre deux Ontologies (Ontologie1 et Ontologie2 dont les codes se trouvent en annexe), puis à comparer les correspondances obtenues par FOnES et ceux réalisés par un mapping manuel (Tableau 10). Et ce, par le calcul des mesures de performance citées précédemment. Enfin, on présentera la taxonomie de l'Ontologie fusionnée (Figure 42).

Soit à fusionner les deux ontologies suivantes :

- Ontologie1 :

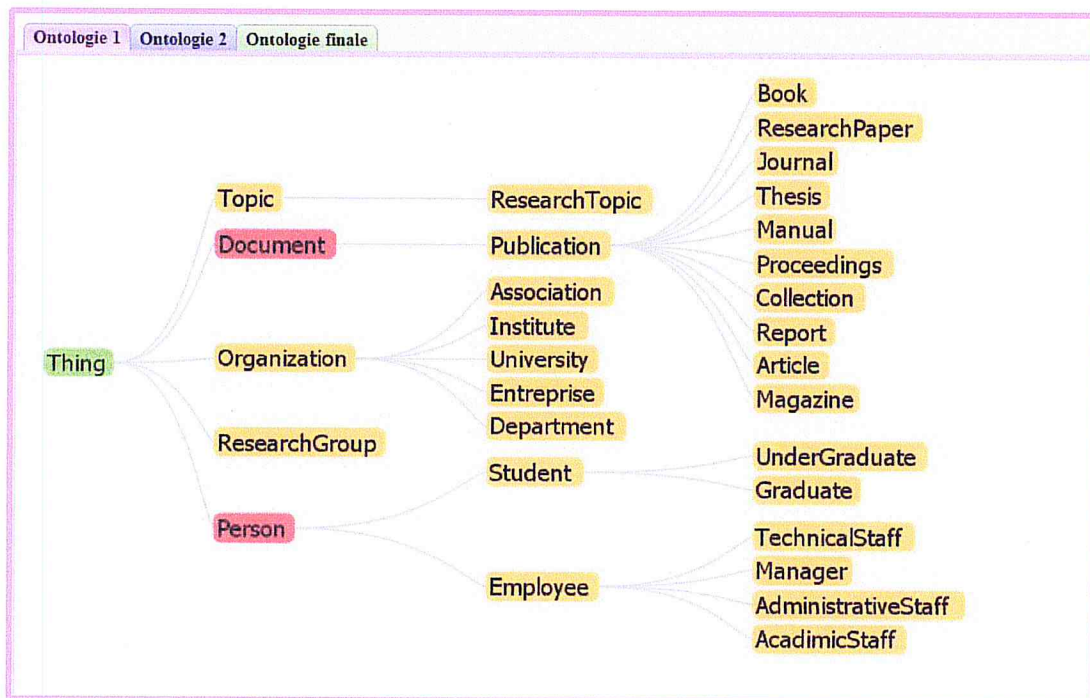


Figure 38: Taxonomie Ontologie1.

- Ontologie2

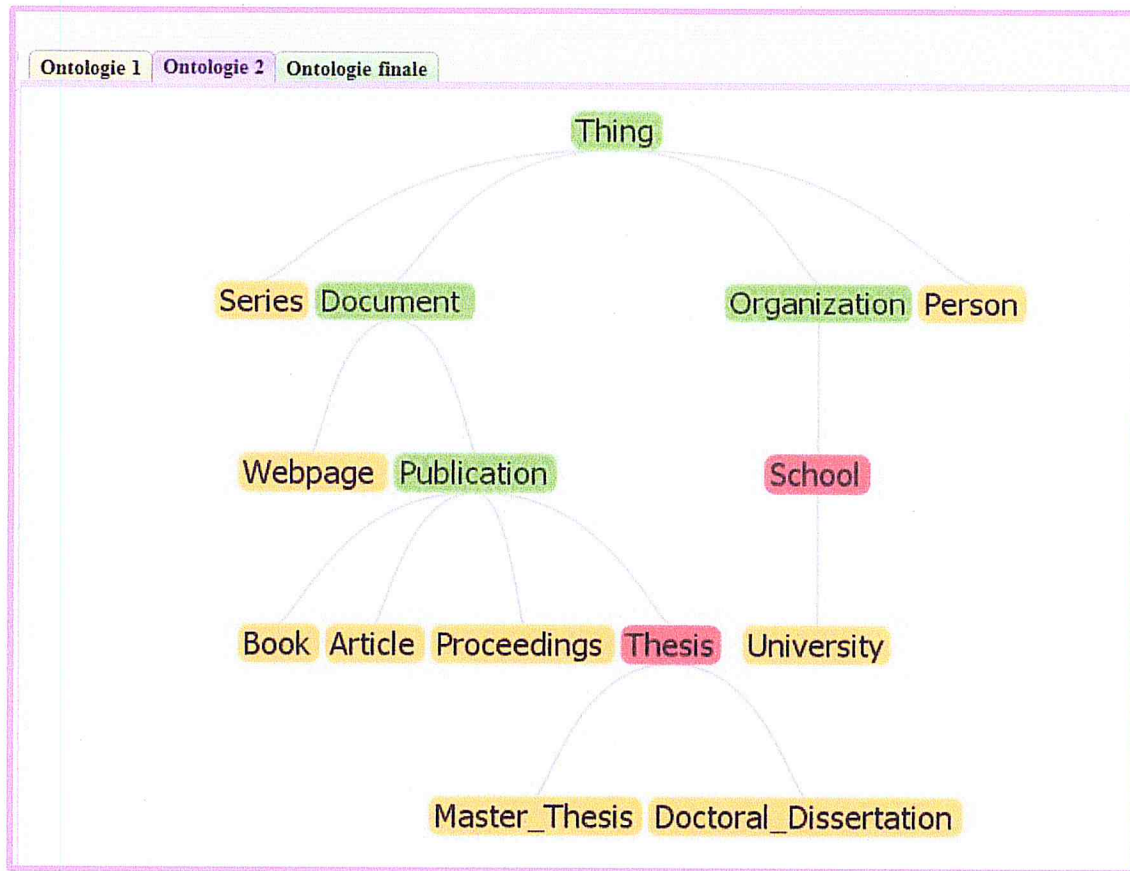


Figure 39: Taxonomie Ontologie2.

Le tableau suivant montre un échantillon des correspondances manuelles et les correspondances de FOnES entre les différents concepts des ontologies initiales (le tableau entier sera présenté en annexes).

Concepts de l'Ontologie1	Concepts de l'Ontologie2	Correspondances manuelles	Mesures de similarité	Correspondances de FOnES
Document	Document	✓	1.0	✓
Article	Proceedings	✗	0.430	✗
Collection	Thesis	✗	0.319	✗
Journal	Book	✗	0.285	✗
Report	Master-Thesis	✗	0.0	✗
Thesis	Thesis	✓	0.97	✓
Doctrotat-Thesis	Doctoral-Dissertation	✓	0.922	✓
Doctrotat-Thesis	Master-Thesis	✗	0.562	✗

Research-Topic	School	x	0.0	x
<p><i>Correspondances produites par le Système : S = 11</i> <i>Correspondances produites manuellement : H = 11</i></p>				

Tableau 10: Correspondances manuelles / FOnES (test et validation).

Dans ce petit échantillon, nous remarquons que l'expert a validé les correspondances (Document-Document, Thesis-Thesis, Doctorat-Thesis et Doctoral Dissertation), FOnES a obtenu des valeurs de similarité $> 0,70$ et donc a aussi considéré que ces concepts là été aptes à êtres fusionnés. Ce qui n'est pas le cas pour les autres concepts (où la mesure de similarité était $< 0,70$).

4. Mesures de performance de FOnES

A travers le tableau 11, nous pouvons constater que mesures de performances de FOnES ont été optimales (du moins par rapport à ce jeux de données).

TP	FP	Précision	Rappel	F-mesure	OVERALL	Fallout
1	1	1	1	1	1	0

Tableau 11: Mesures de performances de FOnES.

FOnES calcule les statistiques (Figure 40) et le degré de similitude (Figure 41) entre les Ontologies initiales. Pour les Ontologies choisies ci-dessus, les résultats qu'il adonné sont les suivants :

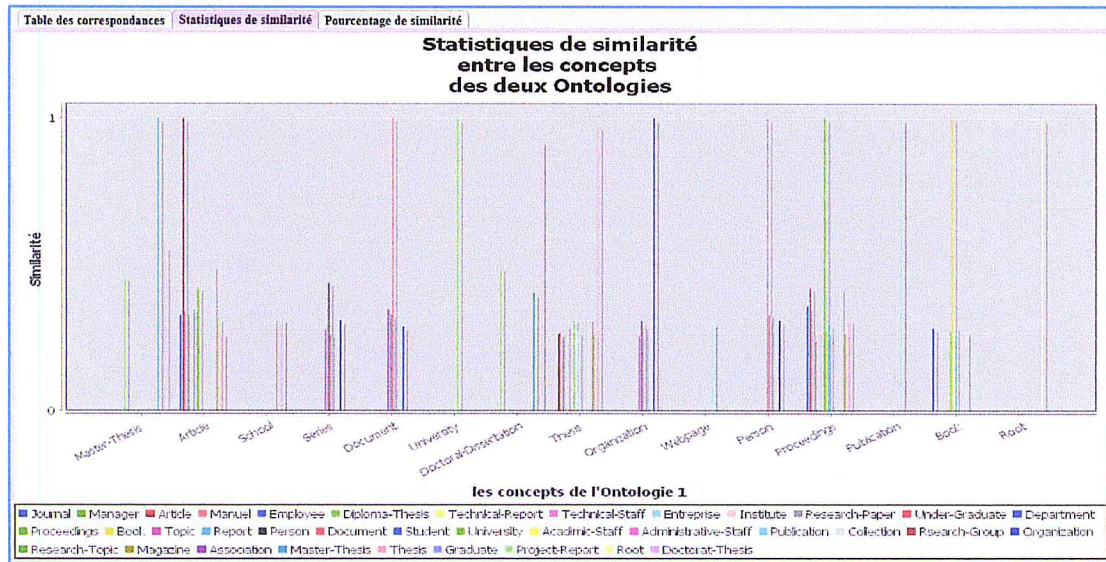


Figure 40: Statistiques de similarité entre Ontologie1 et Ontologie2.

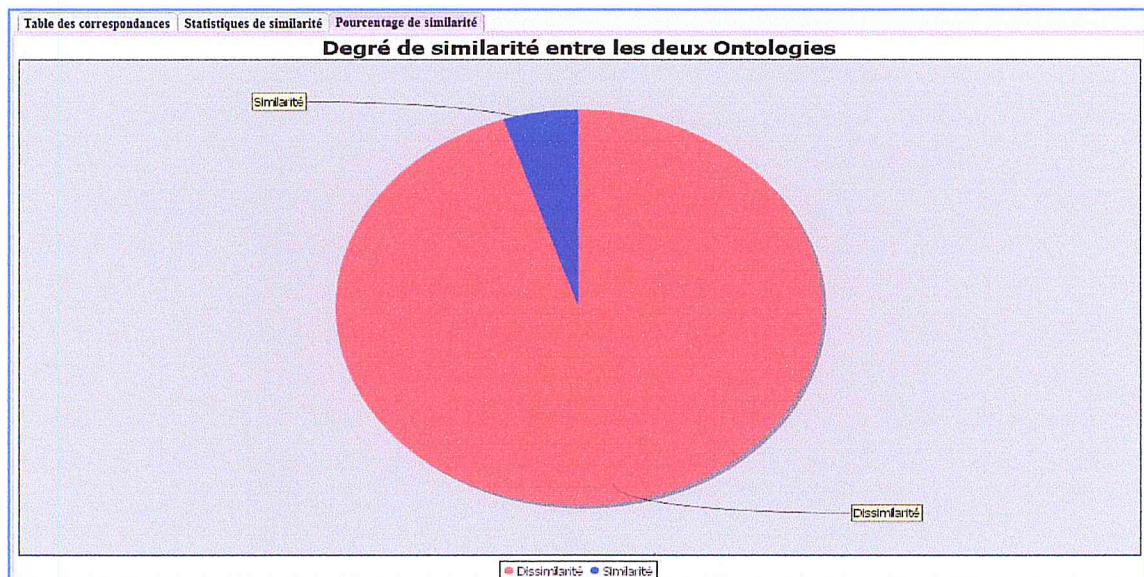


Figure 41: Degré de similarité entre Ontologie1 et Ontologie2.

Après le calcul des mesures de similarité, FOnES procède par la création de la nouvelle Ontologie dont la taxonomie est la suivante :

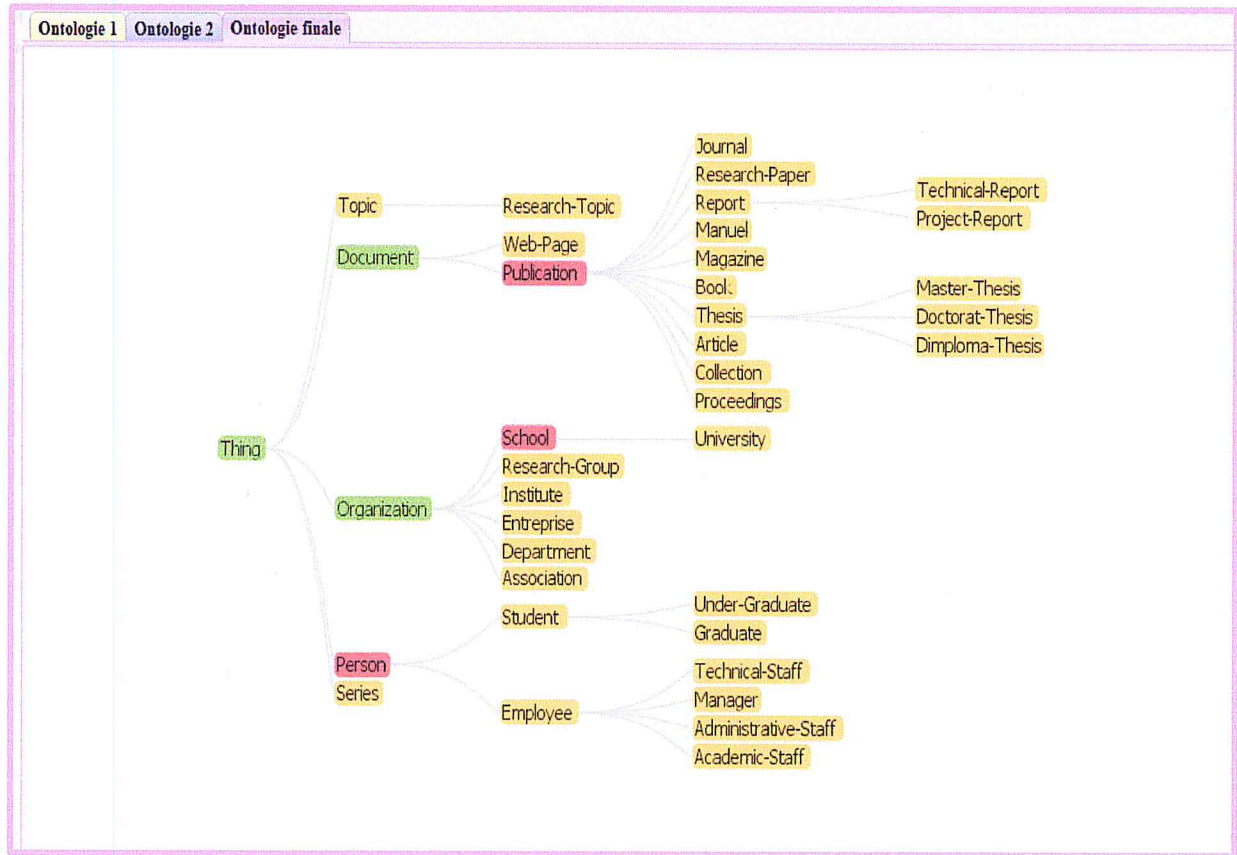


Figure 42: Taxonomie de l'Ontologie fusionnée.

Conclusion générale

Conclusion générale

Conclusion générale

Conçue comme réponse aux problèmes posés par les difficultés qu'incombent la construction d'Ontologies à partir de zéro. La fusion d'Ontologie apparaît aujourd'hui comme une clé pour diminuer les efforts dus la manipulation automatique de l'information au niveau sémantique et de la gestion des connaissances. La puissance des applications qui utilisent des ontologies laisse à penser que leur place au sein des systèmes de recherche d'information ne peut que croître. Cette puissance vient du fait que les systèmes qui utilisent les ontologies font des traitements sémantiques sur de grandes quantités d'informations, réduisant ainsi le taux de bruit et de silence des réponses obtenues.

Les travaux menés dans ce mémoire nous ont permis d'approfondir nos connaissances dans le domaine de l'ingénierie des connaissances (et plus particulièrement l'ingénierie Ontologique). Notre objectif a été de tirer profit des travaux menés dans cette voie et nous nous sommes intéressées plus particulièrement aux méthodes de calcul de similarité entre deux Ontologies, dans le but de construire une nouvelle Ontologie par leur fusion.

Nos recherches et constatations, nous ont conduit à la mise au point d'une méthode originale de fusion d'Ontologie nommée FOnES (Fusion d'Ontologie par Enrichissement Sémantique). L'originalité de notre méthode réside dans la combinaison des caractéristiques suivantes :

- Automatisation totale du processus de fusion, l'utilisateur n'aura qu'à sélectionner les ontologies qu'il veut fusionner et le système, le décharge entièrement des tâches superflues comme attribution de nom ou règlement des conflits dus à la fusion.
- Intégration: le système s'intègre facilement dans n'importe quel environnement car il porte en son sein toutes les API essentielles, logiciels et ressources externes nécessaires à son bon fonctionnement.
- Souplesse dans le travail, l'utilisateur tiers sera soulagé de ne pas avoir à rechercher les Ontologies à fusionner vue que cette tâche sera endossé par l'administrateur du système qui aura pour rôle de créer ou d'importer des

Conclusion générale

Ontologies qui vont servir pour d'éventuelles fusions et d'en alimenter la base d'Ontologies. Toutes ces tâches sont transparentes à l'utilisateur tiers.

- Evolutivité, de par sa construction en architecture MVC, FOnES pourra être mis à jour et ne pas se trouver dépassé face à l'avancée de la recherche dans le domaine de fusion. En effet, FOnES peut être muni et fourni de nouvelles fonctionnalités ce qui n'aura pas d'incidence sur celles qu'il possède déjà.
- Interactivité homme-machine qui constitue l'une des quêtes les plus convoitées, est respectée et mise en avant dans FOnES. ET surtout en ce qui concerne l'importation, la création et l'enrichissement des Ontologies.
- Dynamique : Le fait que résultat final de FOnES soit une Ontologie au même titre que les Ontologies utilisées pour la fusion. Rend notre système réutilisable à l'infini vue que les nouvelles Ontologies peuvent à leur tour servir à la création de nouvelles Ontologies par fusion.
- Ergonomie : Son interface très malléable, rend FOnES très agréable d'utilisation. Lui offrant la possibilité de charger une Ontologie, de visionner ces concepts ainsi que leurs propriétés et instances mais aussi de survoler toute la taxonomie de l'Ontologie avec possibilité d'impression ou de sauvegarde de cette dernière sous forme d'image.
- Extensibilité, Son caractère extensible lui confère un côté relativement générique de par sa faible dépendance vis-à-vis du domaine auquel se rapportent les Ontologies initiales du moment que c'est le même pour les deux.
- Finesse des résultats, La mesure de similarité qu'adopte FOnES étant composite (implique à la fois : similarités terminologiques aussi bien lexicales que syntaxiques, similarités structurelles et bien sûr sémantiques), elle lui confère un haut niveau de pertinence dans le calcul des correspondances entre les Ontologies initiales. Ce qui permet de produire une fusion riche et très fidèle (sémantiquement) au domaine auquel elle doit se rapporter.

En résumé, en dira que les objectifs initiaux ont été totalement satisfaits.

Conclusion générale

Perspectives

Durant les quelques mois consacrés à la réalisation de notre projet de fin d'études nous nous sommes obstinées à atteindre les objectifs qui nous étaient fixés au départ en y ajoutant d'autres idées.

Toute fois, dans le souci d'améliorer FOnES et d'étoffer le spectre de ses fonctionnalités, nous proposons les perspectives suivantes :

- L'ajout de la similarité extensionnelle dans notre mesure de similarité composite. Contrairement aux approches de correspondances qui se basent sur les termes des entités ou sur la structure de l'ontologie, les mesures extensionnelles se basent sur le calcul de ressemblance entre les instances d'Ontologies et donc se fonde sur des informations réelles et concrètes
- Utilisation de la technologie SMA (Système Multi Agents) qui émane du contexte suivant : « l'intelligence d'un individu ne se forge que lorsqu'il est en contact avec d'autres individus de son espèce, cet individu que l'on appelle *agent* a donc besoin d'interagir, de collaborer, de rentrer en conflit, de s'adapter et de communiquer avec son environnement pour pouvoir accomplir des buts beaucoup plus complexes. ». Relevant du domaine de l'intelligence artificielle, Les SMA, du fait de leur nature distribuée, facilitent énormément l'adaptabilité du système à toute évolution (aussi bien positive que négative), comme : changement des systèmes opératoires, changement de la configuration du système, mise à jour de l'ontologie et de l'entrepôt de ressources, ajout et suppression des fonctionnalités, etc.
- L'instauration de la notion de workflow entre les différents acteurs du système. En leur offrant éventuellement des outils de communication et de collaboration (e-mail, visioconférences) avec les quels, les utilisateurs tiers ou les experts pourront demander explicitement à l'administrateur de leur importer ou construire telle ou telle Ontologie et les experts pourront s'échanger des points de vues sur l'enrichissement afin qu'il soit le plus pertinent que possible. Les différents acteurs pourront remplir des formulaires (feed-back) qui témoigneraient de leur satisfaction ou mécontentement du logiciel ou faire des suggestions d'amélioration.

- On aurait par ailleurs, voulu tester FOnES dans un environnement d'application concret avec de grandes Ontologies (contenant plus de cent concepts) afin de lui rajouter des raffinements dans les calculs. Ou dans un environnement Web (dans le domaine de la recherche d'information) afin de constater l'apport de l'ontologie construite par FOnES à ce domaine.

Nous retiendrons les paroles de Mr Aaron Lynch qui a dit :

"L'important n'est pas comment un homme acquiert des idées, mais comment une idée acquiert des hommes"



Annexes

Annexe A : Le langage JAVA

1. Présentation de JAVA

Java est un langage de programmation compilé et interprété, orienté objet. Il a été créé en 1991 par Sun pour pallier aux contraintes que posait le C++, et cela dans le but de développer des logiciels pour l'électronique grand public (appareils ménagers). La syntaxe de ce langage est assez proche du C et est plus claire que le C++. L'objectif pour lequel JAVA a été conçu nécessite certaines caractéristiques comme : La robustesse, la compatibilité, la facilité de programmation et la petite taille du runtime ou des codes générés. JAVA présente beaucoup d'avantages grâce auxquels il a mérité son succès:

- Orienté objet : la brique de base du programme est donc l'objet, instance d'une classe.
- Indépendant des architectures.
- Portable : une fois le programme est compilé, il pourra fonctionner aussi bien sous des stations Unix, que sous Windows ou autre.
- Interprété : par la machine virtuelle de JAVA (JVM).
- Gère la mémoire automatiquement.
- Possède une API très riche : différents packages permettent d'accéder au réseau, aux entrées/sorties et aux différents composants graphiques.
- Distribué, simple, robuste, sécurisé, multithread et dynamique.

Sa portabilité et son indépendance des architectures ou plates-formes constituent son principal atout et grâce auquel il peut fonctionner sur différentes plates-formes et sous différents systèmes d'exploitation. Ces deux caractéristiques sont dues à la machine virtuelle Java (JVM). A côté de ses nombreux avantages, JAVA présente un seul inconvénient : la lenteur lors de la conversion des instructions de la JVM en instructions compréhensibles par la machine. Cependant deux solutions sont mises en place pour éviter cet inconvénient :

- Compiler le code pour une machine spécifique, à ce moment là le programme n'est plus portable
- Doter la JVM d'un JIT (Just In Time compiler) qui traduit le code JVM d'une classe dès sa première utilisation en code spécifique à la machine.

2. JAVA Virtual Machine (JVM) :

La machine virtuelle appelée aussi interpréteur, joue le rôle d'un traducteur. Elle traduit en ByteCode le code compilé (ensemble de fichiers de classes) qui est sous forme de pseudo-code et non en code machine, et cela afin d'être exécuté sur n'importe quelle plate-forme matérielle et logicielle : que l'on soit sur un Pentium, un PowerPC, un Sparc ou sur un Alpha, sous Windows, MacOS, Solaris ou Linux, etc. Cependant, la présence de la JVM au niveau de la plate forme est nécessaire à l'exécution des programmes JAVA sur une plate-forme quelconque.

Les JVMs sont fournies soit par le JDK (Java Development Kit), soit par les navigateurs ou bien par les environnements de développement spécifiques tel Borland JBuilder. Des JVM capables d'exécuter du code Java, peuvent être fournis également par certaines solutions telle Java Plug-in de Sun. La JVM peut être obtenue à partir du site de Sun Microsystems.

3. Environnement de développement

Il existe plusieurs environnements de développement pour JAVA tel le JDK (Java Development Kit) qui est gratuit, mais la plus part sont payants. Ces environnements mettent à la disposition des développeurs : un éditeur intégré, un compilateur, un débogueur sophistiqué et de nombreux générateurs de code (surtout concernant les interfaces graphiques).

3.1. Le Java Development Kit (JDK)

Le JDK est un environnement gratuit comportant un compilateur et une machine virtuelle (minimal et suffisant). Il est téléchargeable à partir du site de Sun. Il comporte l'ensemble des éléments qui ont pour but le développement, la mise au point et l'exécution des programmes Java. Il peut être considéré comme un ensemble d'outils plus un jeu de classes et de service plus un ensemble de spécifications. Il existe plusieurs versions de JDK, il est important de connaître la version employée car les classes disponibles peuvent être différentes d'une version à une autre.

4. Application Programming Interface (API)

Une API fournit aux programmeurs les outils de base nécessaires afin de leur faciliter le travail. Elle se compose d'un ensemble de fonctions, routines et méthodes. Par l'API de JAVA le programmeur va pouvoir accéder à toutes les

ressources de la machine, et celles du réseau Internet. Elle est divisée en packages. Plusieurs packages sont fournis en standard avec JAVA tels :

- java.awt et javax.swing pour les interfaces graphiques,
- java.applet pour la réalisation des applications qui peuvent s'exécuter dans un navigateur Internet.
- java.sql pour accéder aux bases de données.
- java.beans pour la programmation orientée composants (les Java Beans sont des composants logiciels réutilisable tel NetBeans ou JBuilder).
- java.rmi pour la mise en place des applications réparties telles les applications client/serveur.
- D'autres packages se trouvent dans d'autres applications et il faudra les importer tels : Les API PROTEGE et JENA pour la manipulation des ontologies.

5. Eclipse

Eclipse est un environnement de développement intégré (Integrated Development Environment), développé par I.B.M, dont le but est de fournir une plateforme modulaire pour permettre de réaliser des développements informatiques. Eclipse utilise énormément le concept de modules nommés "plugins" dans son architecture. Hormis le noyau de la plateforme nommé "Runtime", tout le reste de la plateforme est développé sous la forme de plugins. Ce concept permet de fournir un mécanisme pour l'extension de la plate-forme et ainsi fournir la possibilité à des tiers de développer des fonctionnalités qui ne sont pas fournies en standard par Eclipse. Les principaux modules fournis en standard avec Eclipse concernent Java mais des modules sont en cours de développement pour d'autres langages notamment C++, Cobol, mais aussi pour d'autres aspects du développement (base de données, conception avec UML, ...). Ils sont tous développés en Java soit par le projet Eclipse soit par des tiers commerciaux ou en open source. Les modules agissent sur des fichiers qui sont inclus dans l'espace de travail (Workspace). L'espace de travail regroupe les projets qui contiennent une arborescence de fichiers. Bien que développé en Java, les performances à l'exécution d'Eclipse sont très bonnes car il n'utilise pas Swing pour l'interface homme-machine mais un toolkit particulier nommé SWT associé à la bibliothèque JFace. SWT (Standard Widget Toolkit) est développé en Java par IBM en utilisant au maximum les composants natifs fournis

par le système d'exploitation sous jacent. JFace utilise SWT et propose une API pour faciliter le développement d'interfaces graphiques.

Les points forts d'Eclipse

Eclipse possède de nombreux points forts qui sont à l'origine de son énorme succès dont les principaux sont :

- Support de plusieurs plateformes d'exécution : Windows, Linux, Mac OS X, ..
- Une plate-forme ouverte pour le développement d'applications et extensible grâce à un mécanisme de plug-ins.
- Plusieurs versions d'un même plugin peuvent cohabiter sur une même plateforme.
- Un support multi langage grâce à des plug-ins dédiés : Cobol, C, PHP, C#, ...
- Malgré son écriture en Java, Eclipse est très rapide à l'exécution grâce à l'utilisation de la bibliothèque SWT.
- Une historique locale des dernières modifications.
- Une exécution des applications dans une JVM dédiée sélectionnable avec possibilité d'utiliser un débogueur complet (points d'arrêts conditionnels, visualiser et modifier des variables, évaluer les expressions dans le contexte d'exécution, changement du code à chaud avec l'utilisation d'une JVM 1.4, ...)
- Propose le nécessaire pour développer de nouveaux plug-ins.

La plate-forme est entièrement internationalisée dans une dizaine de langue sous la forme d'un plug-in téléchargeable séparément.

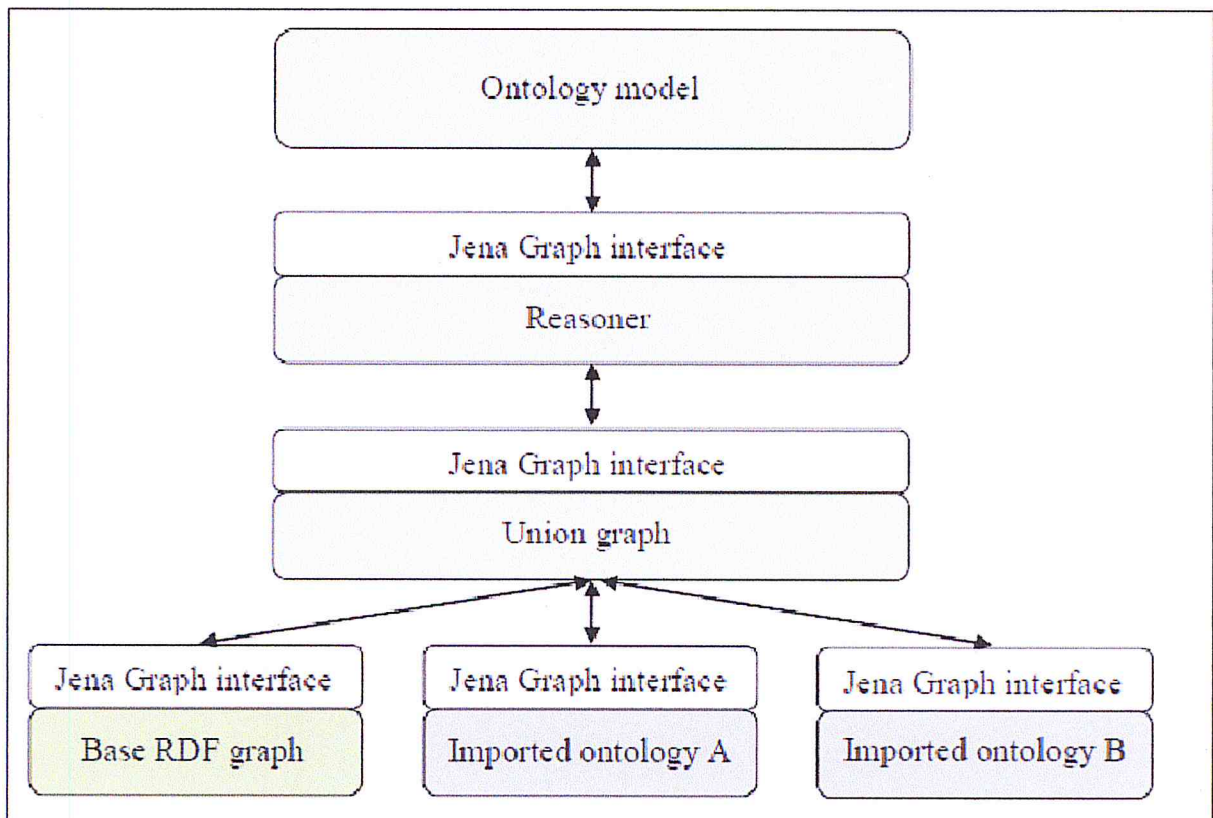
Annexe B : Jena

1. Présentation

Jena est un frame work écrit en Java, dont l'objectif est de fournir un environnement facilitant le développement d'applications dédiées au web sémantique. Jena permet de manipuler des documents RDF, RDFS et OWL, et fournit en plus un moteur d'inférences permettant des raisonnements sur les ontologies. Jena comporte les outils suivantes :

- Une API pour le langage RDF.
- Un module de lecture/ écriture sur les serveurs RDF : RDF/XML, N3 et NTRIPLE.
- Une API pour le langage OWL.
- RDQL : un langage d'interrogation des ontologies Implémentées en RDF et OWL.

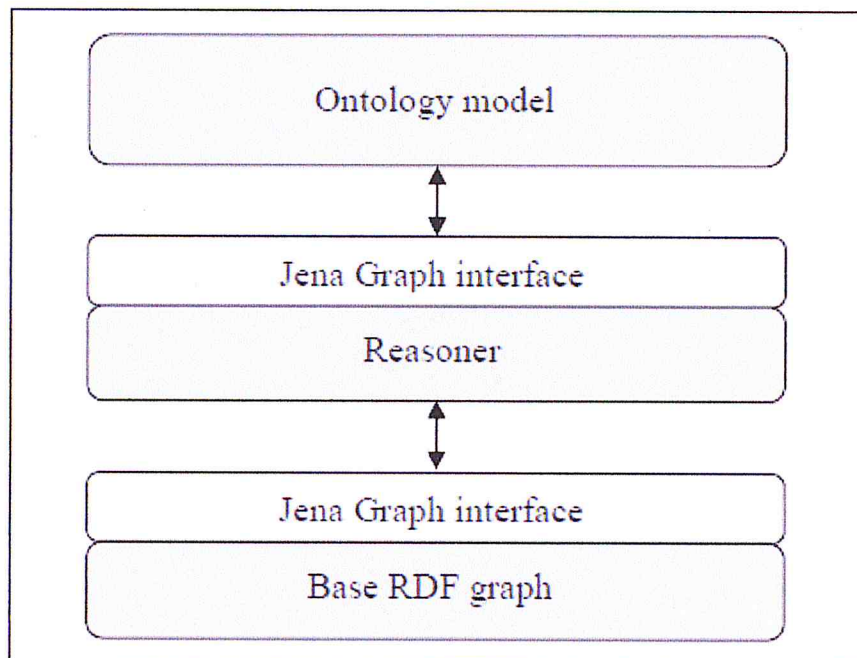
Le schéma ci-dessous montre la manière de création d'une ontologie dans Jena.



2. Le langage des ontologies et l'API JENA

Il existe plusieurs langages d'implémentation des ontologies, mais les langages les plus utilisés sont les langages RDF et OWL, ce dernier comporte trois variantes de complexité variable : OWL Full, OWL DL et le OWL Lite.

L'API JENA est neutre c'est-à-dire : les noms des classes java ne mentionnent pas le langage utilisé dans l'implémentation de l'ontologie i.e. : `OntClass`, `ObjectProperty`. Pour faire la distinction entre les différentes implémentations, chaque langage possède un profil qui permet d'identifier les constructeurs et les URI des classes et des propriétés autorisées. Chaque profil est lié à un modèle de l'ontologie qui est une version étendue de la classe du Modèle de JENA. Le Modèle général autorise l'accès aux déclarations dans une collection de données RDF, `OntModel` étend ceci en ajoutant le support pour les types d'objets attendus supposés être dans une ontologie: classes (dans une hiérarchie de classes), propriétés (dans une hiérarchie de propriétés) et les individus (instances).



3. Création d'une Ontologie avec l'API JENA

3.1. Connexion à une ontologie OWL

Pour pouvoir exploiter une ontologie créée par un éditeur d'ontologie il faut d'abord se connecter à cette ontologie, l'API JENA assure cette connexion par la commande suivante :

```
String URI="http://127.0.0.1:8080/jsp-examples/exemple_ontologie.owl";
```

3.2. Création d'un modèle

La création d'un modèle se fait par la méthode `ModelFactory`. Chaque modèle créé est une instance de la classe « `OntModel` ».

```
OntModel Model = ModelFactory.createOntologyModel ();
```

Cette commande assure la création d'un modèle d'ontologie standard c'est-à-dire une ontologie OWL-Full. Pour spécifier le langage utilisé, le développeur doit spécifier les paramètres de la méthode `createOntologyModel()` propres au langage qu'il utilise, dans le cas du langage OWL-DL la commande de création d'un modèle est la suivante :

```
OntModel model = ModelFactory.createOntologyModel (ProfileRegistry.DAML_LANG);
```

4. Méthodes de Manipulation des ontologies

<i>Méthodes des classes <code>OntClass</code> et <code>OntModel</code></i>	
<i>Méthode</i>	<i>Description</i>
<code>listClasses()</code>	Renvoie dans l'objet <i>Iterator</i> toutes les classes contenues dans le modèle.
<code>listHierarchyRootClasses()</code>	Renvoie dans un objet <i>Iterator</i> des objets <code>OntClass</code> <i>c</i>
<code>isUnionClass()</code>	Permet de savoir si une classe est le résultat d'une union d'autres classes.
<code>listSubClasses()</code>	Renvoie dans un <i>Iterator</i> les objets <code>OntClass</code> contenant les sous-classes d'une classe
<code>listSuperClasses()</code>	Renvoie dans un <i>Iterator</i> les objets <code>OntClass</code> contenant les super-classes de la classe en question.
<code>setSuperClass(Resource cls)</code>	Affirme que la classe est sous-classe d'une classe donnée.
<code>setSubClass(Resource cls)</code>	Affirme que la classe est super-classe d'une classe donnée.

isRestriction()	Réponse vrai si la classe est une restriction.
<i>Méthodes de la classe Restriction</i>	
asHasValueRestriction()	Permet de considérer une classe comme une restriction de type <i>HasValue</i>
asSomeValuesFromRestriction()	Permet de considérer une classe comme une restriction de type <i>SomeValueFrom</i>
asAllValuesFromRestriction()	Permet de considérer une classe comme une restriction de type <i>AllValueFrom</i>
asMinCardinalityRestriction()	Permet de considérer une classe comme une restriction de type <i>MinCardinality</i>

Annexe C : Table des correspondances

Concepts de l'Ontologie1	Concepts de l'Ontologie2	Correspondances Manuelles	Mesures de similarité	Correspondances de FONES
Document	Document	✓	1.0	✓
Document	Publication	✗	0.0	✗
Document	Book	✗	0.0	✗
Document	Proceedings	✗	0.0	✗
Document	Thesis	✗	0.0	✗
Document	Doctoral-Dissertation	✗	0.0	✗
Document	Master-Thesis	✗	0.0	✗
Document	Article	✗	0.0	✗
Document	Webpage	✗	0.0	✗
Document	Organization	✗	0.291	✗
Document	School	✗	0.0	✗
Document	University	✗	0.0	
Document	Person	✗	0.332	✗
Document	Series	✗	0.261	✗
Publication	Document	✗	0.0	✗
Publication	Publication	✓	1.0	✓
Publication	Book	✗	0.0	✗
Publication	Proceedings	✗	0.0	✗
Publication	Thesis	✗	0.0	✗
Publication	Doctoral-Dissertation	✗	0.0	✗
Publication	Master-Thesis	✗	0.0	✗
Publication	Article	✗	0.0	✗
Publication	Webpage	✗	0.303	✗
Publication	Organization	✗	0.0	✗
Publication	School	✗	0.0	✗
Publication	University	✗	0.0	✗
Publication	Person	✗	0.0	✗
Publication	Series	✗	0.0	✗
Article	Document	✗	0.0	✗
Article	Publication	✗	0.0	✗
Article	Book	✗	0.0	✗
Article	Proceedings	✗	0.423	✗
Article	Thesis	✗	0.265	✗

Annexe C : Table des correspondances

Article	Doctoral-Dissertation	x	0.0	x
Article	Master-Thesis	x	0.0	x
Article	Article	✓	1.0	✓
Article	Webpage	x	0.0	x
Article	Organization	x	0.0	x
Article	School	x	0.0	x
Article	University	x	0.0	x
Article	Person	x	0.0	x
Article	Series	x	0.0	x
Book	Document	x	0.0	x
Book	Publication	x	0.0	x
Book	Book	✓	1.0	✓
Book	Proceedings	x	0.273	x
Book	Thesis	x	0.0	x
Book	Doctoral-Dissertation	x	0.0	x
Book	Master-Thesis	x	0.0	x
Book	Article	x	0.0	x
Book	Webpage	x	0.0	x
Book	Organization	x	0.0	x
Book	School	x	0.0	x
Book	University	x	0.0	x
Book	Person	x	0.0	x
Book	Series	x	0.0	x
Collection	Document	x	0.0	x
Collection	Publication	x	0.0	x
Collection	Book	x	0.275	x
Collection	Proceedings	x	0.428	x
Collection	Thesis	x	0.319	x
Collection	Doctoral-Dissertation	x	0.0	x
Collection	Master-Thesis	x	0.0	x
Collection	Article	x	0.499	x
Collection	Webpage	x	0.0	x
Collection	Organization	x	0.0	x
Collection	School	x	0.0	x
Collection	University	x	0.0	x

Annexe C : Table des correspondances

Collection	Person	x	0.0	x
Collection	Series	x	0.0	x
Journal	Document	x	0.0	x
Journal	Publication	x	0.0	x
Journal	Book	x	0.285	x
Journal	Proceedings	x	0.362	x
Journal	Thesis	x	0.0	x
Journal	Doctoral-Dissertation	x	0.0	x
Journal	Master-Thesis	x	0.0	x
Journal	Article	x	0.329	x
Journal	Webpage	x	0.0	x
Journal	Organization	x	0.0	x
Journal	School	x	0.0	x
Journal	University	x	0.0	x
Journal	Person	x	0.0	x
Journal	Series	x	0.0	x
Magazine	Document	x	0.0	x
Magazine	Publication	x	0.0	x
Magazine	Book	x	0.0	x
Magazine	Proceedings	x	0.266	x
Magazine	Thesis	x	0.261	x
Magazine	Doctoral-Dissertation	x	0.0	x
Magazine	Master-Thesis	x	0.0	x
Magazine	Article	x	0.32	x
Magazine	Webpage	x	0.0	x
Magazine	Organization	x	0.0	x
Magazine	School	x	0.0	x
Magazine	University	x	0.0	x
Magazine	Person	x	0.0	x
Magazine	Series	x	0.0	x
Manual	Document	x	0.0	x
Manual	Publication	x	0.0	x
Manual	Book	x	0.0	x
Manual	Proceedings	x	0.253	x
Manual	Thesis	x	0.271	x

Annexe C : Table des correspondances

Manual	Doctoral-Dissertation	x	0.0	x
Manual	Master-Thesis	x	0.0	x
Manual	Article	x	0.341	x
Manual	Webpage	x	0.0	x
Manual	Organization	x	0.0	x
Manual	School	x	0.0	x
Manual	University	x	0.0	x
Manual	Person	x	0.0	x
Manual	Series	x	0.0	x
Proceeding	Document	x	0.0	x
Proceeding	Publication	x	0.0	x
Proceeding	Book	x	0.273	x
Proceeding	Proceedings	✓	0,98	✓
Proceeding	Thesis	x	0.315	x
Proceeding	Doctoral-Dissertation	x	0.0	x
Proceeding	Master-Thesis	x	0.0	x
Proceeding	Article	x	0.423	x
Proceeding	Webpage	x	0.0	x
Proceeding	Organization	x	0.0	x
Proceeding	School	x	0.0	x
Proceeding	University	x	0.0	x
Proceeding	Person	x	0.0	x
Proceeding	Series	x	0.0	x
Report	Document	x	0.0	x
Report	Publication	x	0.0	x
Report	Book	x	0.291	x
Report	Proceedings	x	0.296	x
Report	Thesis	x	0.271	x
Report	Doctoral-Dissertation	x	0.0	x
Report	Master-Thesis	x	0.0	x
Report	Article	x	0.0	x
Report	Webpage	x	0.0	x
Report	Organization	x	0.0	x
Report	School	x	0.0	x
Report	University	x	0.0	x

Annexe C : Table des correspondances

Report	Person	x	0.0	x
Report	Series	x	0.0	x
Project-Report	Document	x	0.0	x
Project-Report	Publication	x	0.0	x
Project-Report	Book	x	0.0	x
Project-Report	Proceedings	x	0.0	x
Project-Report	Thesis	x	0.0	x
Project-Report	Doctoral-Dissertation	x	0.0	x
Project-Report	Master-Thesis	x	0.0	x
Project-Report	Article	x	0.0	x
Project-Report	Webpage	x	0.0	x
Project-Report	Organization	x	0.0	x
Project-Report	School	x	0.0	x
Project-Report	University	x	0.0	x
Project-Report	Person	x	0.0	x
Project-Report	Series	x	0.0	x
Technical-Report	Document	x	0.0	x
Technical-Report	Publication	x	0.0	x
Technical-Report	Book	x	0.0	x
Technical-Report	Proceedings	x	0.0	x
Technical-Report	Thesis	x	0.0	x
Technical-Report	Doctoral-Dissertation	x	0.0	x
Technical-Report	Master-Thesis	x	0.0	x
Technical-Report	Article	x	0.0	x
Technical-Report	Web-page	x	0.0	x
Technical-Report	Organization	x	0.0	x
Technical-Report	School	x	0.0	x
Technical-Report	University	x	0.0	x
Technical-Report	Person	x	0.0	x
Technical-Report	Series	x	0.0	x
Research-Paper	Document	x	0.0	x
Research-Paper	Publication	x	0.0	x
Research-Paper	Book	x	0.0	x
Research-Paper	Proceedings	x	0.287	x
Research-Paper	Thesis	x	0.282	x

Annexe C : Table des correspondances

Research-Paper	Doctoral-Dissertation	x	0.0	x
Research-Paper	Master-Thesis	x	0.0	x
Research-Paper	Article	x	0.347	x
Research-Paper	Webpage	x	0.0	x
Research-Paper	Organization	x	0.0	x
Research-Paper	School	x	0.0	x
Research-Paper	University	x	0.0	x
Research-Paper	Person	x	0.0	x
Research-Paper	Series	x	0.0	x
Thesis	Document	x	0.0	x
Thesis	Publication	x	0.0	x
Thesis	Book	x	0.0	x
Thesis	Proceedings	x	0.315	x
Thesis	Thesis	✓	1.0	✓
Thesis	Doctoral-Dissertation	x	0.0	x
Thesis	Master-Thesis	x	0.0	x
Thesis	Article	x	0.265	x
Thesis	Webpage	x	0.0	x
Thesis	Organization	x	0.0	x
Thesis	School	x	0.0	x
Thesis	University	x	0.0	x
Thesis	Person	x	0.0	x
Thesis	Series	x	0.0	x
Diploma-Thesis	Document	x	0.0	x
Diploma-Thesis	Publication	x	0.0	x
Diploma-Thesis	Book	x	0.0	x
Diploma-Thesis	Proceedings	x	0.0	x
Diploma-Thesis	Thesis	x	0.0	x
Diploma-Thesis	Doctoral-Dissertation	x	0.493	x
Diploma-Thesis	Master-Thesis	x	0.457	x
Diploma-Thesis	Article	x	0.0	x
Diploma-Thesis	Webpage	x	0.0	x
Diploma-Thesis	Organization	x	0.0	x
Diploma-Thesis	School	x	0.0	x
Diploma-Thesis	University	x	0.0	x

Annexe C : Table des correspondances

Diploma-Thesis	Person	x	0.0	x
Diploma-Thesis	Series	x	0.0	x
Docrtorat-Thesis	Document	x	0.0	x
Docrtorat-Thesis	Publication	x	0.0	x
Docrtorat-Thesis	Book	x	0.0	x
Docrtorat-Thesis	Proceedings	x	0.0	x
Docrtorat-Thesis	Thesis	x	0.0	x
Docrtorat-Thesis	Doctoral-Dissertation	✓	0.922	✓
Docrtorat-Thesis	Master-Thesis	x	0.562	x
Docrtorat-Thesis	Article	x	0.0	x
Docrtorat-Thesis	Webpage	x	0.0	x
Docrtorat-Thesis	Organization	x	0.0	x
Docrtorat-Thesis	School	x	0.0	x
Docrtorat-Thesis	University	x	0.0	x
Docrtorat-Thesis	Person	x	0.0	x
Docrtorat-Thesis	Series	x	0.0	x
Master-Thesis	Document	x	0.0	x
Master -Thesis	Publication	x	0.0	x
Master -Thesis	Book	x	0.0	x
Master -Thesis	Proceedings	x	0.0	x
Master -Thesis	Thesis	x	0.0	x
Master -Thesis	Doctoral-Dissertation	x	0.406	x
Master -Thesis	Master-Thesis	✓	1.0	✓
Master -Thesis	Article	x	0.0	x
Master -Thesis	Webpage	x	0.0	x
Master -Thesis	Organization	x	0.0	x
Master -Thesis	School	x	0.0	x
Master -Thesis	University	x	0.0	x
Master -Thesis	Person	x	0.0	x
Master -Thesis	Series	x	0.0	x
Organization	Document	x	0.291	x
Organization	Publication	x	0.0	x
Organization	Book	x	0.0	x
Organization	Proceedings	x	0.0	x
Organization	Thesis	x	0.0	x

Annexe C : Table des correspondances

Organization	Doctoral-Dissertation	x	0.0	x
Organization	Master-Thesis	x	0.0	x
Organization	Article	x	0.0	x
Organization	Webpage	x	0.0	x
Organization	Organization	✓	1.0	✓
Organization	School	x	0.0	x
Organization	University	x	0.0	x
Organization	Person	x	0.311	x
Organization	Series	x	0.311	x
Association	Document	x	0.0	x
Association	Publication	x	0.0	x
Association	Book	x	0.0	x
Association	Proceedings	x	0.0	x
Association	Thesis	x	0.0	x
Association	Doctoral-Dissertation	x	0.0	x
Association	Master-Thesis	x	0.0	x
Association	Article	x	0.0	x
Association	Webpage	x	0.0	x
Association	Organization	x	0.0	x
Association	School	x	0.315	x
Association	University	x	0.0	x
Association	Person	x	0.0	x
Association	Series	x	0.0	x
Department	Document	x	0.0	x
Department	Publication	x	0.0	x
Department	Book	x	0.0	x
Department	Proceedings	x	0.0	x
Department	Thesis	x	0.0	x
Department	Doctoral-Dissertation	x	0.0	x
Department	Master-Thesis	x	0.0	x
Department	Article	x	0.0	x
Department	Webpage	x	0.0	x
Department	Organization	x	0.0	x
Department	School	x	0.0	x
Department	University	x	0.0	x

Annexe C : Table des correspondances

Department	Person	x	0.0	x
Department	Series	x	0.0	x
Entreprise	Document	x	0.0	x
Entreprise	Publication	x	0.0	x
Entreprise	Book	x	0.0	x
Entreprise	Proceedings	x	0.0	x
Entreprise	Thesis	x	0.0	x
Entreprise	Doctoral-Dissertation	x	0.0	x
Entreprise	Master-Thesis	x	0.0	x
Entreprise	Article	x	0.0	x
Entreprise	Webpage	x	0.0	x
Entreprise	Organization	x	0.0	x
Entreprise	School	x	0.0	x
Entreprise	University	x	0.0	x
Entreprise	Person	x	0.0	x
Entreprise	Series	x	0.0	x
Institute	Document	x	0.0	x
Institute	Publication	x	0.0	x
Institute	Book	x	0.0	x
Institute	Proceedings	x	0.0	x
Institute	Thesis	x	0.0	x
Institute	Doctoral-Dissertation	x	0.0	x
Institute	Master-Thesis	x	0.0	x
Institute	Article	x	0.0	x
Institute	Webpage	x	0.0	x
Institute	Organization	x	0.0	x
Institute	School	x	0.0	x
Institute	University	x	0.0	x
Institute	Person	x	0.0	x
Institute	Series	x	0.0	x
University	Document	x	0.0	x
University	Publication	x	0.0	x
University	Book	x	0.0	x
University	Proceedings	x	0.0	x
University	Thesis	x	0.0	x

Annexe C : Table des correspondances

University	Doctoral-Dissertation	x	0.0	x
University	Master-Thesis	x	0.0	x
University	Article	x	0.0	x
University	Webpage	x	0.0	x
University	Organization	x	0.0	x
University	School	x	0.0	x
University	University	✓	1.0	✓
University	Person	x	0.0	x
University	Series	x	0.0	x
Person	Document	x	0.332	x
Person	Publication	x	0.0	x
Person	Book	x	0.0	x
Person	Proceedings	x	0.0	x
Person	Thesis	x	0.0	x
Person	Doctoral-Dissertation	x	0.0	x
Person	Master-Thesis	x	0.0	x
Person	Article	x	0.0	x
Person	Webpage	x	0.0	x
Person	Organization	x	0.311	x
Person	School	x	0.0	x
Person	University	x	0.0	x
Person	Person	✓	1.0	✓
Person	Series	x	0.441	x
Employee	Document	x	0.0	x
Employee	Publication	x	0.0	x
Employee	Book	x	0.0	x
Employee	Proceedings	x	0.0	x
Employee	Thesis	x	0.0	x
Employee	Doctoral-Dissertation	x	0.0	x
Employee	Master-Thesis	x	0.0	x
Employee	Article	x	0.0	x
Employee	Webpage	x	0.0	x
Employee	Organization	x	0.0	x
Employee	School	x	0.0	x
Employee	University	x	0.0	x

Annexe C : Table des correspondances

Employee	Person	x	0.0	x
Employee	Series	x	0.0	x
Academic-Staff	Document	x	0.0	x
Academic-Staff	Publication	x	0.0	x
Academic-Staff	Book	x	0.0	x
Academic-Staff	Proceedings	x	0.0	x
Academic-Staff	Thesis	x	0.0	x
Academic-Staff	Doctoral-Dissertation	x	0.0	x
Academic-Staff	Master-Thesis	x	0.0	x
Academic-Staff	Article	x	0.0	x
Academic-Staff	Webpage	x	0.0	x
Academic-Staff	Organization	x	0.0	x
Academic-Staff	School	x	0.0	x
Academic-Staff	University	x	0.0	x
Academic-Staff	Person	x	0.0	x
Academic-Staff	Series	x	0.0	x
Administrative-Staff	Document	x	0.0	x
Administrative-Staff	Publication	x	0.0	x
Administrative-Staff	Book	x	0.0	x
Administrative-Staff	Proceedings	x	0.0	x
Administrative-Staff	Thesis	x	0.0	x
Administrative-Staff	Doctoral-Dissertation	x	0.0	x
Administrative-Staff	Master-Thesis	x	0.0	x
Administrative-Staff	Article	x	0.0	x
Administrative-Staff	Webpage	x	0.0	x
Administrative-Staff	Organization	x	0.0	x
Administrative-Staff	School	x	0.0	x
Administrative-Staff	University	x	0.0	x
Administrative-Staff	Person	x	0.0	x
Administrative-Staff	Series	x	0.0	x
Manager	Document	x	0.0	x
Manager	Publication	x	0.0	x
Manager	Book	x	0.0	x
Manager	Proceedings	x	0.0	x
Manager	Thesis	x	0.0	x

Annexe C : Table des correspondances

Manager	Doctoral-Dissertation	x	0.0	x
Manager	Master-Thesis	x	0.0	x
Manager	Article	x	0.0	x
Manager	Webpage	x	0.0	x
Manager	Organization	x	0.0	x
Manager	School	x	0.0	x
Manager	University	x	0.0	x
Manager	Person	x	0.0	x
Manager	Series	x	0.0	x
Technical-Staff	Document	x	0.0	x
Technical-Staff	Publication	x	0.0	x
Technical-Staff	Book	x	0.0	x
Technical-Staff	Proceedings	x	0.0	x
Technical-Staff	Thesis	x	0.0	x
Technical-Staff	Doctoral-Dissertation	x	0.0	x
Technical-Staff	Master-Thesis	x	0.0	x
Technical-Staff	Article	x	0.0	x
Technical-Staff	Webpage	x	0.0	x
Technical-Staff	Organization	x	0.0	x
Technical-Staff	School	x	0.0	x
Technical-Staff	University	x	0.0	x
Technical-Staff	Person	x	0.0	x
Technical-Staff	Series	x	0.0	x
Student	Document	x	0.0	x
Student	Publication	x	0.0	x
Student	Book	x	0.0	x
Student	Proceedings	x	0.0	x
Student	Thesis	x	0.0	x
Student	Doctoral-Dissertation	x	0.0	x
Student	Master-Thesis	x	0.0	x
Student	Article	x	0.0	x
Student	Webpage	x	0.0	x
Student	Organization	x	0.0	x
Student	School	x	0.0	x
Student	University	x	0.0	x

Annexe C : Table des correspondances

Student	Person	x	0.0	x
Student	Series	x	0.0	x
Graduate	Document	x	0.0	x
Graduate	Publication	x	0.0	x
Graduate	Book	x	0.0	x
Graduate	Proceedings	x	0.0	x
Graduate	Thesis	x	0.0	x
Graduate	Doctoral-Dissertation	x	0.0	x
Graduate	Master-Thesis	x	0.0	x
Graduate	Article	x	0.0	x
Graduate	Webpage	x	0.0	x
Graduate	Organization	x	0.0	x
Graduate	School	x	0.0	x
Graduate	University	x	0.0	x
Graduate	Person	x	0.0	x
Graduate	Series	x	0.0	x
Under-Graduate	Document	x	0.0	x
Under-Graduate	Publication	x	0.0	x
Under-Graduate	Book	x	0.0	x
Under-Graduate	Proceedings	x	0.0	x
Under-Graduate	Thesis	x	0.0	x
Under-Graduate	Doctoral-Dissertation	x	0.0	x
Under-Graduate	Master-Thesis	x	0.0	x
Under-Graduate	Article	x	0.0	x
Under-Graduate	Webpage	x	0.0	x
Under-Graduate	Organization	x	0.0	x
Under-Graduate	School	x	0.0	x
Under-Graduate	University	x	0.0	x
Under-Graduate	Person	x	0.0	x
Under-Graduate	Series	x	0.0	x
Research-Group	Document	x	0.0	x
Research-Group	Publication	x	0.0	x
Research-Group	Book	x	0.0	x
Research-Group	Proceedings	x	0.0	x
Research-Group	Thesis	x	0.0	x

Annexe C : Table des correspondances

Research-Group	Doctoral-Dissertation	x	0.0	x
Research-Group	Master-Thesis	x	0.0	x
Research-Group	Article	x	0.0	x
Research-Group	Webpage	x	0.0	x
Research-Group	Organization	x	0.0	x
Research-Group	School	x	0.308	x
Research-Group	University	x	0.0	x
Research-Group	Person	x	0.0	x
Research-Group	Series	x	0.0	x
Topic	Document	x	0.349	x
Topic	Publication	x	0.0	x
Topic	Book	x	0.0	x
Topic	Proceedings	x	0.0	x
Topic	Thesis	x	0.0	x
Topic	Doctoral-Dissertation	x	0.0	x
Topic	Master-Thesis	x	0.0	x
Topic	Article	x	0.0	x
Topic	Webpage	x	0.0	x
Topic	Organization	x	0.259	x
Topic	School	x	0.0	x
Topic	University	x	0.0	x
Topic	Person	x	0.0	x
Topic	Series	x	0.279	x
Research-Topic	Document	x	0.0	x
Research-Topic	Publication	x	0.0	x
Research-Topic	Book	x	0.0	x
Research-Topic	Proceedings	x	0.0	x
Research-Topic	Thesis	x	0.0	x
Research-Topic	Doctoral-Dissertation	x	0.0	x
Research-Topic	Master-Thesis	x	0.0	x
Research-Topic	Article	x	0.0	x
Research-Topic	Webpage	x	0.0	x
Research-Topic	Organization	x	0.0	x
Research-Topic	School	x	0.0	x
Research-Topic	University	x	0.0	x

Annexe C : Table des correspondances

Research-Topic	Person	x	0.0	x
Research-Topic	Series	x	0.0	x
Correspondances produites par le Système : S = 11				
Correspondances produites manuellement : H = 11				

Annexe D : Les Ontologies de tests

Ontologie1

```
-<rdf:RDF xml:base="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl">
  <owl:Ontology rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl">
    -<!--
      //////////////////////////////////////
      //
      // Object Properties
      //
      //////////////////////////////////////
    -->
    -->
  -<!--
    http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#affiliation
  -->
  -<owl:ObjectProperty rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#affiliation">
    <rdfs:range rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Organization"/>
    <rdfs:domain rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Person"/>
    <owl:ObjectProperty>
  -<!--
    http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#cite
  -->
  -<owl:ObjectProperty rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#cite">
    <rdfs:range rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Document"/>
    -<rdfs:domain>
      -<owl:Class>
        -<owl:unionOf rdf:parseType="Collection">
          <rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Document"/>
          <rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Person"/>
        </owl:unionOf>
        <owl:Class>
      </rdfs:domain>
    </owl:ObjectProperty>
  -<!--
    http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#citedBy
  -->
  -<owl:ObjectProperty rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#citedBy">
    <rdfs:domain rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Document"/>
    <owl:inverseOf rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#cite"/>
    -<rdfs:range>
      -<owl:Class>
        -<owl:unionOf rdf:parseType="Collection">
          <rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Document"/>
          <rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Person"/>
        </owl:unionOf>
        <owl:Class>
      </rdfs:range>
    </owl:ObjectProperty>
  -<!--
    http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#cooperateWith
  -->
  -<owl:ObjectProperty rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#cooperateWith">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#SymmetricProperty"/>
    -<rdfs:domain>
      -<owl:Class>
        -<owl:unionOf rdf:parseType="Collection">
          <rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Organization"/>
          <rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Person"/>
        </owl:unionOf>
        <owl:Class>
      </rdfs:domain>
    -<rdfs:range>
      -<owl:Class>
```

```

- <owl:unionOf rdf:parseType="Collection">
  <rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Organization"/>
  <rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Person"/>
  <owl:unionOf>
    <owl:Class>
      <rdfs:range>
    <owl:ObjectProperty>
- <!--
  http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#employs
-->
- <owl:ObjectProperty rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#employs">
  <rdfs:domain rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Organization"/>
  <rdfs:range rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Person"/>
  <owl:inverseOf rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#affiliation"/>
  <owl:ObjectProperty>
- <!--
  http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#head
-->
- <owl:ObjectProperty rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#head">
  <rdfs:range rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Person"/>
  <rdfs:domain rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#ResearchGroup"/>
  <owl:inverseOf rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#headOfGroup"/>
  <owl:ObjectProperty>
- <!--
  http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#headOfGroup
-->
- <owl:ObjectProperty rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#headOfGroup">
  <rdfs:domain rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Person"/>
  <rdfs:range rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#ResearchGroup"/>
  <owl:ObjectProperty>
- <!--
  http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#isWorkedOnBy
-->
- <owl:ObjectProperty rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#isWorkedOnBy">
  <rdfs:range rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Person"/>
  <rdfs:domain rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Topic"/>
  <owl:ObjectProperty>
- <!--
  http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#member
-->
- <owl:ObjectProperty rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#member">
  <rdfs:domain rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Organization"/>
  - <rdfs:range>
    - <owl:Class>
      - <owl:unionOf rdf:parseType="Collection">
        <rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Organization"/>
        <rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Person"/>
        <owl:unionOf>
          <owl:Class>
            <rdfs:range>
          <owl:ObjectProperty>
- <!--
  http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#publication
-->
- <owl:ObjectProperty rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#publication">
  <rdfs:range rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Document"/>
  - <rdfs:domain>
    - <owl:Class>
      - <owl:unionOf rdf:parseType="Collection">
        <rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Organization"/>
        <rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Person"/>
        <owl:unionOf>

```



```

    <owl:Class>
    <rdfs:domain>
    <owl:ObjectProperty>
- <!--
    http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#publisherOf
    -->
- <owl:ObjectProperty rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#publisherOf">
    <rdfs:range rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Document"/>
    - <rdfs:domain>
      - <owl:Class>
        - <owl:unionOf rdf:parseType="Collection">
          <rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Organization">
          <rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Person">
          <owl:unionOf>
          <owl:Class>
          <rdfs:domain>
        <owl:ObjectProperty>
- <!--
    http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#studiesAt
    -->
- <owl:ObjectProperty rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#studiesAt">
    <rdfs:domain rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Student"/>
    <rdfs:range rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#University"/>
    <owl:ObjectProperty>
- <!--
    http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#supervises
    -->
- <owl:ObjectProperty rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#supervises">
    <rdfs:domain rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Person"/>
    <rdfs:range rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Person"/>
    <owl:ObjectProperty>

- <!--
    http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#supervisor
    -->
- <owl:ObjectProperty rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#supervisor">
    <rdfs:domain rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Person"/>
    <rdfs:range rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Person"/>
    <owl:inverseOf rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#supervises"/>
    <owl:ObjectProperty>
- <!--
    http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#technicalReport
    -->
- <owl:ObjectProperty rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#technicalReport">
    <rdfs:range rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#TechnicalReport"/>
    - <rdfs:domain>
      - <owl:Class>
        - <owl:unionOf rdf:parseType="Collection">
          <rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Organization">
          <rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Person">
          <owl:unionOf>
          <owl:Class>
          <rdfs:domain>
        <owl:ObjectProperty>
- <!--

    ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
    //
    // Data properties
    //
    ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

-->
- <!--

```

```
-<!--  
  http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Publicationtype  
-->  
-<owl:DatatypeProperty rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Publicationtype">  
  <rdfs:domain rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Publication"/>  
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>  
</owl:DatatypeProperty>  
-<!--  
  http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#email  
-->  
-<owl:DatatypeProperty rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#email">  
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>  
  -<rdfs:domain>  
  -<owl:Class>  
    -<owl:unionOf rdf:parseType="Collection">  
      <rdfs:Description rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Organization"/>  
      <rdfs:Description rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Person"/>  
    </owl:unionOf>  
    <owl:Class>  
  </rdfs:domain>  
</owl:DatatypeProperty>  
-<!--  
  http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#isbn  
-->  
-<owl:DatatypeProperty rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#isbn">  
  <rdfs:domain rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Book"/>  
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>  
</owl:DatatypeProperty>  
-<!--  
  http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#keywords  
-->  
-<owl:DatatypeProperty rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#keywords">  
  <rdfs:domain rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Publication"/>  
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>  
</owl:DatatypeProperty>  
-<!--  
  http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#name  
-->  
-<owl:DatatypeProperty rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#name">  
  <rdfs:domain rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Person"/>  
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>  
</owl:DatatypeProperty>  
-<!--  
  http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#pages  
-->  
-<owl:DatatypeProperty rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#pages">  
  <rdfs:domain rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Publication"/>  
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#integer"/>  
</owl:DatatypeProperty>  
-<!--  
  http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#phone  
-->  
-<owl:DatatypeProperty rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#phone">  
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>  
  -<rdfs:domain>  
  -<owl:Class>  
    -<owl:unionOf rdf:parseType="Collection">  
      <rdfs:Description rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Organization"/>  
      <rdfs:Description rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Person"/>  
    </owl:unionOf>  
    <owl:Class>  
  </rdfs:domain>
```



```
-->
- <owl:Class rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Collection">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Publication"/>
  <owl:Class>
- <!--
  http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Department
-->
- <owl:Class rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Department">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Organization"/>
  <owl:Class>
- <!--
  http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#DiplomaThesis
-->
- <owl:Class rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#DiplomaThesis">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Thesis"/>
  <owl:Class>
- <!--
  http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Document
-->
- <owl:Class rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Document">
- <!--
  http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Employee
-->
- <owl:Class rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Employee">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Person"/>
  <owl:Class>
- <!--
  http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Entreprise
-->
- <owl:Class rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Entreprise">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Organization"/>
  <owl:Class>
- <!--
  http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Graduate
-->
- <owl:Class rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Graduate">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Student"/>
  <owl:Class>
- <!--
  http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Institute
-->
- <owl:Class rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Institute">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Organization"/>
  <owl:Class>
- <!--
  http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Journal1
-->
- <owl:Class rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Journal">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Publication"/>
  <owl:Class>
- <!--
  http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Magazine
-->
- <owl:Class rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Magazine">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Publication"/>
  <owl:Class>
- <!--
  http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Manager
-->
- <owl:Class rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Manager">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Employee"/>
  <owl:Class>
- <!--
  http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Manual
```

```
-->
- <owl:Class rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Manual">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Publication"/>
</owl:Class>
- <!--
  http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#MasterThesis
-->
- <owl:Class rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#MasterThesis">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Thesis"/>
</owl:Class>
- <!--
  http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Organization
-->
- <owl:Class rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Organization">
</owl:Class>
- <!--
  http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Person
-->
- <owl:Class rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Person">
</owl:Class>
- <!--
  http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Proceedings
-->
- <owl:Class rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Proceedings">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Publication"/>
</owl:Class>
- <!--
  http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#ProjectReport
-->
- <owl:Class rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#ProjectReport">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Report"/>
</owl:Class>
- <!--
  http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Publication
-->
- <owl:Class rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Publication">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Document"/>
</owl:Class>
- <!--
  http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Report
-->
- <owl:Class rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Report">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Publication"/>
</owl:Class>
- <!--
  http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#ResearchGroup
-->
- <owl:Class rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#ResearchGroup">
</owl:Class>
- <!--
  http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#ResearchPaper
-->
- <owl:Class rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#ResearchPaper">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Publication"/>
</owl:Class>
- <!--
  http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#ResearchTopic
-->
- <owl:Class rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#ResearchTopic">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Topic"/>
</owl:Class>
- <!--
  http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Student
-->
- <owl:Class rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Student">
```



```
<rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Person"/>
</owl:Class>
- <!--
  http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#TechnicalReport
-->
- <owl:Class rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#TechnicalReport">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Report"/>
  </owl:Class>
- <!--
  http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#TechnicalStaff
-->
- <owl:Class rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#TechnicalStaff">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Employee"/>
  </owl:Class>
- <!--
  http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Thesis
-->
- <owl:Class rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Thesis">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Publication"/>
  </owl:Class>
- <!--
  http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Topic
-->
  <owl:Class rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Topic"/>
- <!--
  http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#UnderGraduate
-->
- <owl:Class rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#UnderGraduate">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Student"/>
  </owl:Class>

- <!--
  http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#University
-->
- <owl:Class rdf:about="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#University">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ontologies/2011/5/Ontology1308645496223.owl#Organization"/>
  </owl:Class>
</rdf:RDF>
- <!--
  Generated by the OWL API (version 3.1.0.20069) http://owlapi.sourceforge.net
-->
```

Ontologie 2

```

- <rdf:RDF xml:base="http://swat.cse.lehigh.edu/resources/onto/dblp.owl">
- <owl:Ontology rdf:about="http://swat.cse.lehigh.edu/resources/onto/dblp.owl">
  <rdfs:label rdf:datatype="http://www.w3.org/1999/02/22-rdf-syntax-ns#PlainLiteral">Dblp Ontology</rdfs:label>
  <owl:versionInfo rdf:datatype="http://www.w3.org/1999/02/22-rdf-syntax-ns#PlainLiteral">2008-05-27</owl:versionInfo>
- <rdfs:comment rdf:datatype="http://www.w3.org/1999/02/22-rdf-syntax-ns#PlainLiteral">
  This file specifies in RDF Schema format the classes and properties for DBLP. Adapted from SWETO_DBLP from UGA.
  </rdfs:comment>
</owl:Ontology>
- <!--
  //////////////////////////////////////
  //
  // Annotation properties
  //
  //////////////////////////////////////
-->
<owl:AnnotationProperty rdf:about="http://www.w3.org/2002/07/owl#versionInfo">
<owl:AnnotationProperty rdf:about="http://www.w3.org/2000/01/rdf-schema#label">
<owl:AnnotationProperty rdf:about="http://www.w3.org/2000/01/rdf-schema#comment">
- <!--
  //////////////////////////////////////
  //
  // Datatypes
  //
  //////////////////////////////////////
-->
<!-- http://www.w3.org/2001/XMLSchema#date -->
<rdfs:Datatype rdf:about="http://www.w3.org/2001/XMLSchema#date">
- <!--
  //////////////////////////////////////
  //
  // Object Properties
  //
  //////////////////////////////////////
-->
- <!--
  http://swat.cse.lehigh.edu/resources/onto/dblp.owl#affiliation
-->
- <owl:ObjectProperty rdf:about="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#affiliation">
  <rdfs:range rdf:resource="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Organization">
  <rdfs:domain rdf:resource="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Person">
  </owl:ObjectProperty>
- <!--
  http://swat.cse.lehigh.edu/resources/onto/dblp.owl#at_university
-->
- <owl:ObjectProperty rdf:about="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#at_university">
  <rdfs:domain rdf:resource="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Publication">
  <rdfs:range rdf:resource="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#University">
  </owl:ObjectProperty>
- <!--
  http://swat.cse.lehigh.edu/resources/onto/dblp.owl#author
-->
- <owl:ObjectProperty rdf:about="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#author">
  <rdfs:range rdf:resource="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Person">
  <rdfs:domain rdf:resource="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Publication">
  </owl:ObjectProperty>
- <!--
  http://swat.cse.lehigh.edu/resources/onto/dblp.owl#cites
-->

```



```

- <owl:ObjectProperty rdf:about="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#cites">
  <rdfs:domain rdf:resource="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Publication"/>
  <rdfs:range rdf:resource="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Publication"/>
</owl:ObjectProperty>
- <!--
  http://swat.cse.lehigh.edu/resources/onto/dblp.owl#coauthor
  -->
- <owl:ObjectProperty rdf:about="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#coauthor">
  <rdfs:range rdf:resource="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Person"/>
  <rdfs:domain rdf:resource="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Person"/>
</owl:ObjectProperty>
- <!--
  http://swat.cse.lehigh.edu/resources/onto/dblp.owl#in_series
  -->
- <owl:ObjectProperty rdf:about="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#in_series">
  <rdfs:domain rdf:resource="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Publication"/>
  <rdfs:range rdf:resource="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Series"/>
</owl:ObjectProperty>
- <!--
  //////////////////////////////////////
  //
  // Data properties
  //
  //////////////////////////////////////
  -->
- <!--
  http://swat.cse.lehigh.edu/resources/onto/dblp.owl#book_title
  -->
- <owl:DatatypeProperty rdf:about="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#book_title">
  <rdfs:domain rdf:resource="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Publication"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
- <!--
  http://swat.cse.lehigh.edu/resources/onto/dblp.owl#isbn
  -->
- <owl:DatatypeProperty rdf:about="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#isbn">
  <rdfs:domain rdf:resource="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Publication"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
- <!--
  http://swat.cse.lehigh.edu/resources/onto/dblp.owl#last_modified_date
  -->
- <owl:DatatypeProperty rdf:about="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#last_modified_date">
  <rdfs:domain rdf:resource="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Document"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#date"/>
</owl:DatatypeProperty>
- <!--
  http://swat.cse.lehigh.edu/resources/onto/dblp.owl#number
  -->
- <owl:DatatypeProperty rdf:about="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#number">
  <rdfs:domain rdf:resource="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Publication"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
</owl:DatatypeProperty>
- <!--
  http://swat.cse.lehigh.edu/resources/onto/dblp.owl#title
  -->
- <owl:DatatypeProperty rdf:about="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#title">
  <rdfs:domain rdf:resource="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Publication"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
- <!--

```

```

    http://swat.cse.lehigh.edu/resources/onto/dblp.owl#volume
-->
- <owl:DatatypeProperty rdf:about="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#volume">
  <rdfs:domain rdf:resource="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Publication"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <owl:DatatypeProperty>
- <!--

    //////////////////////////////////////
    //
    // Classes
    //
    //////////////////////////////////////

-->
- <!--
    http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Article
-->
- <owl:Class rdf:about="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Article">
  <rdfs:subClassOf rdf:resource="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Publication"/>
  <owl:Class>
- <!--
    http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Book
-->
- <owl:Class rdf:about="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Book">
  <rdfs:subClassOf rdf:resource="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Publication"/>
  <owl:Class>
- <!--
    http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Doctoral_Dissertation
-->
- <owl:Class rdf:about="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Doctoral_Dissertation">
  <rdfs:subClassOf rdf:resource="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Thesis"/>
  <owl:Class>
- <!--
    http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Document
-->
  <owl:Class rdf:about="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Document"/>
- <!--
    http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Master_Thesis
-->
- <owl:Class rdf:about="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Master_Thesis">
  <rdfs:subClassOf rdf:resource="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Thesis"/>
  <owl:Class>
- <!--
    http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Organization
-->
  <owl:Class rdf:about="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Organization"/>
- <!--
    http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Person
-->
  <owl:Class rdf:about="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Person"/>
- <!--
    http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Proceedings
-->
- <owl:Class rdf:about="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Proceedings">
  <rdfs:subClassOf rdf:resource="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Publication"/>
  <owl:Class>
- <!--
    http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Publication
-->
- <owl:Class rdf:about="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Publication">
  <rdfs:subClassOf rdf:resource="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Document"/>
  <owl:Class>

```



```
-<!--  
  http://swat.cse.lehigh.edu/resources/onto/dblp.owl#School  
  -->  
-<owl:Class rdf:about="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#School">  
  <rdfs:subClassOf rdf:resource="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Organization"/>  
</owl:Class>  
-<!--  
  http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Series  
  -->  
-<owl:Class rdf:about="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Series">  
-<!--  
  http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Thesis  
  -->  
-<owl:Class rdf:about="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Thesis">  
  <rdfs:subClassOf rdf:resource="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Publication"/>  
</owl:Class>  
-<!--  
  http://swat.cse.lehigh.edu/resources/onto/dblp.owl#University  
  -->  
-<owl:Class rdf:about="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#University">  
  <rdfs:subClassOf rdf:resource="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#School"/>  
</owl:Class>  
-<!--  
  http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Webpage  
  -->  
-<owl:Class rdf:about="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Webpage">  
  <rdfs:subClassOf rdf:resource="http://swat.cse.lehigh.edu/resources/onto/dblp.owl#Document"/>  
</owl:Class>  
</rdf:RDF>  
-<!--  
  Generated by the OWL API (version 3.1.0.20069) http://owlapi.sourceforge.net  
  -->
```

Glossaire

Glossaire

Alignement

C'est le processus de découverte des correspondances entre deux ontologies différentes (équivalence de concepts ou de relations, subsumption, etc.)

Graphes conceptuels

Les graphes conceptuels sont un formalisme de représentation de connaissances et de raisonnements. Ce formalisme a été introduit par John F. Sowa (en) en 1976.

Logique de description

Langages de représentation de connaissances. Il peut être utilisé pour représenter la connaissance terminologique d'un domaine d'application d'une manière formelle et structurée.

Métadonnée

Donnée qui renseigne sur la nature de certaines autres données et qui permet ainsi leur utilisation pertinente.

METHONTOLOGIE

Est une méthodologie de construction d'ontologie à partir de début, elle se base sur un cycle de vie bien défini.

Langage de prédicat

Langage formel riche, qui permet d'exprimer des propriétés vraies pour certains individus pris dans un ensemble (relations).

Web

Le World Wide Web, communément appelé le Web, parfois la Toile, ou la « toile mondiale », correspond au service d'informations en ligne (textes, sons, animations, images, multimédias) offerts et couramment utilisés sur Internet.

Web sémantique

Le Web sémantique désigne un ensemble de technologies visant à rendre le contenu des ressources du World Wide Web accessible et utilisable par les programmes et agents logiciels, grâce à un système de métadonnées formelles, utilisant notamment la famille de langages développés par le W3C.

Références bibliographiques

Références bibliographiques

- [NEE et al, 91] : Neeches R., Fikes R. E., Finin T., Gruber T. R., Senator T. et Swartout. Enabling technology for knowledge sharing, W.R. *AI Magazine*. vol. 12, no 3, (1991).
- [GRU, 93] : Gruber T, A translation approach to portable ontology specifications, *Knowledge Acquisition Journal*, academic Press, (1993).
- [GRF et al, 95] : GRUNINGER M. & FOX M. S, Methodology for the design and evaluation of ontologies, in *Proceedings of the Workshop on Basic Ontological Issues on Knowledge Sharing*, (1995)
- [GUA et al, 95] : Guarino N., et Giaretta P, Ontologies and Knowledge Bases: Towards a Terminological Clarification. In *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing*, Mars N. J. I., Amsterdam: IOS Press, (1995).
- [BOR, 97] : Borst W. N, Construction of Engineering Ontologies. Center for Telematica and Information Technology, University of Twente, Enschede, NL, (1997).
- [GUA, 97]: Nicola Guarino, Semantic Matching: Formal Ontological Distinctions for Information Organization, Extraction, and Integration. In *Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology*, S C I E 1997, M. T. Paziienza (Eds.), Springer Verlag, pp.139-170- (1997).
- [STE et all, 97]: Steve G, Gangemi A, Pisanelli DM, Integrating Medical Terminologies with ONIONS Methodology in Kangassalo H, Charrel JP (eds.) *Information Modelling and Knowledge Bases VIII*, Amsterdam, IOS Press, (1997)
- [STU et al, 98] : Studer R., Benjamins V. R. et Fensel D, Knowledge engineering: Principles and Methods, *Data Knowledge Engineering*, vol. 25, no 1-2, (1998).

Références bibliographiques

- [BLA et al, 98] : BLAZQUEZ M., FERNANDEZ M., GARCIA-PINAR J. M. & GOMEZ-PEREZ A, Building Ontologies at the Knowledge Level using the Ontology Design Environment, in *Proceedings of the Banff Workshop on Knowledge Acquisition for Knowledge-based System*, (1998).
- [GOM, 99] : Gomez Pérez A, Développements récents en matière de conception, de maintenance et d'utilisation d'ontologies. 3èmes rencontres *Terminologie et intelligence artificielle TIA*, (1999).
- [BAC, 00] : BACHIMONT B, Engagement sémantique et engagement ontologique conception et réalisation d'ontologies en ingénierie des connaissances. *Ingénierie des connaissances. Évolution Récentes et nouveaux défis* Eyrolles, 305-323. Paris-2000.
- [STA et al, 00]: S. Staab, A. Maedche, Axioms are objects too: Ontology engineering beyond the modeling of concepts and relations, Research report 399, Institute AIFB, Karlsruhe, (2000).
- [FRE, 01] : Alain FERGANI, Ontology dynamics for Semantic Web: the MOMIS approach, (2001/2002).
- [FRE, 02] : FRÉDÉRIC Fürst, Ingénierie des Connaissances, Rapport de recherche, Institut de recherche en informatique de Nates – France- (Octobre 2002)
- [MAE, 02] MAEDCHE A., MOTIK B., SILVA N. & VOLZ R., MAFRA: a mapping framework for distributed ontologies, in *Proceedings of the International Conference EKAW'2002*, Springer LNAI 2473, pages 235-250, (2002).
- [CHA et al, 04]: Charlet J., Bachimont B. et Troncy R, Ontologies pour le Web Sémantique. In *Le Web sémantique*, CHARLET J., LAUBLET P. et REYNAUD C. (Ed.), Hors série de la Revue *Information - Interaction - Intelligence* (I3), 4(1), Cépaduès,

Références bibliographiques

Toulouse, (2004).

- [VAL, 04]:** VALERY PSYCHE, Proposition d'une méthode d'ingénierie ontologique pour les EIAH : application aux systèmes auteurs, Programme de doctorat en informatique, Université du Québec à Montréal Canada- (Mai 2004)
- [NAT, 05] :** Nathalie HERNANDEZ, Ontologies de domaine pour la modélisation du contexte en recherche d'information, Thèse de doctorat, université Paul Sabatier de Toulouse, (2005).
- [MOS, 05] :** Sihem Mostefai, Fusion d'ontologies dans une optique PLM, (2005).
- [GAY, 06] :** Gayo Diallo, Une Architecture à Base d'Ontologies pour la Gestion Unifiée des Données Structurées et non Structurées, THESE présentée pour obtenir le grade de Docteur de l'Université Joseph Fourier – Grenoble I (Spécialité : Informatique), (11 Décembre 2006)
- [ELI, 07] :** Eliane Consola, Qu'est-ce qu'une ontologie, Article pour le journal du net (03/04/2007)
- [MEL, 07] :** Naçima MELLAL, Réalisation de l'interopérabilité sémantique des systèmes, basée sur les ontologies et les flux d'information, THÈSE pour obtenir le grade de DOCTEUR DE L'UNIVERSITÉ DE SAVOIE, Discipline : Informatique, (19 Décembre 2007).
- [MAI et al, 07] :** Nora Maiz, Omar Boussaid, Fadila Bentayel, Fusion automatique des ontologies par classification hiérarchique pour la conception d'un entrepôt de données, (3 juillet 2007).
- [GIL, 08] :** Gilles Kassel, Exposé sur Ingénierie Ontologique (Concepts, méthodes et outils), (2008)
- [GHO, 08] :** Mme Ghomari Leïla (Née Zemmouchi), Alignement d'ontologies de domaine : Etude comparative syntaxique

Références bibliographiques

versus sémantique, cas d'application COMA++ VS OWL-CTXMatch, Mémoire pour l'obtention du diplôme de Magister en Informatique, à l'Ecole Supérieur d'Informatique (E.S.I) Alger, Algérie(2008-2009)

[ELB, 09] : M Abdeltif ELBYED, ROMIE, une approche d'alignement d'ontologies à base d'instances, Thèse présentée pour l'obtention du grade de Docteur de l'INSTITUT NATIONAL DES TELECOMMUNICATIONS, (*16 Octobre 2009*)

Références Webographique

- [1] : <http://www.ista-ntic.com/v2/programmation/Java/Generalites/Generalites.htm>
- [2] : <http://www.techno-science.net>
- [3] : <http://www.eclipse.org>
- [4] : <http://protege.stanford.edu/index.shtml>
- [5] : <http://jena.sourceforge.net/>
- [6] : <http://www.jfree.org/jfreechart/>
- [7] : <http://owl2prefuse.sourceforge.net/documentation.php>
- [8] : <http://www.jtattoo.net>
- [9] : <http://lyle.smu.edu/~tspell/jaws/index.html>
- [10] : <http://tahe.ftp-developpez.com/fichiers-archive/baseswebmvcjava.pdf>

