

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE SAAD DAHLEB DE BLIDA
FACULTE DES SCIENCES
DEPARTEMENT D'INFORMATIQUE



MEMOIRE DE FIN D'ETUDES

Pour l'obtention

D'un diplôme de master en informatique.

Option : Génie des systèmes informatiques

THÈME :

Affectation automatique des projets de recherche à une ontologie organisationnelle d'une division de recherche

Promotrice :

Mme FAREH.

Réalisé par :

Madani Fella Aida.

Encadreur :

Mme Guemadi Djalila née Keddari.

Mme Mellah Hakima.

Zouaoui Imene.

Soutenue le: 26/6/2016, devant le jury composé de :

Nom présidente du jury Mlle Boustia.

Nom examinateur M. Nahel.

2015/2016

Remerciements

Nous remercions, tout d'abord, Dieu tout puissant de nous avoir donné le courage, la force et la patience d'achever ce modeste travail.

Nous tenons à exprimer notre profonde gratitude à nos deux encadreur Mme Guemadi Djalila née keddari et Mme Mellah Hakima ingénieurs en informatique au centre de recherche de l'information scientifique et technique (CERIST) de nous avoir encadré avec un intérêt constant et une grande compétence ainsi pour l'intérêt qu'elles ont bien voulu porter à notre travail.

Nous exprimons aussi nos reconnaissances à Mme Fareh notre promotrice pour sa disponibilité, son soutien, ses conseils et ces encouragements qui ont permis de mener à bien ce travail.

Que Messieurs les jurys, trouvent ici l'expression de nos remerciements les plus sincères d'avoir accepté d'examiner ce mémoire.

Enfin on remercie toutes les personnes qui ont participé de manière directe ou indirecte à la concrétisation de ce travail.

Dédicace

Je dédie ce modeste travail, avec une énorme joie et un infini plaisir, aux deux Merveilleuses personnes qui m'ont aidé et guidé vers la voie de la réussite :

A mon père «Rachid», pour son attention et son sacrifice.

A ma mère « Nacera », océan de tendresse et fleuve de gentillesse.

Que DIEU m'aide à les honorer et exprimer ma profonde reconnaissance pour tout ce qu'ils m'ont offert d'amour, de soutien et des encouragements.

Je le dédie aussi à :

Mon frère (Zoheir) et mes chères sœurs (Hayet, Hassina et Imene).

Mes amis (Imene, Sarah, Mohamed Zeta).

Et à tous ceux qui nous ont aidés à l'aboutissement de ce modeste travail.

Zouaoui Imene

Dédicace

C'est avec toute l'ardeur de mes sentiments et mes sincères mots que je dédie cet humble travail :

A mes chères parents, pour leur amour affection et tous leurs sacrifices, la détermination ainsi que l'abnégation qu'ils ont su m'inculquer durant toutes mes années d'études.

A mes chers frères, source d'inspiration, de noblesse et de fierté.

Ainsi qu'à toute ma famille et à l'ensemble des personnes qui ont contribué de près ou de loin à ce travail.

Sans oublier tous mes amis de la faculté.

Que dieu vous garde.

Madani Fella Aida

Résumé

L'avènement de l'informatique et l'apparition de nouveaux formalismes de traitement nous ont permis d'automatiser plusieurs tâches pénibles et fastidieuses. La machine s'est montrée nettement supérieure à l'homme dans les domaines de calcul. Actuellement, beaucoup d'outils existent pour résoudre l'hétérogénéité des données, et chercher la similarité que ce soit au niveau syntaxique ou sémantique.

La division DSISM du CERIST a voulu automatiser la procédure d'affectation des projets de recherche aux thèmes de recherche correspondant qui était jusqu'à présent manuelle.

Ce travail est une proposition d'un système d'affectation automatique de projets de recherches aux thèmes de recherche d'une ontologie de domaine de la division DSISM du CERIST, en utilisant une combinaison de techniques et méthodes de mesures de similarité terminologiques, structurelles et lexicales pour effectuer les différentes correspondances et relations entre les projets de recherche et leurs thèmes de recherche correspondants.

Mots clé : Ontologie, mesures de similarité, affectation automatique, sémantique.

Abstract

The advent of computer science and the appearance of new formalism of treatment allowed us to automate many difficult and tedious tasks. The machine has proved clearly superior to man in the computing fields. Currently, a lot of tools exist to solve the heterogeneity of the data, and look for the similarity whatsoever to the syntactic or semantic level.

The DSISM division of CHRIST wanted to automate the assignment procedure of research projects to corresponding research topics which until this moment it is done manually.

This work is a proposal of a system of automatic assignment of research projects to research topics for a domain ontology of DSISM division of CERIST, using a combination of techniques and methods of terminology similarity measures, structural and lexical to perform the different connections and relationships between research projects and corresponding research topics.

Keywords: Ontology, similarity measures, automatic assignment, semantic.

ملخص

سمح ظهور الإعلام الآلي و الطرق الجديدة لمعالجة المعلومات التشغيل الأوتوماتيكي للعديد من المهام الشاقة و التي تستغرق وقتا طويلا.

وقد أثبتت الآلة تفوقا واضحا على الإنسان في مجالات الحوسبة. حاليا، العديد من الأدوات موجودة لحل مشكل عدم تجانس البيانات، والبحث عن التشابه على المستوى النحوي أو الدلالي.

ارادت وحدة CERIST ل DSISM جعل طريقة بعث مشاريع البحث اوتوماتيكية نحو مواضيع البحث الخاصة بها التي لحد الان تجري بطريقة يدوية.

هذا العمل هو اقتراح لنظام البعث الأوتوماتيكي لمشاريع البعث نحو مواضيع البعث الخاصة بأنطولوجيا المجالات , وهذا لوحدة CERIST ل DSISM باستعمال مزيج من تقنيات و أساليب معيار التشابه ذو مقياس المصطلحات، ذو

مقياس البنية و ذو مقياس

الدلالة للبحث عن مختلف التطابقات و العلاقات بين مشاريع البحث و المواضيع الموافقة لها.

الكلمات المفتاحية : انطولوجيا , معيار التشابه , البحث الأوتوماتيكي , الدلالة.

Table des matières

INTRODUCTION GENERALE2

Partie I : ETAT DE L'ART

Chapitre I : Etat de l'art sur les ontologies

Introduction.....6

2. Définition d'une ontologie 6

3. Buts des ontologies 7

4. Types d'ontologies.....8

4.1. Ontologie de domaine8

4.2. Ontologie d'application 9

4.3. Ontologie générique (Core ontologies ou Meta-ontologies)9

4.4. Ontologie de représentation des connaissances..... 9

4.5. Ontologie de tâche.....9

5. Les composants d'une ontologie 9

5.1. Concepts.....9

5.2. Les attributs..... 10

5.3. Les relations entre concepts 10

5.4. Axiomes..... 10

5.5. Les instances (individus) 10

6. Cycle de vie d'une ontologie..... 11

7. Langages et outils des ontologies 12

7.1. Langages des ontologies..... 12

7.2. Outils des ontologies : 13

7.2. 1. PROTÉGÉ 14

7.2. 2. WebODE..... 14

7.2. 3. OntoEdit 14

8. Conclusion 14

Chapitre II: Etat de l'art sur les mesures de similarités

1.Introduction..... 16

2. Alignement d'ontologie..... 16

3.Mesures de similarité..... 16

| | |
|---|----|
| 3.1. Définition de la similarité..... | 17 |
| 3.2. Méthodes de calcul de similarité | 17 |
| 3.2.1. Les mesures simples | 18 |
| 3.2.1.1. Les techniques terminologiques | 18 |
| 3.2.1.1.1. Approche syntaxique (Les méthodes se basant sur des chaînes de caractères) . | 18 |
| 3.2.1.1.2. Approche lexicale(Les méthodes linguistiques) | 20 |
| 3.2.1.2. Les techniques structurelles..... | 22 |
| 3.2.1.3. Les techniques extensionnelles | 23 |
| 3.2.1.4. Les techniques sémantiques..... | 24 |
| 3.2.2. Les techniques combinées | 24 |
| 3.2.2.1. L'approche hybride..... | 24 |
| 3.2.2.2. L'approche composite | 25 |
| Conclusion | 25 |

Partie II : CONCEPTION ET IMPLEMENTATION

Chapitre III : Conception

| | |
|--|----|
| 1. Introduction | 27 |
| 2. Présentation de la démarche utilisée..... | 27 |
| 2.1. Spécification des besoins | 27 |
| 2.1.1. Les diagrammes de cas d'utilisation | 28 |
| 2.1.1.1. Diagramme de cas d'utilisation global..... | 28 |
| 2.1.1.2. Diagramme de cas d'utilisation « Gestion chercheur »..... | 29 |
| 2.1.1.3. Diagramme de cas d'utilisation : « Gestion Projet »..... | 29 |
| 2.1.1.4. Diagramme de cas d'utilisation : «Affectation »..... | 30 |
| 2.2. Analyse des besoins | 30 |
| 2.2.1. Les diagrammes de séquences | 31 |
| 2.2.1.1. Diagramme de séquence relatif au cas d'utilisation « authentification » | 31 |
| 2.2.1.2. Diagramme de séquence relatif au cas d'utilisation « Ajout Projet »..... | 32 |
| 2.2.1.3. Diagramme de séquence relatif au cas d'utilisation «Affectation »..... | 33 |
| 3. Conception du système..... | 34 |
| 3.1. Conception générale | 34 |
| 3.1.1. Diagramme de paquetage | 34 |
| 3.2. Conception détaillée | 35 |
| 3.2.1. Architecture détaillé du système | 35 |
| 1. La similarité terminologique | 39 |

| | |
|---|-----------|
| Il existe plusieurs méthodes de calcul de la mesure de similarité terminologique qui s'appuie sur l'approche syntaxique et lexicale. | 39 |
| 1.1 L'approche syntaxique | 39 |
| 1.2. Approche lexicale | 40 |
| 2. La similarité structurelle | 40 |
| 3. La similarité sémantique (globale) | 40 |
| 3.2. Moyenne pondérée | 41 |
| 4. Le procédé de calcul de la similarité | 42 |
| 5. Discussion des résultats obtenus | 42 |
| 6. Conclusion | 44 |

Chapitre VI : Implémentation

| | |
|---|-----------|
| 1.Introduction | 46 |
| Partie I : Environnement de développement | 46 |
| 1. Langages utilisé | 46 |
| 1.1. JAVA | 46 |
| 1.2. SQL | 46 |
| 1.3. XML | 47 |
| 2. Outils | 47 |
| 2.1. Netbeans | 47 |
| 2.2. Protégé | 47 |
| 2.3 Les APIs | 48 |
| 3. Visualisation de l'ontologie dans protégé | 48 |
| Partie II : Interface utilisateur | 50 |
| 1. Authentification : | 50 |
| 2. Ajout des individus et relations | 51 |
| 2.1. L'ajout d'un individu | 51 |
| 2.2. Ajout des relations | 52 |
| 3. Affectation | 53 |
| 4. La sérialisation | 53 |
| 5. Recherche des informations | 54 |
| 5.1. Définition de SPARQL | 54 |
| 5.2. Caractéristique et construction de requête SPARQL | 54 |
| 5.3. Fonctionnalité recherche de l'interface utilisateur | 55 |
| 6. Suppression de l'individu | 57 |
| 7. Tests de validation | 57 |
| 7.1. Test 1: | 57 |

| | |
|---------------------------|----|
| 7.2. Test 2..... | 59 |
| 8. Conclusion..... | 61 |
| Conclusion Générale | 63 |

La liste des figures

| | |
|---|----|
| Figure 1: Utilisation des Ontologies. | 9 |
| Figure 2: Cycle de vie d'une ontologie [8]. | 12 |
| Figure 3: Schéma d'un triplé RDF. | 14 |
| Figure 4: Mesures de calcul de similarités. | 18 |
| Figure 5: Un fragment de la hiérarchie de WordNet pour mettre en évidence les liens entre les termes : «Author, writer, creator, illustrator & person». | 22 |
| Figure 6: mesure Wu et Palmer appliqué aux concepts C1 et C2. | 24 |
| Figure 8: Diagramme de cas d'utilisation : « Gestion chercheur ». | 31 |
| Figure 9: Diagramme de cas d'utilisation : « Gestion Projet ». | 31 |
| Figure 10: Diagramme de cas d'utilisation : «Affectation ». | 32 |
| Figure 11: diagramme de séquence « authentification ». | 33 |
| Figure 12: Diagramme de séquence de cas d'utilisation « Ajout Projet ». | 34 |
| Figure 13 : Diagramme de séquence relatif au cas d'utilisation « Affectation ». | 35 |
| Figure 14: diagramme de paquetage. | 36 |
| Figure 15: Schéma de fonctionnement du système. | 37 |
| Figure 16: Arborescence des thèmes dans ACM. | 39 |
| Figure 17: Schéma de combinaison des Méthodes. | 41 |
| Figure 18: Visualisation de l'ontologie. | 51 |
| Figure 19: Interface d'authentification | 52 |
| Figure 20: Menu de l'interface utilisateur. | 52 |
| Figure 21: Ajout d'un individu de « Projet_de_Recherche ». | 53 |
| Figure 22: Algorithme d'ajout d'un individu avec JENA. | 54 |
| Figure 23: Ajout d'une relation. | 54 |
| Figure 24: Affectation automatique du projet de recherche. | 55 |
| Figure 25: Sauvegarde des changements. | 56 |
| Figure 26: exemple de requête SPARQL. | 57 |
| Figure 27: Fenêtre recherche. | 57 |
| Figure 28: Chercher les thèmes d'un projet. | 58 |

| | |
|---|----|
| Figure 29: Résultat de la recherche | 58 |
| Figure 30: Supprimer un individu | 59 |
| Figure 31: Confirmation de Suppression sur l'ontologie | 59 |
| Figure 32: Résultats du calcul de "Test 1" | 60 |
| Figure 33 : Résultat de l'affectation de "Test 1" | 61 |
| Figure 34: Résultats du calcul de "Test 2" | 62 |
| Figure 35: Résultat e l'affectation de "Test 2" | 63 |

La liste des tableaux

| | |
|--|----|
| Tableau 1: description de cas d'utilisation global..... | 30 |
| Tableau 2 : description de cas d'utilisation « Gestion chercheur » | 31 |
| Tableau 3 : description de cas d'utilisation « Gestion Projet »..... | 32 |
| Tableau 4: description de cas d'utilisation « Affectation » | 32 |
| Tableau 5: description de cas d'utilisation « Authentification » | 34 |
| Tableau 6: description de cas d'utilisation «Ajout Projet »..... | 35 |
| Tableau 7: description de cas d'utilisation «Affectation » | 35 |
| Tableau 8: Explication des relations | 35 |

Les acronymes:

OWL: Web Ontology Language

OIL: ontology interchange language

DAML: Darpa Agent Markup Language

ACM :Association for Computing Machinery

CCS: Computer and Communications Security

*XML: **Extended** Markup Language*

*RDF: **Ressource** Description Framework*

*RDFS: **Ressource** Description Framework Schema*

*URL: **Uniform** Resource Locator*

*URI : **Uniform** Resource Identifier*

Introduction Générale

INTRODUCTION GENERALE

Contexte de travail

La Recherche d'informations peut être définie comme une activité dont la finalité est de délivrer des granules documentaires aux utilisateurs, en fonction de leur besoin en qualité d'informations. La spécificité des informations les rend difficilement exploitables et leur évolution constante rend complexes les techniques de recherche d'informations. Le web sémantique est apparu comme réponse au besoin de pallier les problèmes de la recherche d'information comme le confirme cette citation : « Dans le web de demain, pour traiter une masse de données sans cesse croissante, on devra avoir recours à la représentation des connaissances pour attribuer un sens aux informations. Cette représentation du sens, manipulable par des programmes informatiques, nous aidera à sélectionner, à trier et à traduire les informations qui nous sont utiles. La notion d'ontologie, instrument du web sémantique, y est vue comme un type de modèles particulier, une théorie d'un domaine de connaissances, et l'ingénierie ontologique, comme une forme d'ingénierie des connaissances utile pour construire le web sémantique.»[1]

Les ontologies issues de l'ingénierie des connaissances constituent une des voies les plus prometteuses pour la modélisation et l'opérationnalisation des systèmes notionnels. Elles intéressent la terminologie à plus d'un titre. Une ontologie est l'ensemble structuré des termes et concepts représentant le sens d'un champ d'informations. L'ontologie constitue en soi un modèle de données représentatif d'un ensemble de concepts dans un domaine, ainsi que des relations entre ces concepts. Elle est employée pour raisonner à propos des objets du domaine concerné.

Problématique

Au sein de la division DSISM (Division Système d'Information et Système Multimédia) du CERIST une ontologie de domaine a été construite pour faciliter la recherche, le partage et la réutilisation de l'information scientifique et technique. Notre travail porte sur deux concepts essentiels de cette ontologie qui permettent essentiellement de représenter les différents thèmes et projet de recherches et qui sont respectivement thèmes de recherches et projets de recherches. Un projet de recherche concerne au moins un thème de recherche tel qu'il doit lui être affecté, l'affectation permet de relier un projet à un thème par la relation «Est affecté à » cette relation doit être faite manuellement par l'utilisateur de l'ontologie.

Ce mémoire présente une étude élaborée pour la mise en œuvre d'une fonctionnalité permettant l'affectation automatique de tout projet de recherches à l'ontologie, et cela une fois le projet défini et conformément au mot clés le caractérisant. Une fonctionnalité répondant aux raisons d'avoir réfléchi à mettre en relation le projet de recherche avec les thématiques de l'ontologie. Or c'est ainsi que l'on peut se demander comment réaliser cette dernière.

Objectif de travail

Notre étude vise à proposer une solution à la problématique d'automatisation de l'affectation manuelle des projets de recherche à l'ontologie, cette tâche implique une autre tâche qui est la sélection des thèmes correspondant au projet. Cette dernière demande un effort humain de l'utilisateur du système et peut même être amplifié du fait qu'un chercheur qui utilise le système peut mobiliser d'autre chercheur voir même l'expert de la division pour effectuer cette sélection. Donc c'est une tâche lourde qui donne lieu à des contraintes coûteuses en facteur temps et argent.

Notre mémoire est structurée comme suit :

La première partie : Etat de l'art

Chapitre 1 Ontologie: Ce chapitre propose une généralité sur les ontologies. Nous présentons les ontologies avec leurs caractéristiques, leurs constructions et leurs méthodes d'application, ainsi que les domaines d'application de ces ontologies. Nous allons décrire quelques langages de représentation des ontologies.

Chapitre 2 Mesure de similarités: Ce chapitre décrit les principales mesures de similarité. Nous présentons les classifications des techniques de calcul de la similarité ainsi que quelques exemples des méthodes les plus utilisées.

La deuxième partie : Conception et Implémentation

Chapitre 3 Conception : Ce chapitre contient la spécification et l'analyse des Besoins pour répondre aux besoins et exigences de la problématique citée ci-dessus, En utilisant des diagrammes UML et le schéma de fonctionnement de notre système qui décrit les différents processus dans notre système.

Chapitre 5 Implémentation : La réalisation de notre système ainsi que l'environnement de développement et des tests effectués sur le système sont présentés dans ce chapitre permettent de vérifier les résultats de l'implémentation

Notre mémoire se termine par une conclusion qui synthétise nos principales contributions et donne quelques perspectives à notre travail.

Chapitre I :
L'état de l'art sur les ontologies

Introduction

Le présent chapitre est une introduction aux ontologies, nous introduisons les concepts de base des ontologies en passant d'abord par les notions essentielles de la connaissance.

Nous énonçons ensuite quelques principes fondamentaux, sur l'utilisation des ontologies, types d'ontologies, les grands modèles des ontologies et les langages de représentation des ontologies les plus connus et les plus utilisés.

2. Définition d'une ontologie

Le mot ontologie vient du grec, par étymologie onto "ontos" veut dire ce qui existe ou l'existant et logo "logos" signifie le discours, l'étude, l'univers ou encore la science.

En théorie de l'existence qui est une branche de la philosophie, une ontologie est l'étude des propriétés générales de ce qui existe (l'étude de l'être en tant qu'être) [2].

L'informatique (science de l'information et intelligence artificielle) définit une ontologie comme étant un modèle de donnée qui permet la représentation et l'organisation des connaissances par un ensemble de termes, de concepts et de relations entre ces concepts, d'un domaine donné.

D'après Gruber (1993) : « Une ontologie est une «spécification formelle et explicite d'une conceptualisation partagée» [3].

Où :

Spécification formelle : compréhensible par une machine.

Spécification explicite : les concepts, relations, fonctions, axiomes, propriétés et les restrictions sont définis de façon déclarative.

Conceptualisation : c'est un modèle abstrait représentant les différents concepts d'un domaine.

Partagée : les connaissances représentées sont partagées par une communauté.

3. Buts des ontologies

Les ontologies sont développées pour [4]:

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

_ Intégration : dans une application « entrepôt de données », une ontologie peut jouer le rôle d'un schéma conceptuel commun reliant entre elles plusieurs sources d'information hétérogènes.

Les ontologies sont utilisées dans plusieurs domaines à savoir les systèmes d'information, l'intelligence artificielle, le Web sémantique, le génie logiciel, le E-commerce ou encore l'architecture de l'information comme une forme de représentation de la connaissance d'un domaine comme le représente la figure suivante :

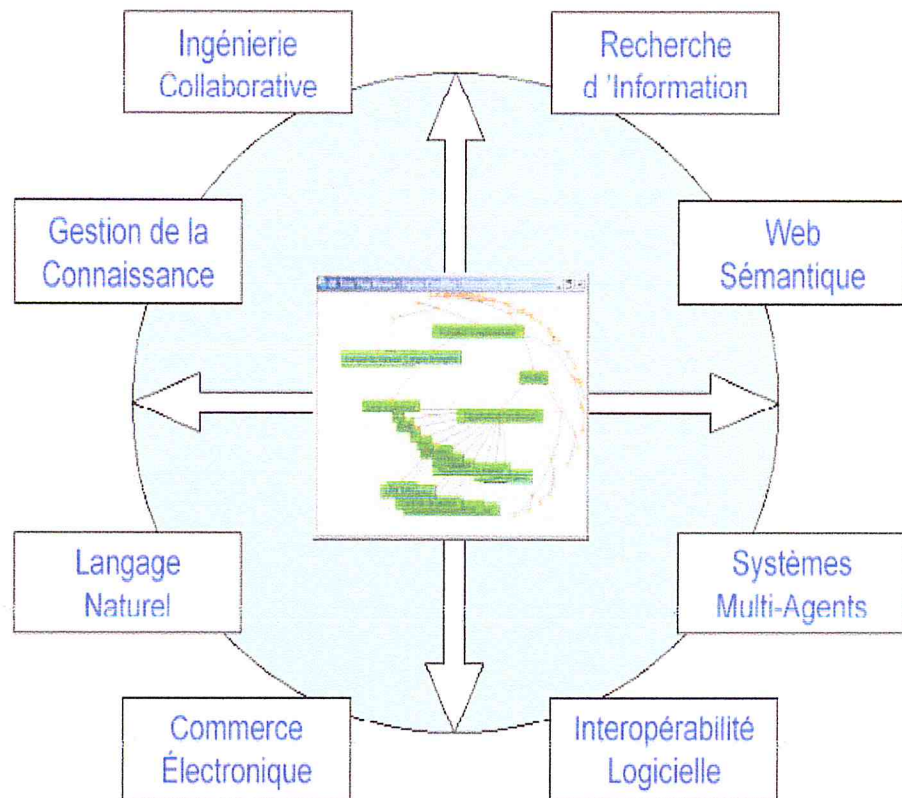


Figure 1: Utilisation des Ontologies.

4. Types d'ontologies

On distingue cinq types d'ontologies [5]:

4.1. Ontologie de domaine

Elles sont appelées de la sorte parce qu'elles expriment des conceptualisations spécifiques à un domaine. Elles rendent compte du vocabulaire d'un domaine spécifique à travers des concepts et des relations qui modélisent les principales activités, les théories et les principes de base du domaine en question. Elles sont réutilisables pour plusieurs applications concernant le domaine pour lequel elles ont été créées, car elles offrent une indépendance dans la manipulation des connaissances qu'elles comportent.

4.2. Ontologie d'application

Ce sont les ontologies les plus spécifiques, elles contiennent les connaissances requises pour une application particulière et ne sont pas réutilisables.

4.3. Ontologie générique (Core ontologies ou Meta-ontologies)

Elles expriment des conceptualisations valables dans différents domaines de valeur génériques (relativement générale) comme les notions d'objets, de propriété, de valeur, d'état, ou encore de temporalité. Théoriquement, ces ontologies doivent pouvoir être reliées au sommet des ontologies de domaines.

4.4. Ontologie de représentation des connaissances

Elles regroupent les concepts (primitives de représentation) impliqués dans la formalisation des connaissances [6].

4.5. Ontologie de tâche

Ce type d'ontologie est utilisé pour conceptualiser des tâches spécifiques dans les systèmes, telles que la tâche de diagnostic, de planification, de conception, de configuration, soit tout ce qui concerne la résolution de problèmes.

5. Les composants d'une ontologie

Une ontologie comporte plusieurs composants à savoir [7] :

5.1. Concepts

Un concept est la description d'un ou plusieurs objets distincts étant défini comme une entité qui comporte trois éléments :

Terme : qui définit le terme en langage naturel.

Notion (signification ou intension): l'ensemble des propriétés du concept.

Extension (réalisation) : l'ensemble des instances dénotées par le concept.

Exemple : citons le concept de voiture :

- Son terme est le nom voiture, bagnole, automobile.

- Sa notion est véhicule de transport, automobile léger.
- L'extension c'est le nom des voiture comme : Mercedes, Renault.

5.2. Les attributs

Ce sont de simples propriétés décrivant le concept.

Exemple : nom, prénom, fonction, âge,...etc.

5.3. Les relations entre concepts

Ce sont les relations (associations pertinentes) qui lient les instances de concepts comme suit :

Terme : c'est la représentation linguistique.

Intension : ensemble des propriétés et des attributs de la relation.

Extension : ensemble de réalisation effective d'une relation entre concepts.

Signature : c'est l'ensemble des instances de concept liées par la relation.

- Exemple de relation : fabriqué par, appartient à, composé de.

La voiture est fabriquée par l'humain.

5.4. Axiomes

Les axiomes sont des expressions qui sont toujours vraies, ils sont incluse dans une ontologie pour exprimer des restrictions, définir la signification des composants, ou encore définir des arguments d'une relation.

Exemple : Si deux étudiants font partie de la même section et du même groupe alors ils ont les mêmes enseignants dans chaque module.

5.5. Les instances (individus)

Elles consistent en la définition extensionnelle de l'ontologie ; et sont utilisées pour représenter des éléments dans un domaine

6. Cycle de vie d'une ontologie

Les ontologies sont utilisées comme des composants logiciels dans les Systèmes informatiques pour cela, leur développement suit les principes du génie logiciel comme suit [8]:

Evaluation des besoins : une ontologie doit répondre à toutes les questions pour lesquelles elle doit être créée.

Construction : l'ontologie doit être construite en suivant la méthodologie la plus adaptée.

Diffusion : c'est la disponibilité de l'ontologie aux utilisateurs.

Utilisation : l'étape où l'utilisateur utilise l'ontologie.

Validation : c'est l'étape <<construire le bon système>>, elle permet de voir si elle est bien construite et surtout voir si elle répond au besoin. En d'autres termes, vérifier si elle est conforme à la sémantique du domaine de connaissances et la conformité de ce dernier.

Maintenance : cette étape permet de détecter les fautes et les erreurs commises pour les corriger. Cette étape regroupe les étapes d'évaluation et d'évolution du cycle de vie.

La figure ci-dessous représente le cycle de vie d'une ontologie

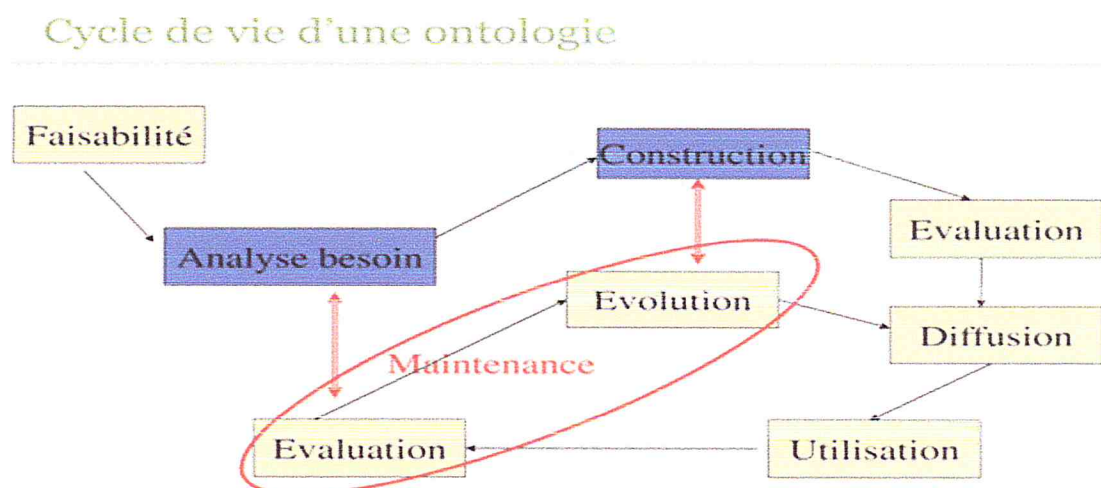


Figure 2: Cycle de vie d'une ontologie [8].

7. Langages et outils des ontologies

7.1. Langages des ontologies

Actuellement les logiciels informatiques ne cessent d'évoluer cependant il existe différents langages de représentation d'ontologie tel que : XML, RDF et OWL et qui constituent les trois couches de base du Web Sémantique:

XML

XML (eXtensibleMarkupLanguage) est un langage informatique de balisage générique. Il est recommandé par le W3C pour exprimer des langages de balisages spécifiques, tels que RDF ou OWL. Un document XML se présente sous la forme de données « taguées » par un ensemble de balises, chacune pouvant comporter des attributs et des valeurs.

Son objectif initial est de faciliter l'échange automatisé de contenus entre systèmes d'information hétérogènes, on parle alors d'interopérabilité.

RDF

RDF (Resource Description Framework) est un modèle standard pour l'échange de données sur le web, développé par le W3C. Tout document RDF est composé d'un ensemble de triplets (sujet, prédicat, objet). Un ensemble de tels triplets est appelé un graphe RDF.

RDF utilise la syntaxe XML [9] pour décrire les ressources du web tels que chaque ressource est pourvue d'un Identifiant de ressource uniforme URI (Uniform Resource Identifier) et basée sur les triplets :

- Sujet** : une entité d'informations pouvant être référencée par un identificateur. Cet identificateur doit être une URI.
- prédicat** : l'attribut ou la relation utilisée pour décrire un sujet.
- objet** : la valeur d'un prédicat associée à un sujet spécifique.

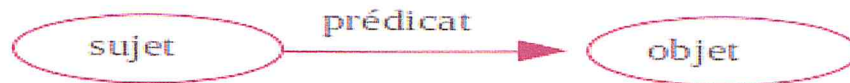


Figure 3: Schéma d'un triplé RDF.

RDFS

RDF Schéma a été construit par W3C comme une extension du langage RDF. Il permet notamment de déclarer les propriétés des ressources ainsi que le type des ressources. La combinaison de RDF et RDF Schéma est connu sous le nom RDF(S). Bien que relativement limités dans la mesure où ils ne sont pas très expressifs, les langages RDF(S) peuvent cependant spécifier des concepts, des taxonomies et des relations binaires.

OWL

OWL (ontologie web langage) est une recommandation du W3. Il est conçu pour être utilisé par les applications qui ont besoin de traiter le contenu des informations au lieu de simplement présenter les informations à l'utilisateur. OWL facilite une grande interprétation du contenu du web que celui fournit par XML, RDF ou RDF Schéma (RDFS) en fournissant un vocabulaire riche avec une sémantique formelle.

OWL est subdivisé en trois sous langages [10]: OWL Lite, OWL DL et OWL Full.

OWL Lite est destiné à être utilisé dans les situations où seulement une simple hiérarchie de classe et des contraintes simples sont nécessaires. OWL DL est beaucoup plus expressives qu'OWL Lite, il permet ainsi d'automatiser le calcul de la classification hiérarchique et la vérification des incohérences dans une ontologie OWL DL. OWL Full est beaucoup plus expressifs qu'OWL DL mais il ne permet pas d'automatiser le raisonnement au sein de l'ontologie.

7.2. Outils des ontologies :

Voici quelques éditeurs d'ontologies les plus utilisés :

7.2. 1. PROTÉGÉ

Protégé est un système auteur pour la création d'ontologies. Il a été créé à l'université Sanford et il est très populaire dans le domaine du Web sémantique et au niveau de la recherche en informatique. C'est une plate-forme Open Source autonome, qui fournit un environnement graphique permettant l'édition, la visualisation et le contrôle de l'ontologie. L'architecture logicielle de protégé 4.3 permet le rajout de nouvelles fonctionnalités via des plugins parmi ces nouvelles fonctionnalités les possibilités d'importer et d'exporter des ontologies construites dans d'autre langages [11].

Toutes ces caractéristiques ont participé à son succès et le rendent l'éditeur d'ontologie jouissant de la plus grande renommée à l'heure actuelle, servant de référence pour une importante communauté d'utilisateurs.

7.2. 2. WebODE

Développé par l'université polytechnique de Madrid. Composé de plusieurs modules : un éditeur d'ontologie qui intègre la plupart des services nécessaires à la construction d'ontologies (édition, navigation, comparaison, fusion, raisonnement...), un système de gestion des connaissances à base ontologique, un outil pour annoter les ressources du web et un éditeur de services pour le Web sémantique.

7.2. 3. OntoEdit

Cet éditeur est payant contrairement à Protégé .il permet aussi la hiérarchie des concepts, l'expression des axiomes et l'exportation des ontologies sous divers langages [12].

8. Conclusion

Dans ce chapitre nous avons présenté les ontologies avec leurs caractéristiques, ainsi que les domaines d'application de ces ontologies. Nous avons décrit quelques langages de représentation des ontologies et quelques outils de construction

Le prochain chapitre est consacré aux mesures de calcul de similarité.

Chapitre II

L'état de l'art sur les mesures de

similarité

Introduction

Intégrer de nouvelles entités dans une ontologie revient à trouver les relations entre les différentes entités de l'ontologie et celles à intégrer qui se fait en évaluant leurs similarités. Cela définit une grande partie de l'alignement des ontologies.

Pour déterminer les relations entre les entités de l'ontologie nous présentons ici des méthodes de base pour évaluer la similarité qui sont utilisées très souvent dans l'alignement.

Ces mêmes mesures de similarité peuvent être utilisées en dehors de l'alignement des ontologies, dans des applications qui utilisent la similarité entre termes ou concepts.

2. Alignement d'ontologie

L'alignement d'ontologie est défini comme étant le résultat d'un matching d'ontologies. Le matching d'ontologies vise à trouver des correspondances entre les entités sémantiquement liées de différentes ontologies. Ces correspondances peuvent reposer sur l'équivalence ainsi que sur d'autres relations, comme la conséquence, la subsomption, ou la disjonction.

Les entités de l'ontologie désignent généralement les entités nommées des ontologies, tels que les classes, les propriétés ou individus. Cependant, ces entités peuvent également être des expressions plus complexes, tels que des formules, des définitions de concept, des requêtes ou expressions de construction de terme.

L'alignement ainsi exprime avec divers degrés de précision les relations entre les ontologies considérées.

En outre, les approches courantes d'alignement d'ontologies [13] [14] sont basées sur les mesures de similarité entre chaînes de caractères et des structures composites.

3. Mesures de similarité

Une mesure de similarité est généralement une fonction de quantification exprimant un rapport de comparaison de deux objets de mêmes types en fonction de leurs points de ressemblances et dissemblances.

Les valeurs de similarité sont souvent normalisées pour pouvoir être combinées dans des formules plus complexes. La normalisation impose que les valeurs appartiennent à

l'intervalle $[0,1]$. Si la valeur de similarité et la valeur de dissimilarité entre deux entités sont normalisées, notées S et DS , alors on a $S+DS=1$ [14].

3.1. Définition de la similarité

La similarité $S : O \times O \rightarrow \mathbb{R}$ [14] est une fonction à deux arguments vers un nombre réel satisfaisant les conditions suivantes :

La positivité : $\forall a, b \in O, S(a, b) \geq 0$

La maximalité: $\forall a, b \in O, S(a, a) \geq S(a, b)$.

La symétrie: $\forall a, b \in O, S(a, b) = S(b, a)$

La transitivité: $\forall a, b, c \in O, S(a, b) = S(b, c) \Rightarrow S(a, b) = S(a, c)$

La finitude: $\forall a, b \in O, S(a, b) < \infty$.

3.2. Méthodes de calcul de similarité

La Figure ci-dessous résume différentes mesures de similarité, catégorisées selon les techniques utilisées [15] [16].

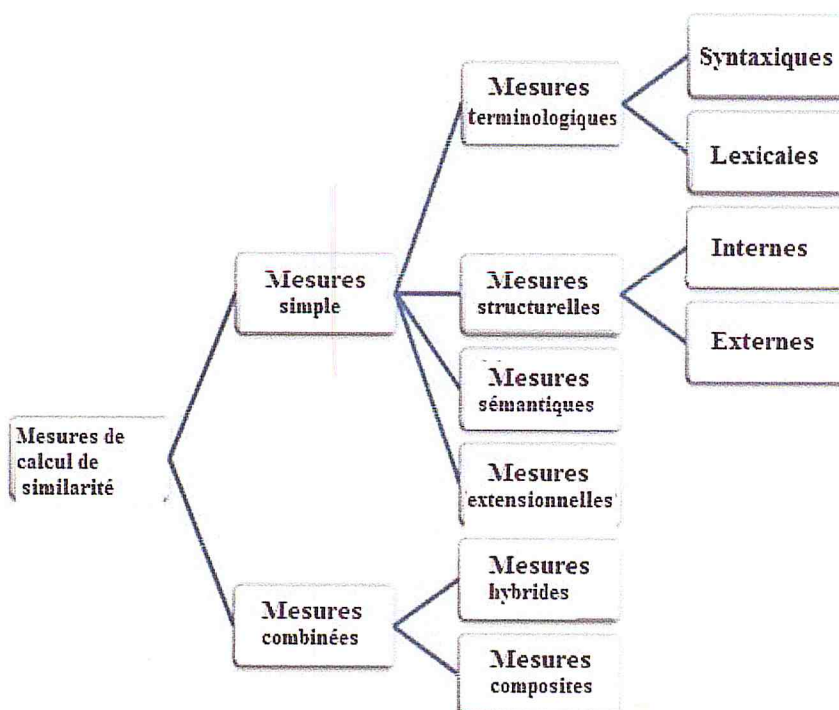


Figure 4: Mesures de calcul de similarités.

3.2.1. Les mesures simples

3.2.1.1. Les techniques terminologiques

Ces techniques se basent sur la comparaison des termes ou des chaînes de caractères. Elles sont employées pour calculer la valeur de la similarité des entités textuelles, des termes ou chaîne de caractères, telles que des noms, des étiquettes (label), des commentaires, des ou des propriétés des concepts descriptions. Le problème de ces comparaisons est l'existence des synonymes et des homonymes, il n'est pas possible d'assumer que deux entités sont similaires si elles portent le même nom, ou qu'elles sont différentes si elles possèdent des noms différents. Ces méthodes peuvent encore être divisées en deux sous-catégories : l'une contient des méthodes qui comparent des termes en se basant sur les caractères contenus dans ces termes et l'autre utilise certaines connaissances linguistiques [17].

3.2.1.1.1. Approche syntaxique (Les méthodes se basant sur des chaînes de caractères)

Ces méthodes analysent la structure des chaînes de caractères, l'ordre des caractères dans la chaîne, le nombre d'apparitions d'une lettre dans une chaîne pour concevoir des mesures de la similarité. Par contre, elles n'exploitent pas la signification des termes. Par exemple, les mesures dans cette catégorie retournent une grande valeur de similarité (jusqu'à 1) si elles comparent les termes « Voiture » et « voitures », mais une petite valeur, voir la valeur 0, si elles comparent les termes « voiture » et « bagnole ». Parmi ces méthodes on cite les plus utilisées :

La mesure de Cosine

La similarité de Cosine est très utilisée [18] en tant que mesure de ressemblance entre deux vecteurs à n dimensions en déterminant le cosinus de l'angle entre eux.

Soit deux vecteurs A et B, l'angle θ s'obtient par le produit scalaire et la norme des vecteurs :

$$\text{SIM}(A, B) = \frac{A \cdot B}{\|A\| \cdot \|B\|}$$

La similarité obtenue $\cos \theta \in [0; 1]$. Plus la mesure de l'angle est faible, plus son cosinus est élevé : avec cette formule, plus la similarité (A, B) s'approche de 1, plus la similarité entre les objets représentés par les vecteurs A et B est grande.

La mesure de Jaccard

C'est une mesure simple de similarité entre deux ensembles d'objets proposée par Paul Jaccard en 1901 [19] [20]. Il consiste à diviser le nombre d'objets en commun par le nombre d'objets distincts dans les deux ensembles, autrement dit le cardinal de l'intersection divisé par le cardinal de l'union

Le calcul de la similarité de Jaccard se fait comme suit :

$$\text{SIM}(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

La mesure de Jaro

La méthode de Jaro mesure la similarité entre deux chaînes de caractères. Plus la distance 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 [21].

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

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

Où:

- $|S_i|$ est la longueur de la chaîne de caractères s_i
- m est le nombre de caractères correspondants
- t est le nombre de transpositions.

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

$$\left(\frac{\max(|s1|, |s2|)}{2} \right) - 1$$

Le nombre de transpositions est obtenu en comparant le i ème caractère correspondant de $s1$ avec le i ème caractère correspondant de $s2$. Le nombre de fois où ces caractères sont différents, divisé par deux, donne le nombre de transpositions.

3.2.1.1.2. Approche lexicale (Les méthodes linguistiques)

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. Il existe plusieurs méthodes pour calculer la similarité en se basant sur l'approche lexicale, on cite les plus utilisées :

WordNet

WordNet est une ressource largement utilisée pour la découverte de relations entre des termes. Cette ressource est intégrée dans des dictionnaires en ligne comme un support linguistique (ex : The Free Dictionary). Les noms, les verbes, les adjectifs et les adverbes en anglais sont organisés en des ensembles des synonymes. Ces ensembles des synonymes, appelés synsets, sont reliés par différentes relations sémantiques. En utilisant WordNet, il est possible de trouver les synonymes d'un mot ou d'un terme [22].

Pour le calcul de la similarité linguistique la fonction $Syn(c)$ calcul l'ensemble des Synsets de WordNet du concept c ; soit $S = Syn(c1) \cap Syn(c2)$ l'ensemble des sens communs entre $c1$ et $c2$ à comparer, la cardinalité de S est :

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

Soit $\min(|Syn(c1)|, |Syn(c2)|)$ le minimum entre les cardinalité des deux ensembles $Syn(c1)$ et $Syn(c2)$ alors la similarité entre deux concepts $c1$ et $c2$ sera définie comme suit :

$$Sim_{ling} = \frac{\lambda(s)}{\min(|Syn(c1)|, |Syn(c2)|)}$$

3.2.1.2. Les techniques structurelles

Ce sont des techniques 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 [18].

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,...) [18]. 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 parmi ces méthodes on a :

La méthode de WU-Palmer

Cette méthode permet de quantifier la similarité en se basant sur le nombre d'arcs c'est-à-dire que c'est une mesure qui prend en compte la profondeur des concepts dans la hiérarchie. Étant donné une ontologie de racine R et soit deux concepts C1 et C2 de cette ontologie le principe de calcul est donné par la formule suivante [23]:

$$\text{SIM}(C1, C2) = \frac{2N}{N1 + N2}$$

N1 et N2 : sont respectivement les profondeurs du concept C1 et C2 en d'autre terme ce sont les distances séparant les nœuds c1 respectivement C2 du nœud racine R.

N : est la distance qui sépare le concept subsumant CS de c1 et de c2 du nœud R.

Sachant que N1 et N2 peuvent s'écrire respectivement comme : n1+N et n2 +N avec n1 et n2 les distances séparant les nœuds c1 respectivement C2 du nœud qui les subsume CS. Alors la formule de wu & palmer devient :

$$\text{SIM}(C1, C2) = \frac{2N}{n1 + n2 + 2 * N}$$

La figure suivante montre un exemple d'une structure de Wu-Palmer :

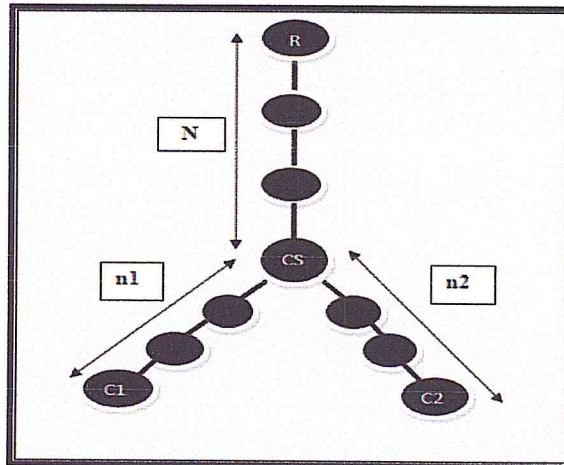


Figure 6: mesure Wu et Palmer appliqué aux concepts C1 et C2.

3.2.1.3. Les techniques extensionnelles

Elles déduisent la similarité entre deux entités qui sont notamment des concepts ou des classes en analysant leurs extensions (leurs ensembles d'instances). Elles considèrent que deux entités sont similaires si leur intersection se réduit à l'une des entités (en termes d'instances).

Le problème de ces méthodes est que si deux entités ne partagent pas les mêmes instances, ces méthodes vont donner un 0 comme similarité sans tenir compte à la similarité entre les instances.

Distance de Hamming

Soit S et T deux ensembles. La distance de Hamming entre S et T est une fonction de la similarité.

$$\text{SIM}(S, T) = \frac{|S \cup T - S \cap T|}{|S \cup T|}$$

Distance de Jaccard

Soit S et T deux ensembles. Soit P(x) la probabilité d'une instance aléatoire être dans l'ensemble X. La distance de Jaccard est une fonction de la similarité $S_{Jaccard}$ tel que :

$$\text{SIM}(S, T) = \frac{P(S \cap T)}{P(S \cup T)}$$

3.2.1.4. Les techniques sémantiques

Les mesures de similarité sémantiques permettent d'évaluer la proximité sémantique entre concepts c'est à dire d'estimer le degré par lequel les concepts sont proches dans leur sens [24], et cela à partir d'un calcul appelé métrique ou distance sémantique qui détermine si deux concepts sont similaires donc atteignent un certain niveau de ressemblance ou dissimilaire qui peuvent être également liés sémantiquement par des relations lexicales comme : l'antonymie, et spécialisation. Cette métrique peut être associée à des documents, des termes ou des entités.

Lin[24] définit la similarité sémantique entre deux concepts par rapport à leurs caractéristiques communes et à leurs différences cela veut dire plus ils ont de caractéristiques communes, plus les concepts sont similaires et par analogie plus ils ont de différences, moins ils sont similaire. La similarité maximale est obtenue lorsque deux concepts sont identiques.

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 come le lien « partie de » [24].

3.2.2. Les techniques combinées

La similarité entre deux entités peut donc être calculée en se basant sur plusieurs aspects. Il existe deux approches pour combiner les différentes techniques de mesure de similarités [25].

3.2.2.1. L'approche hybride

L'hybridation permettant de combiner, en un seul algorithme, plusieurs techniques basées sur différents critères et types d'information et les exécuter de manière séquentielle on choisissant un ordre d'exécution. Les méthodes hybrides ont l'avantage d'obtenir de meilleures performances que l'exécution de plusieurs algorithmes individuels séparément [25].

3.2.2.2. L'approche composite

Consiste tout d'abord à lancer parallèlement plusieurs mesures, puis par la suite à combiner leurs résultats [25]. Les méthodes composites ont l'avantage d'être adaptables plus facilement à différentes représentations de structures d'entrée.

Conclusion

Dans ce chapitre nous avons présenté un état de l'art sur les différentes techniques et méthodes utilisées qui peuvent résoudre le problème de recherche de la similarité et de la disimilarité ou de correspondance entre deux entités en général. On tient juste à rappeler que ces méthodes de similarités ne sont pas propres au domaine du matching des ontologies mais sont utilisés dans des applications qui ont besoin de calculer la similarité entre termes ou concepts.

Nous allons entamer donc dans le chapitre suivant la conception de notre système.

Chapitre III
Conception

1. Introduction

La conception est une étape très importante du processus de modélisation d'un système, ce chapitre porte sur la conception de notre système qui consiste à mettre en relation les projets de recherches avec les thèmes de recherches leur correspondant dans l'ontologie de la division DSISM du CERIST. Pour une première expérience l'affectation a été faite manuellement, cependant nous nous trouvons face à un problème tel que chaque projet de recherche est affecté à au moins un thème bien précis de la division et que le nombre gigantesque de thème de la nomenclature ACM rend la tâche difficile. D'où la nécessité d'automatiser l'affectation des projets de recherche aux thèmes de recherche correspondant dans la nomenclature ACM.

UML Unified Modeling Language est un langage de modélisation graphique conçu pour comprendre et décrire un problème, spécifier un système, concevoir et construire des solutions. Il est couramment utilisé en développement logiciel et en conception orientée objet.

Par le présent chapitre nous entamons la conception de notre système en donnant une brève description de la démarche utilisée qui sera suivi par notre approche proposée qui permet le calcul de la similarité.

2. Présentation de la démarche utilisée

Afin de développer notre système, Il nous a été indispensable de choisir une méthode de travail. Ainsi plusieurs méthodes ou langages ont vu le jour. En occurrence UML qui nous a permis de faire la conception de notre application.

Pour concevoir notre système nous avons suivi les phases suivantes: la Spécification des besoins, la Conception Préliminaire qui est l'analyse pour notre système, la Conception Détaillée et l'Implémentation.

2.1. Spécification des besoins

Cette phase consiste à comprendre le contexte du système. Il s'agit de déterminer les fonctionnalités et les acteurs les plus pertinents, d'identifier les cas d'utilisation. Tout cela est exprimé par l'ensemble des diagrammes de cas d'utilisation.

2.1.1. Les diagrammes de cas d'utilisation

Notre système comporte deux acteurs. Il s'agit d'un expert et un chercheur travaillant dans la division DSISM. La différence réside sur les droits d'accès et les limites de chacun.

Ø Le chercheur : Il s'agit d'un simple employeur, il utilise le système pour aligner les projets avec les thèmes.

Ø L'expert : Celui-ci peut faire à part les tâches du chercheur mais aussi gérer ces derniers. Ceci est intéressant car UML présente le principe d'héritage entre les acteurs. Donc on pourra faire de l'expert un héritier de chercheur et on va lui ajouter la particularité de pouvoir gérer des chercheurs et l'ontologie.

2.1.1.1. Diagramme de cas d'utilisation global

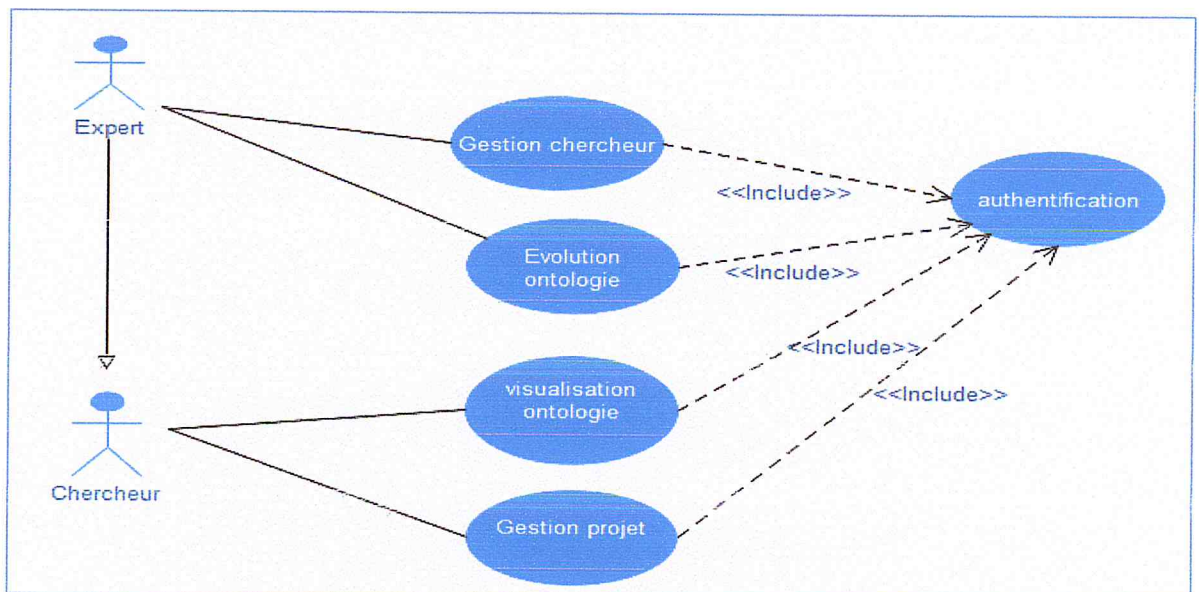


Figure 7: Diagramme de cas d'utilisation global.

| Cas d'utilisation | Description |
|---------------------------|--|
| Gestion chercheur | L'expert gère les chercheurs (ajout, modification, suppression, recherche). |
| Evolution d'ontologie | L'expert peut modifier l'ontologie (ajout et suppression de concept). |
| Visualisation d'ontologie | L'utilisateur peut voir la taxonomie de l'ontologie. |
| Gestion projet | L'utilisateur gère les projets de recherche (ajout, suppression, affectation). |

Tableau 1: description de cas d'utilisation global.

2.1.1.2. Diagramme de cas d'utilisation « Gestion chercheur »

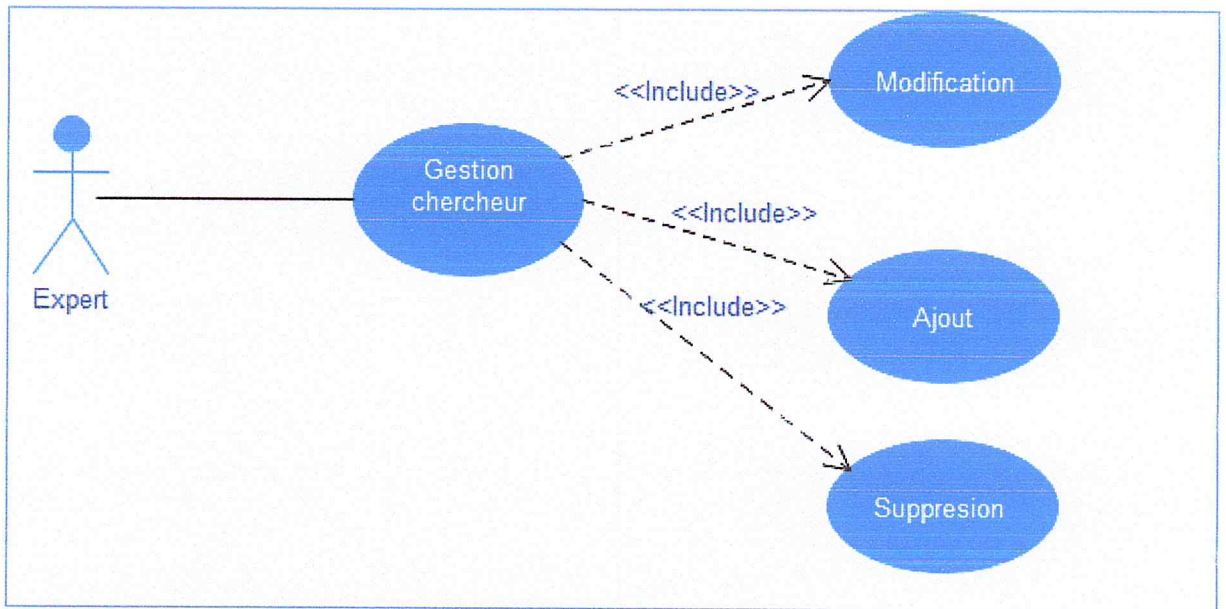


Figure 8: Diagramme de cas d'utilisation : « Gestion chercheur ».

| Cas d'utilisation | Description |
|-------------------|---|
| Modification | Permet de modifier les propriétés d'un chercheur. |
| Ajout | Permet d'ajouter un nouveau chercheur. |
| Suppression | Permet de supprimer un chercheur existant. |

Tableau 2 : Description de cas d'utilisation « Gestion chercheur ».

2.1.1.3. Diagramme de cas d'utilisation : « Gestion Projet »

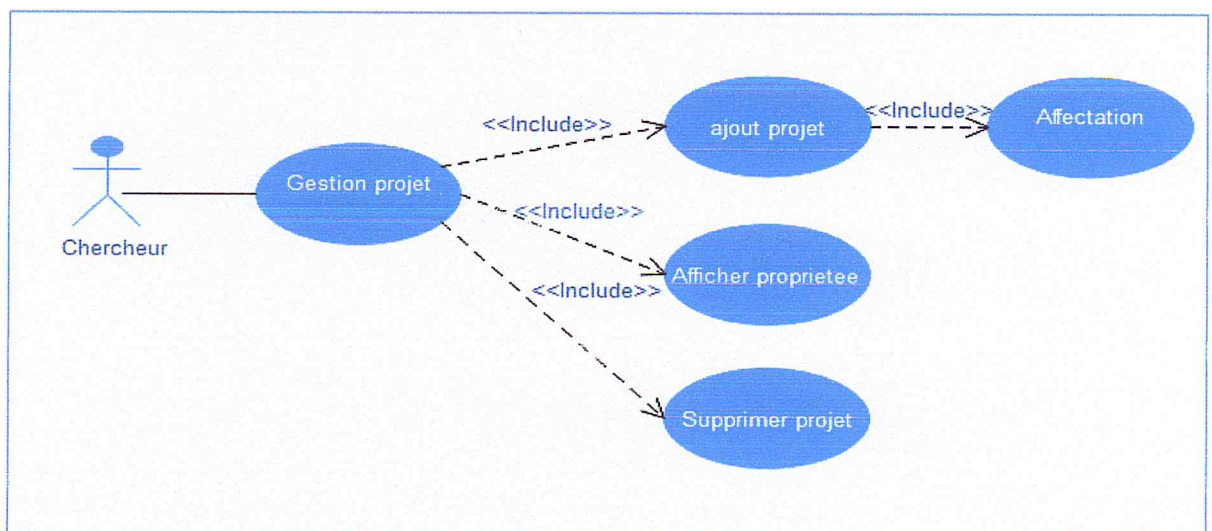


Figure 9: Diagramme de cas d'utilisation : « Gestion Projet ».

| Cas d'utilisation | Description |
|---------------------|---|
| Affectation | Permet d'affecter un projet à un thème. |
| Afficher propriétés | Faire la recherche sur les individus. |
| Ajout projet | Ajout de projet de recherche. |
| Suppression projet | Supprimer un projet. |

Tableau 3 : description de cas d'utilisation « Gestion Projet ».

2.1.1.4. Diagramme de cas d'utilisation : «Affectation »

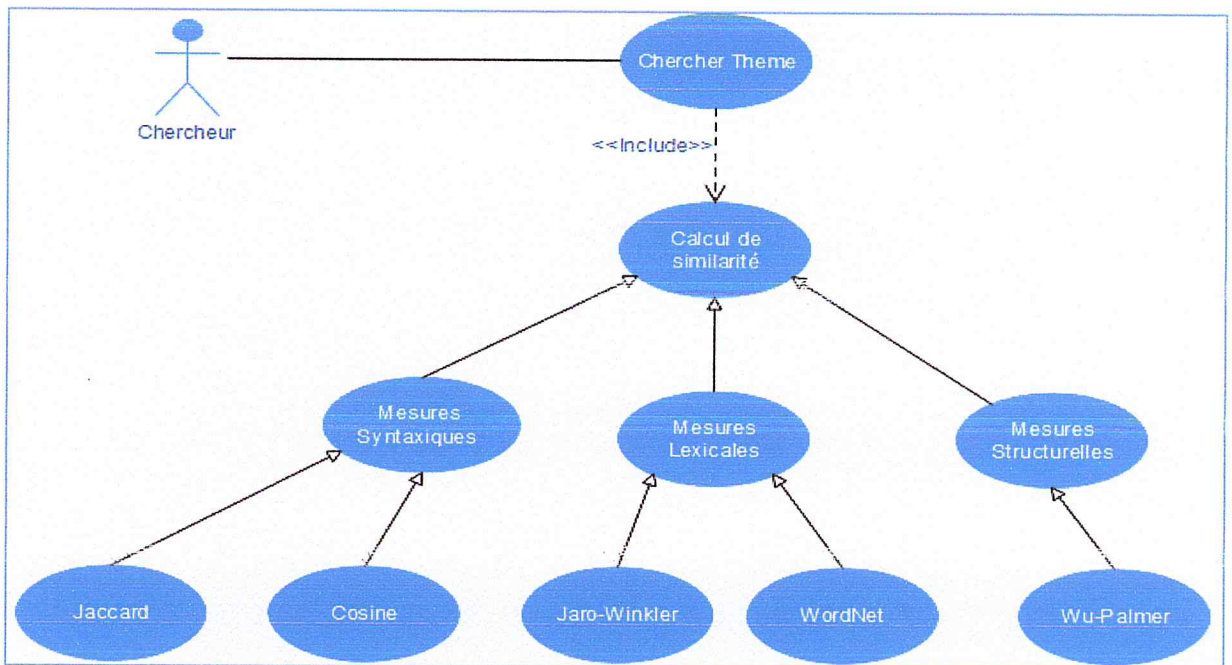


Figure 10: Diagramme de cas d'utilisation : «Affectation ».

| Cas d'utilisation | Description |
|--------------------|---|
| Calcul des mesures | Englobe les différentes méthodes de calcul des similarités (Syntaxiques, Lexicales et Structurelles). |
| Chercher thème | Calculer la similarité entre le projet et les thèmes et trouver les thèmes les plus appropriés. |

Tableau 4: description de cas d'utilisation « Affectation ».

2.2. Analyse des besoins

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. Pour ce faire, nous allons utiliser trois diagrammes de séquences.

2.2.1. Les diagrammes de séquences

Le diagramme de séquence représente la succession chronologique des opérations réalisées par un acteur : saisir une donnée, consulter une donnée, lancer un traitement, etc. Il indique les objets que l'acteur va manipuler, et les opérations qui font passer d'un objet à l'autre.

Voici quelques notions de base du diagramme : [21]

Scénario : une liste d'actions qui décrivent une interaction entre un acteur et le système.

Interaction : un comportement qui comprend un ensemble de messages échangés par un ensemble d'objets dans un certain contexte pour accomplir une certaine tâche.

Message : Un message représente une communication unidirectionnelle entre objets qui transporte de l'information avec l'intention de déclencher une réaction chez le récepteur.

2.2.1.1. Diagramme de séquence relatif au cas d'utilisation « authentification »

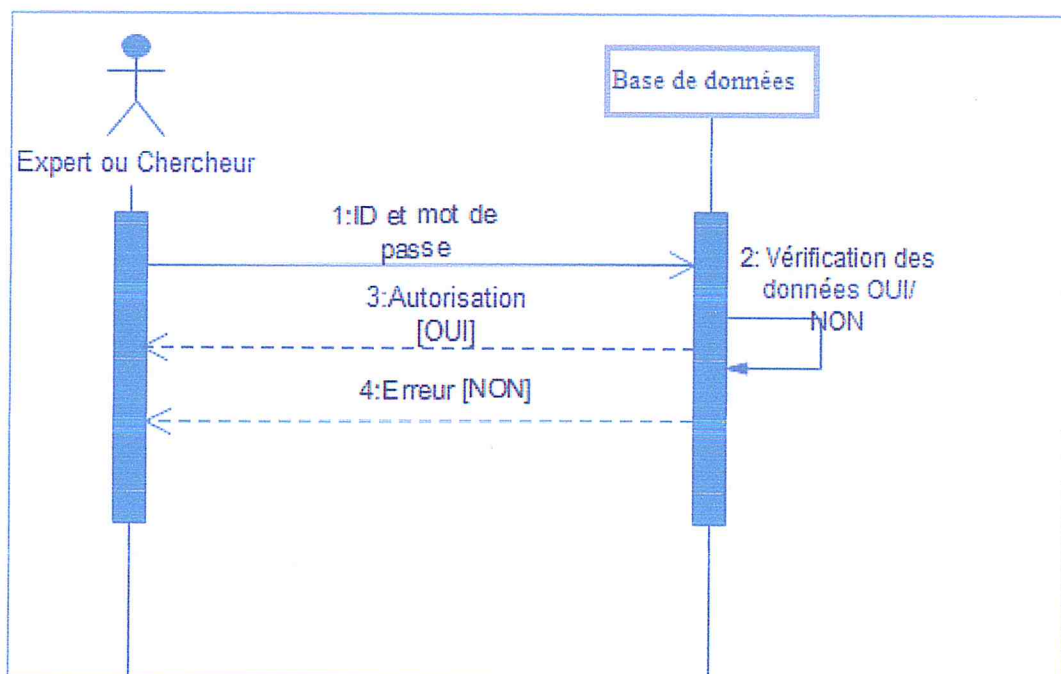


Figure 11: diagramme de séquence « authentification ».

Le diagramme de la figure décrit les scénarios possibles lors de l'identification de l'utilisateur.

| N° | Explication |
|----|---|
| 1 | L'utilisateur demande l'accès au system et donne l'ID et le mot de passe. |
| 2 | Un test doit être réalisé (existence et compatibilité du login/mot de passe). |
| 3 | Si les données sont correctes alors permette à l'internaute d'accéder à la totalité du system. |
| 4 | Si les données ne sont pas correctes et pas valide, l'utilisateur doit soit réessayer ou quitter le system. |

Tableau 5: description de cas d'utilisation « Authentification ».

2.2.1.2. Diagramme de séquence relatif au cas d'utilisation « Ajout Projet »

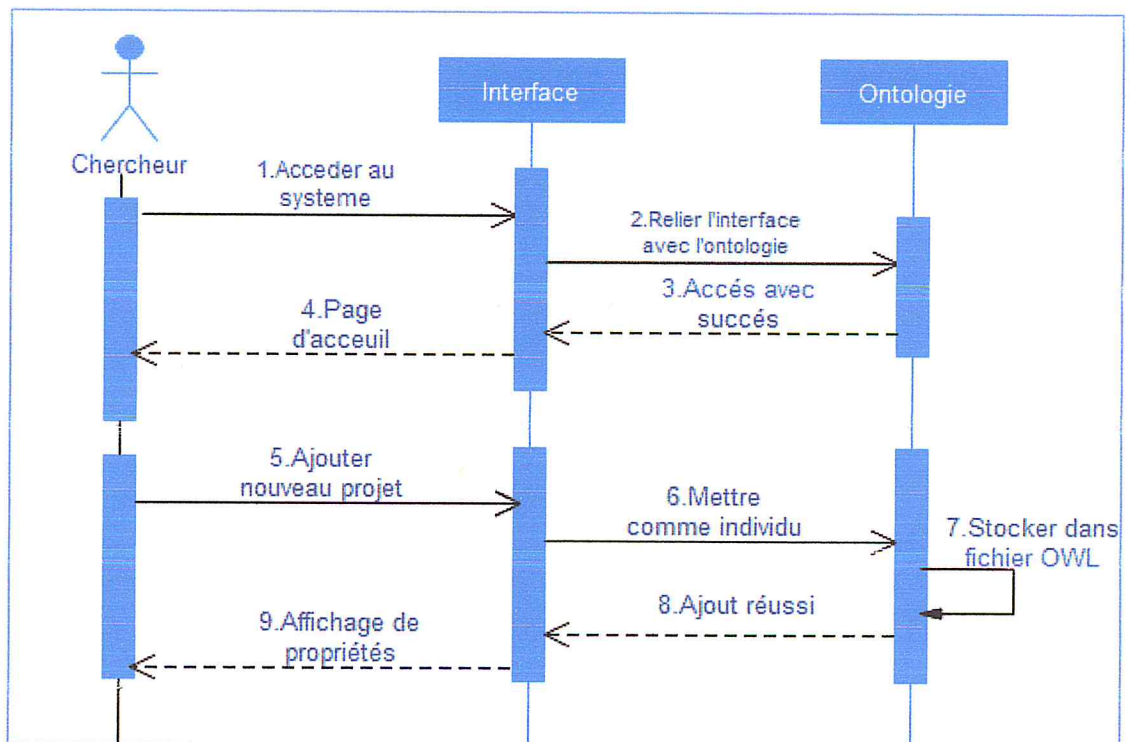


Figure 12: Diagramme de séquence de cas d'utilisation « Ajout Projet ».

| N° | Explication |
|----|---|
| 1 | Accéder au système en s'identifiant. |
| 2 | Mettre l'interface et l'ontologie en relation. |
| 3 | Accès à l'ontologie réussi. |
| 4 | Affichage de la page d'accueil à l'utilisateur |
| 5 | Instancier l'ontologie en ajoutant un nouveau projet. |

| | |
|---|---|
| 6 | Mettre le projet comme individu dans l'ontologie. |
| 7 | Une fois ajouté, le projet est stocké dans le fichier OWL de l'ontologie. |
| 8 | Ajout de projet avec succès. |
| 9 | Afficher le projet avec ces propriétés à l'utilisateur. |

Tableau 6: description de cas d'utilisation «Ajout Projet ».

2.2.1.3. Diagramme de séquence relatif au cas d'utilisation «Affectation »

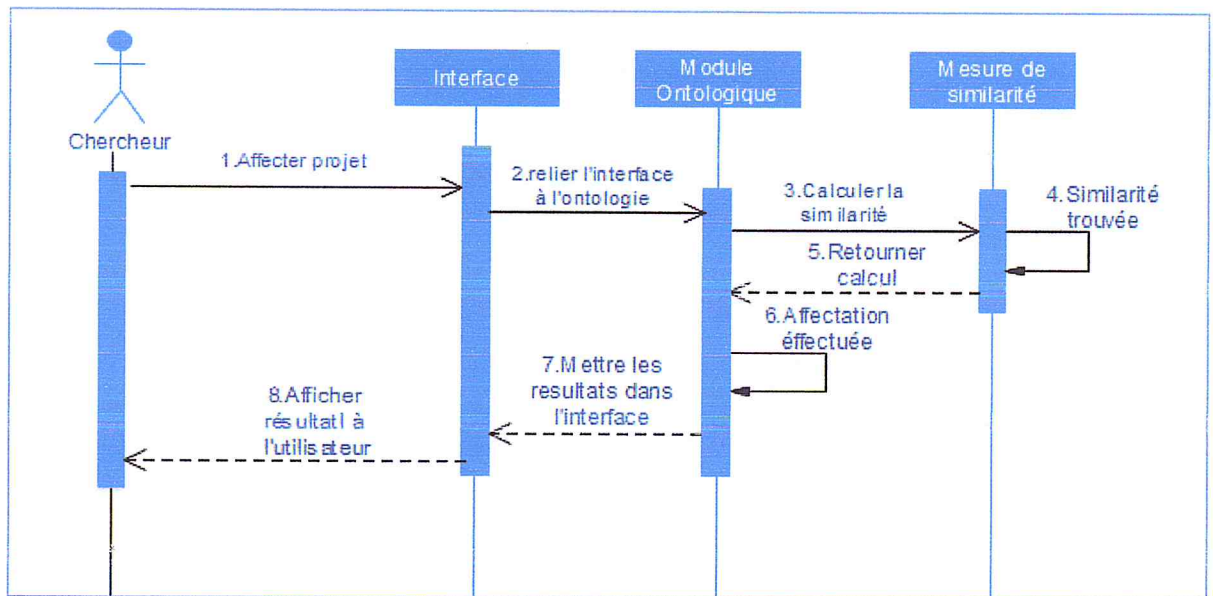


Figure 13 : Diagramme de séquence relatif au cas d'utilisation « Affectation ».

| N ° | Explication |
|-----|--|
| 1 | L'utilisateur appuie sur le bouton Affecté projet. |
| 2 | L'interface est relié à l'ontologie. |
| 3 | Le système utilise la classe de mesures de similarité pour calculer la similarité entre le projet et les thèmes. |
| 4 | Similarité trouvée. |
| 5 | Le module ontologique obtient le calcul. |
| 6 | Le module ontologique affecte le projet aux thèmes. |
| 7 | Le système affiche une interface de calcul |
| 8 | Affichage des résultats de l'affectation à l'utilisateur. |

Tableau 7: description de cas d'utilisation «Affectation ».

3. Conception du système

3.1. Conception générale

Il s'agit de concevoir un système pouvant satisfaire les exigences formulées à l'issue de l'étape de l'analyse et elle détermine les principaux modules de notre système.

3.1.1. Diagramme de paquetage

Permet de décomposer le système en catégories ou parties plus facilement observables, appelés « packages ». Cela permet également d'indiquer les acteurs qui interviennent dans chacun des packages.

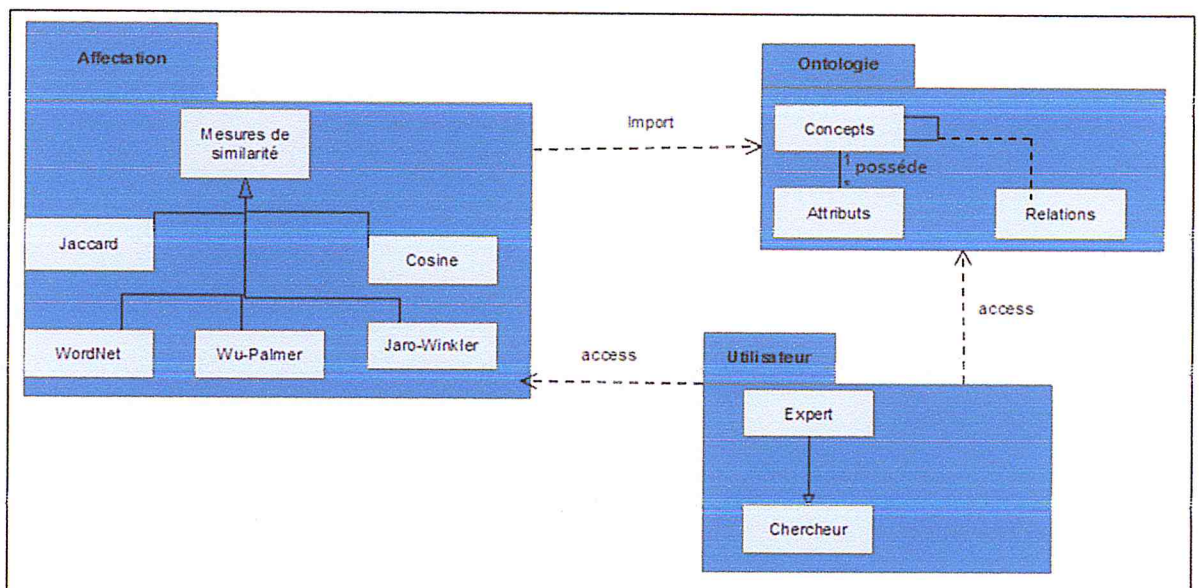


Figure 14: diagramme de paquetage.

Détails de paquetage de notre système

Nous voyons que le système que nous concevons peut être divisé en trois parties (ou packages) observables séparément.

Utilisateur : représente le paquetage de base et définit les méthodes d'authentification d'un expert et d'un chercheur et il accède aux 2 autres paquetages "espaces de noms" par la relation access et donc par les noms complets.

L'affectation: Ce package représente toutes les fonctions utilisées pour calculer les mesures de similarité, il importe donc des informations sur les projets de recherche et les thèmes de recherche) depuis les classes du paquetage de l'ontologie, il contient :

Mesure de similarité : comporte toutes les techniques de mesure de similarité entre les concepts pour calculer le degré de correspondance entre un projet et un thème en utilisant les mesures calculées par les autres classes filles (Jaro-Winkler, Jaccard, Cosine, Wu- Palmer, WordNet).

WordNet : c'est un dictionnaire lexical utilisé pour le calcul de la mesure de similarité lexicale.

Jaccard : Elle calcule le nombre des objets communs divisé par le nombre total des objets.

3.2. Conception détaillée

3.2.1. Architecture détaillé du système

L'architecture globale du système se présente comme suit :

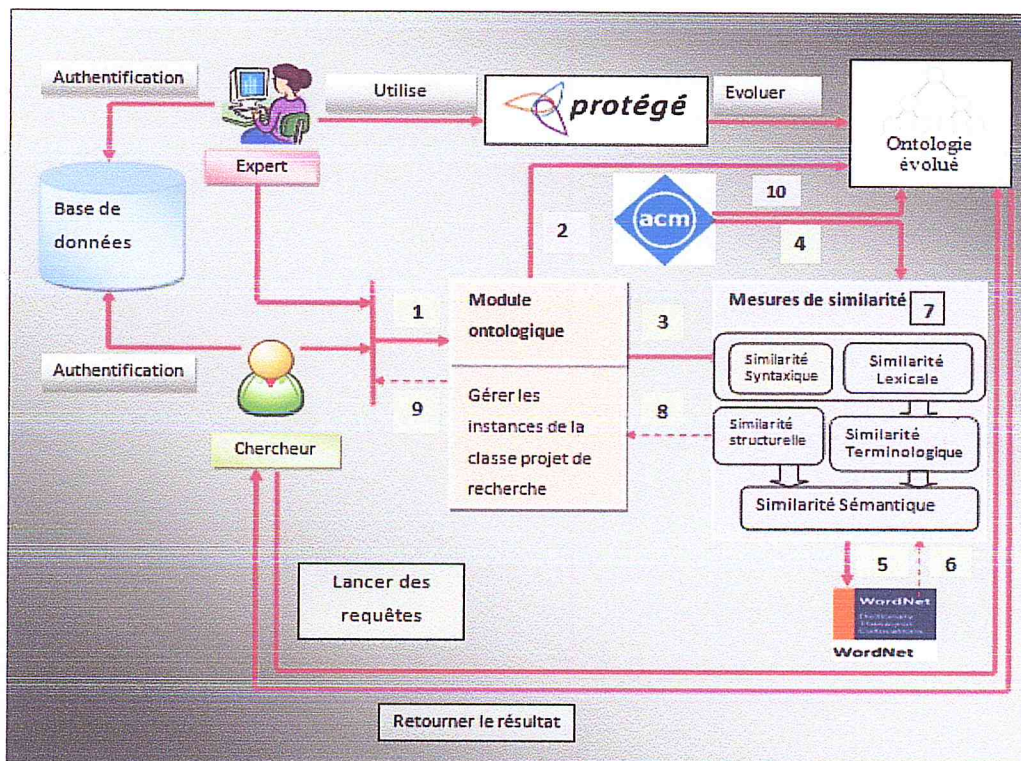


Figure 45:Schéma de fonctionnement du système.

Pour décrire l'architecture de notre système nous allons présenter les éléments importants et les relations les plus pertinentes existantes entre ces éléments.

| Numéro de la relation | Explication de la relation |
|-----------------------|---|
| 1 | L'expert et le chercheur lance la requête d'ajout d'un projet. |
| 2 | Le module ontologique exécute le processus d'ajout projet à l'ontologie. |
| 3 | Le module ontologique après avoir ajouter un projet à l'ontologie il effectue (lance) l'affectation automatique de ce projet de recherche à un thème de recherche dans l'ontologie. |
| 4 | Pour calculer la similarité le module de similarité récupère la liste des thèmes à partir du fichier XML de la nomenclature ACM. |
| 5 | Chercher les synonymes dans WordNet. |
| 6 | Réponse concernant la recherche des synonymes. |
| 7 | Le module mesure de similarité calcule les différentes similarités et la similarité globale. |
| 8 | La réponse est la liste des thèmes correspondant au projet de recherche. |
| 9 | Affichage des résultats aux utilisateurs. |
| 10 | L'évolution dans la nomenclature ACM permet également l'évolution dans l'ontologie |

Tableau 8:Explication des relations.

Dans cette section, nous allons nous intéresser à chaque module du système individuellement afin d'apporter plus d'explication.

Module(1) : « Authentification »

C'est la procédure qui consiste à autoriser l'accès aux utilisateurs à notre système

Utilisateurs: On distingue deux acteurs humains qui sont l'expert et le chercheur :

Un expert : il peut gérer les chercheurs comme il peut faire toutes les autres tâches.
C'est un administrateur qui a un rôle principal dans notre système

Un chercheur : C'est la personne qui utilise notre système pour effectuer l'affectation automatique.

La base de données : Contient les informations des utilisateurs donc elle permet à l'expert ainsi qu'au chercheur d'effectuer l'authentification.

Module(2) : « Evolution de l'ontologie »

L'expert accède à l'ontologie via l'éditeur Protégé qui permet d'appliquer des changements sur les concepts de l'ontologie, on trouve :

Protégé : Un éditeur qui a permis à l'expert d'évoluer l'ontologie en rajoutant et supprimant des concepts.

Ontologie évoluée: L'ontologie représentant la nomenclature ACM qui est stockée dans un répertoire sous forme d'objet ontologyserialisable (pour stockage ou échange),

La nomenclature ACM(CCS) est un système de classification de sujet pour l'informatique mis au point par l'association for ComputingMachinery (ACM) en 1972.

Elle est représentée notamment comme une ressource externe et correspond à une méta-description sous forme de fichier XML décrivant les thèmes de recherche. Les thématiques contenues dans l'ontologie seront alors chargées grâce au parcours du fichier XML qui est dans un environnement toujours en évolution en rajoutant de nouveaux éléments dans l'arborescence des thèmes et leurs sous thèmes

Voilà un schéma décrivant une partie de la structure des thèmes dans la nomenclature ACM.

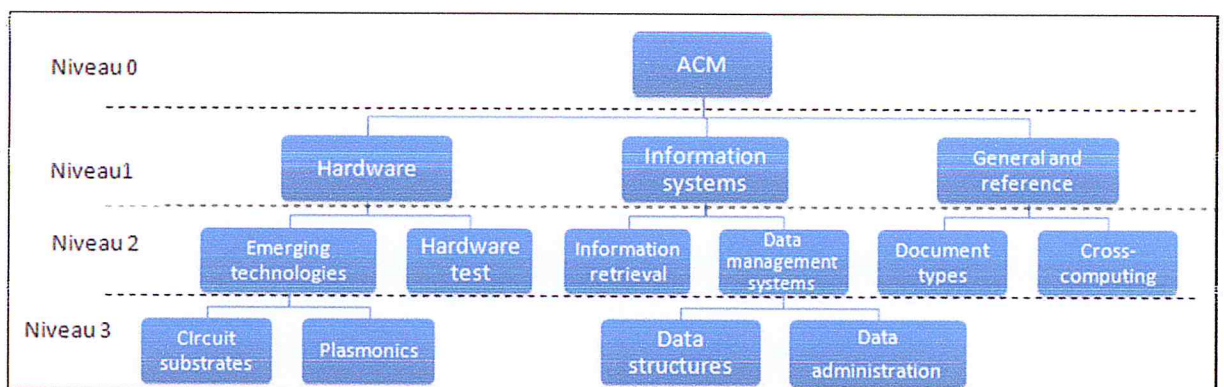


Figure 16: Arborescence des thèmes dans ACM.

Module(3) : « Module ontologique »

Un module qui consiste à exécuter des opérations sur l'ontologie (à mettre en œuvre toutes les requêtes de modification concernant notre ontologie) en incluant l'ajout de projet et sa suppression.

Le projet de recherche est représenté par son intitulé et son ensemble de mots clés qui est un mot ou un groupe de mots qui a une importance particulière permettant de caractériser le contenu du projet.

La détection automatique des relations se base sur une recherche et sélection de thèmes précéder d'un calcul qui est le calcul de la similarité globale entre projets et thèmes en incluant les mots clés des projets. Les résultats des calculs sont alors affichés dans une liste d'affichage qui permet non seulement de visualiser les résultats mais encore de les classer par ordre numérique décroissant.

Module(4) « Recherche d'information »

La recherche dans notre système permet d'interroger notre ontologie en utilisant le langage SPARQL afin de permettre aux chercheurs de retrouver les informations concernant les projets de recherche de la division DSISM.

Module(5) « Mesures de similarités »

Dans notre approche, nous avons utilisé la composition comme une méthode de combinaison des différentes stratégies de calcul de similarité. Elle permet de combiner les résultats produits par les différentes stratégies exécutées. Les résultats obtenus montrent que la méthode proposée donne de meilleurs résultats en comparaison avec ceux des méthodes considérés séparément. Elle est générale et applicable à tout type de méthodes.

Notre approche fonctionne en deux étapes successives. La première étape permet de calculer les similarités (terminologiques et structurelles) à savoir que les différentes stratégies sont exécutées indépendamment, chaque stratégie produit des valeurs de similarités partielles.

La deuxième étape permet de calculer la similarité globale, dite sémantique, comme dans ce qui suit:

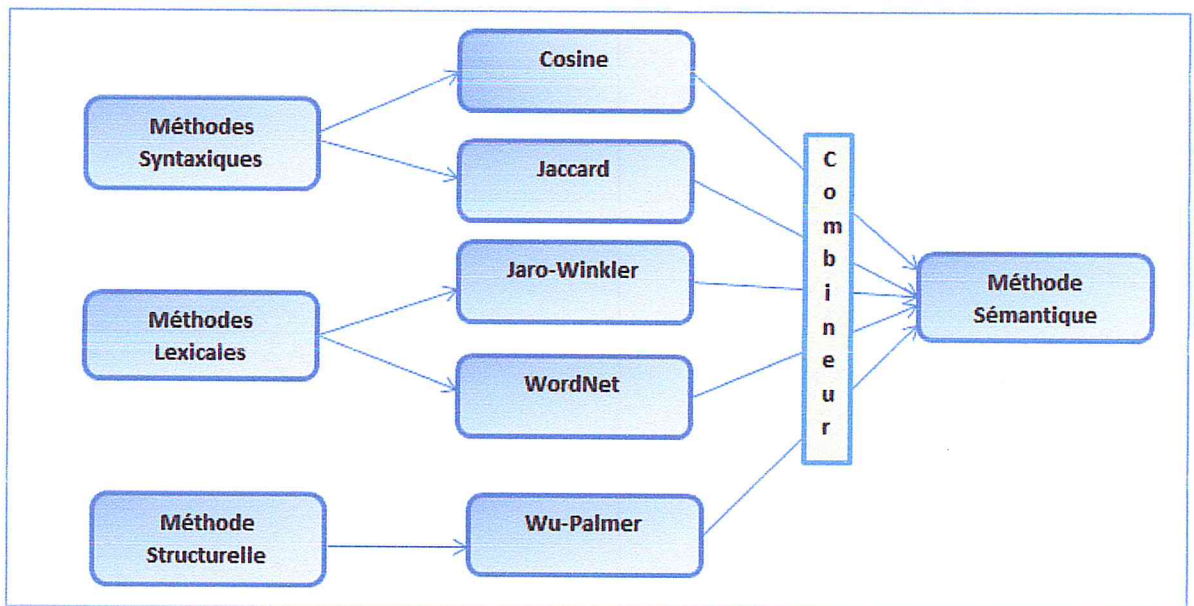


Figure 17: Schéma de combinaison des Méthodes.

1. La similarité terminologique

Il existe plusieurs méthodes de calcul de la mesure de similarité terminologique qui s'appuie sur l'approche syntaxique et lexicale.

1.1 L'approche syntaxique

Parmi ces méthodes on a choisi celle de « Cosine » et « Jaccard » car elles sont très utilisées dans le domaine de calcul de la similarité, faciles à implémenter et elles ont les meilleures performances au niveau des résultats dans plusieurs expériences

Cosine : la formule de calcul de la méthode « Cosine » se fait comme suit :

$$\text{SIM}(A, B) = \frac{A \cdot B}{\|A\| \cdot \|B\|}$$

Jaccard : la formule de calcul de la méthode « Jaccard » se fait comme suit :

$$\text{SIM}(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

1.2. Approche lexicale

Pour découvrir les relations sémantiques qui peuvent exister entre les différents termes on a fait recours à une ressource externe appelée WORDNET et aussi la méthode de Jaro-Winkler.

WordNet : le WordNet utilise la fonction $Syn(c)$ qui calcule l'ensemble des synonymes de WordNet du concept c .

$$Sim_{ling} = \frac{\lambda(s)}{\min(|Syn(c1)|, |Syn(c2)|)}$$

Tel que :

$$\lambda(S) = |Syn(c1) \cap Syn(c2)|$$

La mesure de Jaro-Winkler : la formule de calcul de la méthode « Jaro-Winkler » se fait comme suit :

$$SIM(A, B) = SIM_{jaro} + (l.p (1 - SIM_{jaro}))$$

2. La similarité structurelle

Pour les techniques structurelles on a fait recours à la mesure de Wu-Palmer car c'est la mesure la plus utilisée, cette mesure a l'avantage d'être facile à implémenter et peut donner une idée sur le lien sémantique entre les concepts. Elle est très performante parce qu'elle prend en compte toutes les relations entre les entités et qu'elle peut éliminer les correspondances incompatibles. Sa formule est la suivante :

$$SIM(C1, C2) = \frac{2N}{n1 + n2 + 2 * N}$$

3. La similarité sémantique (globale)

La similarité sémantique sera calculée par la combinaison des similarités terminologique et structurelle. Il faut donc un moyen pour combiner toutes les valeurs de

similarité calculées de chaque aspect pour produire une seule valeur de similarité représentative pour deux entités à comparer qui la moyenne pondérée [25].

3.2. Moyenne pondérée

Soit O l'ensemble d'objets qui peuvent être analysés dans n dimensions. Soit x et x' deux objets dans O . Soit w_i le poids de la dimension i . Et $\delta(x_i, x'_i)$ la similarité de la paire des objets à la dimension i .

$$\forall x, x' \in O, \delta(x, x') = \frac{\sum_{i=1}^n w_i \times \delta(x_i, x'_i)}{\sum_{i=1}^n w_i}$$

Nous avons proposé à titre de combinaison, la formule suivante :

$$\mathbf{Simsem(A, B) =}$$

$$\frac{(\mathbf{simjw(A, B)} * \omega . \mathbf{jw}) + (\mathbf{simwp(A, B)} * \omega . \mathbf{wp}) + (\mathbf{simjacc(A, B)} * \omega . \mathbf{jacc}) + (\mathbf{simcos(A, B)} * \omega . \mathbf{cos}) + (\mathbf{simWN(A, B)} * \omega . \mathbf{WN})}{\mathbf{coef. jw} + \mathbf{coef. wp} + \mathbf{coef. WN} + \mathbf{coef. jacc} + \mathbf{coef. cos} + \mathbf{coef. WN}}$$

$$\omega . \mathbf{jw} = \exp(\mathbf{simjw(A, B)})$$

Les autres coefficients se calculent de la même manière que le coefficient de la similarité de description.

Tels que :

Simsem(A, B) : est la similarité sémantique (globale) entre le projet et le thème.

Simjw(A, B) : est la similarité Jaro-Winkler entre le projet et le thème.

Simjacc(A, B) : est la similarité Jaccard entre le projet et le thème.

Simwp(A, B) : est la similarité Wu et Palmer entre le projet et le thème.

Simcos(A, B) : est la similarité Cosine entre le projet et le thème.

SimWN(A, B) : est la similarité WordNet entre le projet et le thème.

Si la mesure syntaxique entre le projet et le thème est < 0.3 (c'est-à-dire faible) alors on l'ignore car dans notre cas on compare deux phrases, donc si elles ne contiennent pas au

moins un terme semblable ca donnera un faible résultat sans prendre en considération la synonymie entre les termes alors la formule devient :

$$\text{Simsem}(A, B) = \frac{(\text{sim}_{jw}(A, B) * \omega . jw) + (\text{sim}_{wp}(A, B) * \omega . wp) + (\text{sim}_{WN}(A, B) * \omega . WN)}{\text{coef. } jw + \text{coef. } wp + \text{coef. } WN}$$

4. Le procédé de calcul de la similarité

La similarité entre un projet et un thème se fait comme suit

1. Calcul de la similarité entre l'intitulé du projet avec les thèmes de la nomenclature ACM.
2. Calcul de la similarité entre les mots clé du projet et les thèmes
3. La moyenne des résultats du calcul des mots clés avec les thèmes
4. Valoriser les résultats des mots clés en multipliant leur moyenne par 2 par rapport a l'intitulé du projet qui n'est parfois pas suffisamment spécifique au thème abordé
5. Rajouter le résultat à la valeur obtenue du calcul de l'intitulé du projet et la diviser par 3 (faire la moyenne).
6. Le résultat obtenu est le degré de similarité sémantique (globale) entre le projet de recherche et le thème de recherche.

5. Discussion des résultats obtenus

Au début nous avons calculé la similarité entre l'intitulé du projet et l'ensemble de thèmes de la nomenclature ACM en utilisant la formule proposé auparavant, mais les résultats étaient faibles à cause du manque de précision au niveau de intitulé qui est généralement insuffisant a décrire l'intention du projet lui-même donc il ne décrit pas assez la notion de ce dernier d'autant plus qu'il ne donne pas une spécification sur le thème ou le sujet abordée dans le projet de recherche.

De même que l'intitulé peut contenir des propositions comme « for, of, the, and... » Qui n'ont aucune influence sur le sens et donc peuvent diminuer le degré de correspondances.

Afin d'éviter leurs effets nous avons opté à les éliminer pour délimiter le calcul avec les termes les plus pertinents seulement.

Ce faible résultat nous a imposé un seuil limité égal à « 0.5 » pour déterminer les thèmes les plus appropriés au sujet.

Après plusieurs tests nous avons remarqué que les résultats obtenus n'étaient pas très justes, ainsi nous sommes arrivés aux thèmes recherchés prévus et donc les résultats étaient insuffisants.

Pour cela on a eu recours à l'utilisation des mots clés qui permettent de mieux décrire les projets de recherche pour pouvoir détecter d'éventuelles thèmes de recherche, cela a une influence positive donc sur la détection automatique des relations entre les thèmes et les projets de recherche.

Pour qu'un mot-clé soit pertinent et fonctionne, il faut qu'il soit reconnu par un outil de recherche dans un index déjà constitué qui est dans notre cas la nomenclature ACM.

Ces mots clés seront calculés avec tous les thèmes de l'ontologie en utilisant notre formule de combinaison de similarité. Nous avons privilégié la similarité entre les mots clés et les thèmes de recherche (en augmentant le coefficient) par rapport à la similarité entre l'intitulé du projet et les thèmes de recherche car les mots clés caractérisent mieux le projet que l'intitulé lui-même.

Après avoir effectué plusieurs tests avec la nouvelle méthode et faire une étude avec un expert de domaine sur les résultats obtenus par notre système d'affectation, nous avons réussi à avoir les résultats souhaités et augmenter leurs degrés de similarité.

Par conséquent, une augmentation du seuil doit être faite. Nous avons considéré les résultats du calcul qui ont une valeur de similarité globale supérieure ou égale à « 0.75 » sont des résultats valides et les résultats qui ne vérifient pas cette condition sont des résultats non valides.

Un seul projet de recherche peut être affecté à un ou plusieurs thèmes. Thème principal et d'autres secondaires, tel que le premier qui est classé dans la liste d'affichage des résultats de calcul de la similarité entre le projet et le thème est le thème principal car il a la plus grande similarité donc c'est le thème le plus approprié.

6. Conclusion

A l'issu de ce chapitre, nous avons présenté la spécification et l'analyse des besoins du système, ainsi que son architecture globale de fonctionnement où nous avons défini les principaux modules en détaillant la phase de calcul des mesures de similarité qui est le composant majeur de notre système. On a pu trouver la combinaison appropriée et la plus utile des différentes techniques pour atteindre notre objectif qui est l'affectation automatique des projet de recherches à leur thèmes adéquats .

Dans le chapitre suivant, nous allons implémenter et mettre en œuvre ce que nous avons proposé dans l'étude conceptuelle, en d'autres termes, l'implémentation de notre approche.

Chapitre IV
Implémentation

Introduction

Nous entamons par ce chapitre l'implémentation de notre système qui est une phase importante de la démarche de modélisation elle suit la conception et permet donc de transformer le modèle conceptuel établi précédemment en composants logiciels formant notre système.

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 pour présenter les différentes interfaces du système.

Partie I : Environnement de développement

1. Langages utilisé

1.1. JAVA

Java est un langage de programmation informatique orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems avec le soutien du cofondateur de Sun Microsystems Bill Joy.

« Sun Microsystems caractérise java par le fait qu'il est simple, orienté objet, distribué, interprété, robuste, sécurise, indépendant des architectures, portable, performant, multithread et dynamique. [25]

« In 1996, James Gosling, Bill Joy steele wrote for the first of the java Language Specification : "We believe that the java programming language is mature language ,ready for widespread use. Nevertheless ,we expect some evolution of the langage in the years to come .We intend to manage this evolution in a way that is completely compatible with existing applications." » [25]

1.2. SQL

Le SQL (Structured Query Language) est un langage de requête utilisé pour la manipulation des bases de données relationnelles. Il a été créé au milieu des années 70 par IBM et commercialisé par ORACLE en 1979. L'intérêt du SQL réside dans les caractéristiques suivantes : normalisation, Standard non procédurale, universel

1.3. XML

Recommandation du W3C [26] depuis le 10 février 1998, le langage XML eXtensible Markup Language a été défini dès son origine comme étant un méta-langage facilitant l'élaboration de langages à balises spécialisés.

UN document XML est une source de données dite bien formée c'est à dire qu'elle correspond parfaitement à la structure logique et à la structure physique, les deux structures imposées par la spécification de XML.

2. Outils

2.1. Netbeans

L'IDE NetBeans est un outil rapide et complet pour développer des logiciels Java où un Java Development Kit JDK est requis. Il respecte les standards et s'exécute sur n'importe quel système d'exploitation où une Machine Virtuelle Java est disponible, il est multilingue, et est doté d'une documentation, il peut supporter n'importe quel langage de programmation comme C, C++, JavaScript, XML, Groovy, PHP et HTML et cela de manière native ou par ajout de greffon comme pour Python ou Ruby et est multilingue. L'IDE NetBeans est un environnement de développement placé en open source par Sun en juin 2000 il peut être utilisé libre de coût pour n'importe quel type de développement logiciel. Bien que l'IDE soit modulaire (ses fonctionnalités peuvent être étendues par des plug-ins).

Le produit de base inclus le support pour le développement bureautique (AWT/Swing), le web (Servlets/JSP/JSF/Struts) et l'Édition Entreprise de Java (EJB et Services web), et inclus une base de données et un serveur d'application Java EE.

2.2. Protégé

Parmi les différents langages de représentation plus particulièrement dédiés à la définition d'ontologies, il faut citer OWL (Ontology Web Language). Ce langage issu du W3C (World Wide Web Consortium), basé sur la logique des descriptions et sur une syntaxe de type XML (Extensible Markup Language) est devenu une norme, principalement comme format d'échange. OWL est mis en œuvre dans l'environnement de développement d'ontologies Protégé est un logiciel libre de droits qui offre l'avantage de disposer d'outils de visualisation d'ontologies. Cependant Protégé est un outil de développement et non pas une méthode de construction d'ontologies [27].

2.3 Les APIs

JENA

Jena est une API Java fourni par le Framework Web sémantique et open source Apache Jena, elle possède des classes pour représenter des graphes, des ressources, des propriétés et des littéraux. Dans Jena, un graphe est nommé un modèle et est représenté par l'interface Model .L'API Jena offre la possibilité de créer et de manipuler des ontologies décrites en OWL ainsi que des graphes RDF.

WS4J

Ws4jWordNetSearching for java c'est un API qui permet aux applications java de récupérer des données à partir de la base de données WordNet.

3. Visualisation de l'ontologie dans protégé

L'onglet OWLViz de protégé permet de visualiser la hiérarchie des classes de l'ontologie.

La figure ci-dessous montre la visualisation de l'ontologie de la division DSISM du CERIST :



Figure 58: Visualisation de l'ontologie.

Cette visualisation n'est pas complète car on ne peut pas capturer toute la page dans protégé.

Les bulles en jaune désignent tous les concepts de l'ontologie.

Le concept « thing » est la classe mère.

La relation « is-a » désigne qu'un concept hérite d'un autre.

Partie II : Interface utilisateur

L'interface utilisateur que nous avons développée est schématisée comme suit :

1. Authentification :

Voici l'interface de l'authentification :



Figure 19: Interface d'authentification

L'interface d'authentification permet à un utilisateur d'accéder à l'interface principale et d'exécuter ces tâches, en entrant son identifiant et son mot de passe.

Une interface de menu principale s'affiche :

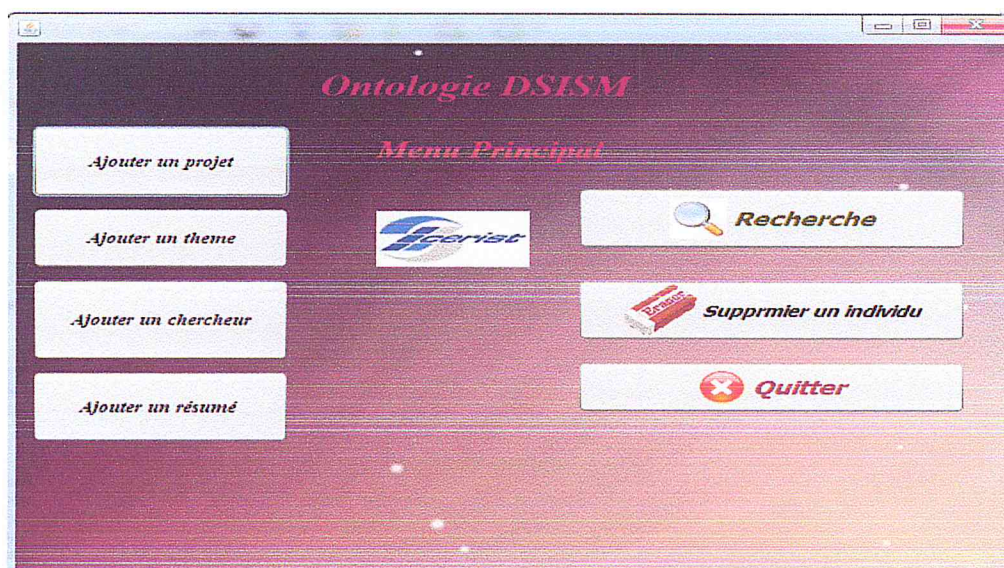


Figure 20: Menu de l'interface utilisateur.

Le menu comporte :

Le bouton Ajouter permet d'ajouter des individus dans l'ontologie pour chaque concept.

Le bouton Recherche permet d'effectuer des recherches d'individus et de relations.

Le bouton supprimer permet de supprimer un individu de l'ontologie.

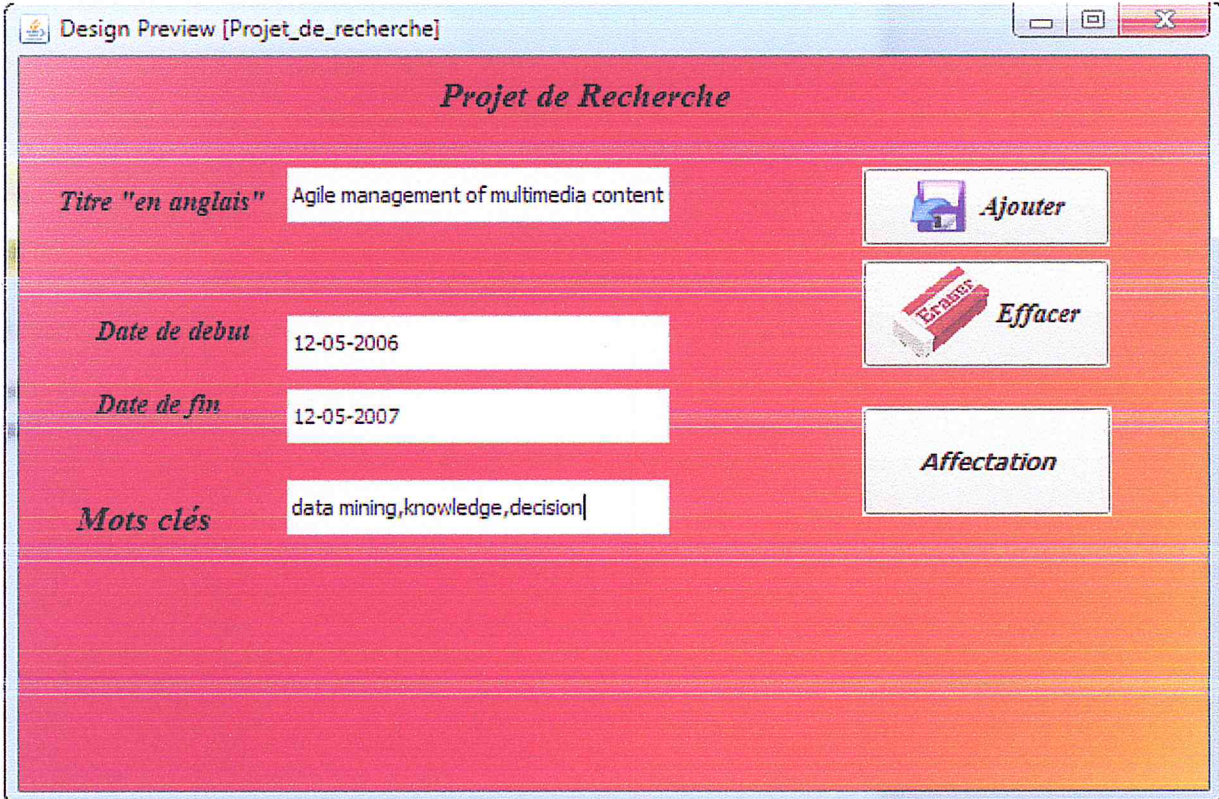
Le bouton quitter pour quitter l'interface.

Dans ce qui suit nous présentons les différentes fonctionnalités en détails.

2. Ajout des individus et relations

2.1. L'ajout d'un individu

La figure ci-dessous montre l'ajout de l'individu « Construction _ of _ platform _ for _ solutions _ Big_Data » de la classe « *Projet_de_Recherche* »



The screenshot shows a window titled "Design Preview [Projet_de_recherche]". The main content area has a red-to-orange gradient background and is titled "Projet de Recherche". It contains a form with the following fields and buttons:



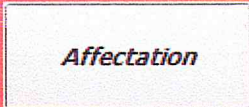
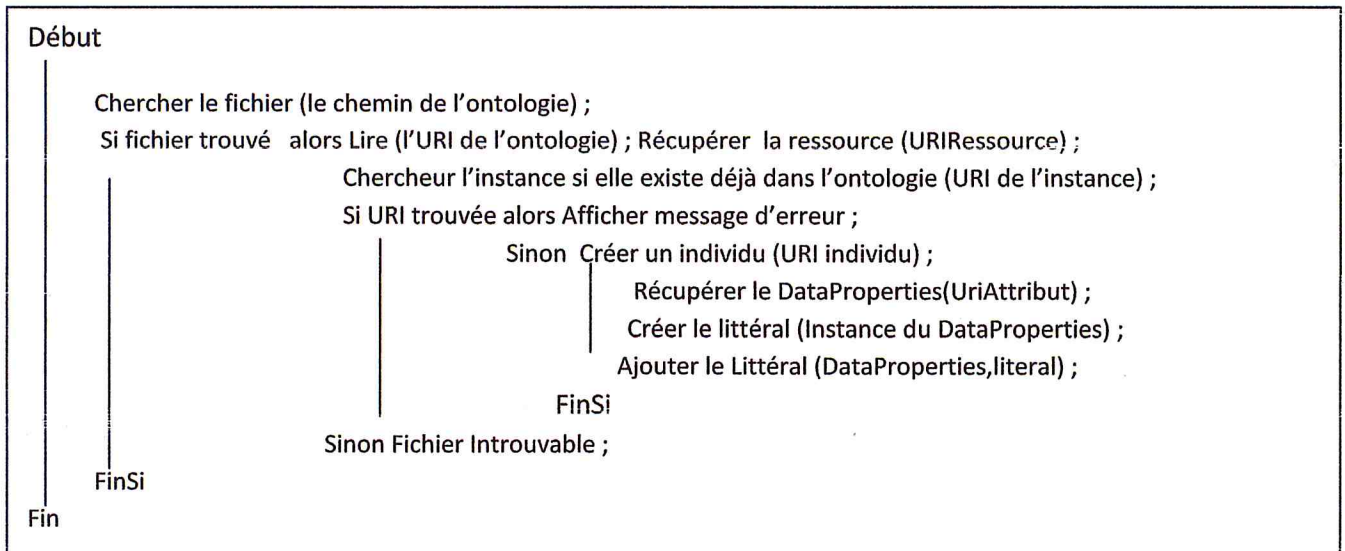
| | | |
|---------------------------|--|---|
| <i>Titre "en anglais"</i> | Agile management of multimedia content |  |
| <i>Date de debut</i> | 12-05-2006 |  |
| <i>Date de fin</i> | 12-05-2007 | |
| <i>Mots clés</i> | data mining,knowledge,decision |  |

Figure 21: Ajout d'un individu de « *Projet_de_Recherche* ».

Cet algorithme permet d'illustrer l'ajout d'un individu via l'Api JENA :



2.2. Ajout des relations

L'ajout des relations se fait en même temps que l'ajout des instances du « Domain » et « Range »

La figure suivante montre l'ajout de la relation « concerne » entre l'individu « Rapport_de_Projet » qui a déjà été ajoutée auparavant avec l'individu « Projet_de_Recherche » de la classe « Chercheur ce qui donne comme résultat : « Rapport_de_Projet » « concerne » « Projet_de_Recherche »,» illustré comme suit :

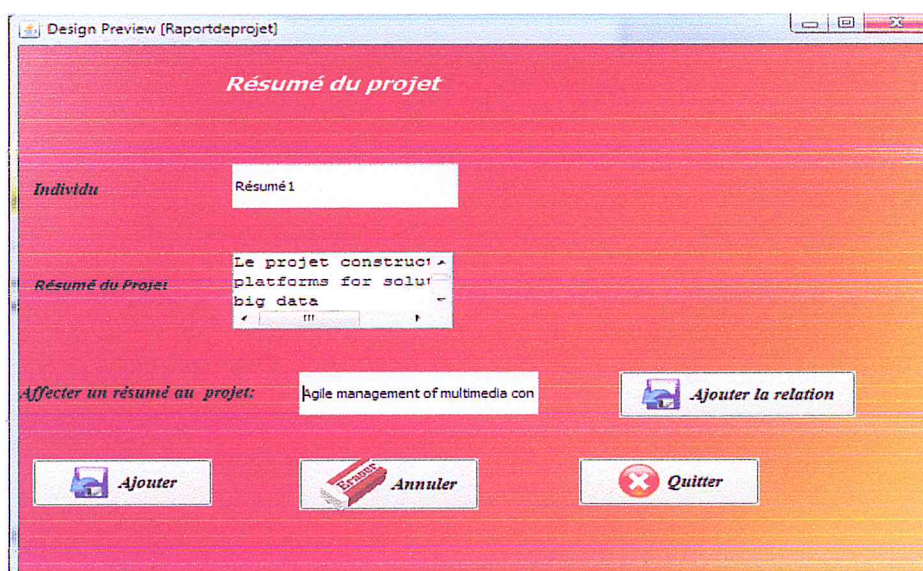


Figure 23: Ajout d'une relation.

3. Affectation

L'affectation du projet de recherche se fait juste après le calcul de la similarité de l'intitulé du projet et ses mots clés avec l'ensemble de thèmes.

La figure suivante montre l'interface qui permet d'effectuer cette affectation :

The screenshot shows a software window titled "Affectation Automatique". It contains three main sections:

- Top Section:** A text input field labeled "theme" with an "Ajoute" button to its right. Below this is a small horizontal slider and a button labeled "AFFIcher liste de theme".
- Middle Section:** A dropdown menu labeled "Theme:" with "Information systems" selected. Below it is a text input field labeled "Valeur:" containing "Construcion of platforms for solutions big data". Below that is a text input field labeled "Mots Clés" containing "data analytics, decision ,data mining, data wherhouse". A "Calcul" button is located at the bottom right of this section.
- Bottom Section:** A dropdown menu labeled "Resulta:" with "Information systems" selected. Below it is a large button labeled "affecter".

Figure24: Affectation automatique du projet de recherche.

4. La sérialisation

La sérialisation consiste à ouvrir le fichier en écriture et sauvegarder tous les changements, après l'ajout des ressources, l'utilisateur doit recharger l'ontologie modifiée par

l'interface comme la montre la figure ci-dessous :

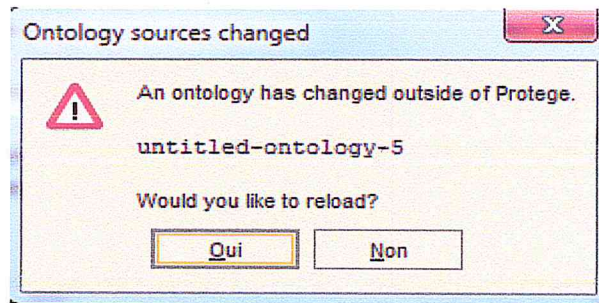


Figure 25: Sauvegarde des changements.

5. Recherche des informations

Pour effectuer des recherches, nous avons utilisé le langage SPARQL (SPARQL Protocol And RDF QueryLanguage).

5.1. Définition de SPARQL

SPARQL (SPARQL Protocol And RDF QueryLanguage) recommandation du W3C depuis janvier 2008, est un langage qui permet l'interrogation de descriptions RDF. SPARQL exploite, en premier lieu RDF à travers la notion de triplets ou d'ensembles de triplets. La réponse à la requête posée va ainsi correspondre à la restitution du sous-graphe RDF satisfaisant la requête.

5.2. Caractéristique et construction de requête SPARQL

SPARQL permet d'exprimer des requêtes interrogatives ou constructives :

Le PREFIX indique l'adresse (espace de noms) d'un schéma pouvant être exploité, voilà les caractéristiques d'une requête SPARQL :

-Une requête SELECT de type interrogative, permet d'extraire du graphe RDF un sous graphe correspondant à un ensemble de ressources vérifiant les conditions définies dans la clause WHERE.

-Une requête CONSTRUCT de type constructive, engendre un nouveau graphe qui complète le graphe interrogé.

- UNION : graphes alternatifs (correspond à au moins un des graphes précisés)

- DESCRIBE : retourne une description des ressources satisfaisant la requête.

- OPTIONAL : pattern(s) de graphe optionnel(s).
- DISTINCT : permet de retourner le résultat sans redondance.
- FILTER : restreindre les valeurs
- ASK : évalue si la requête va retourner un ensemble de ressources ou bien l'ensemble vide.

Exemple de requête SPARQL :

```
PREFIX onto: <http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?y ?Nom
WHERE
```

Figure 26: exemple de requête SPARQL.

5.3. Fonctionnalité recherche de l'interface utilisateur

La fenêtre recherche permet d'interroger notre ontologie en utilisant le langage SPARQL afin de permettre aux chercheurs de retrouver les informations concernant les projets de recherche de la division DSISM.

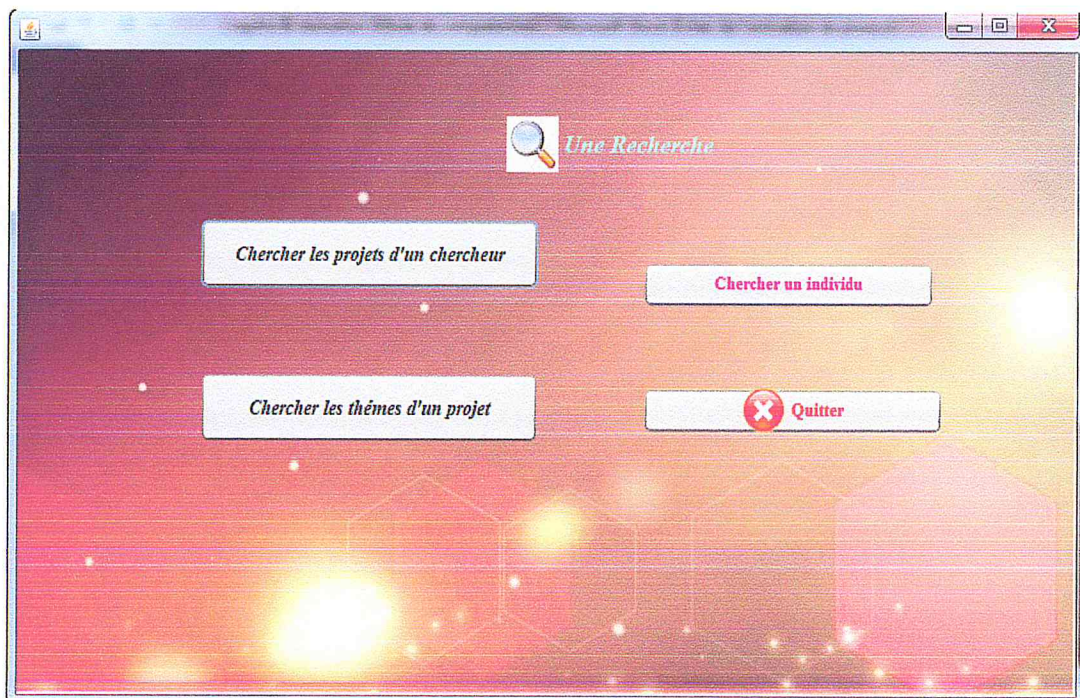


Figure 27: Fenêtre recherche.

La fonctionnalité rechercher permet de faire la recherche par individu, par nom de chercheur ou bien par relation.

La figure ci-dessous permet de chercher les thèmes d'un projet ajouté auparavant :

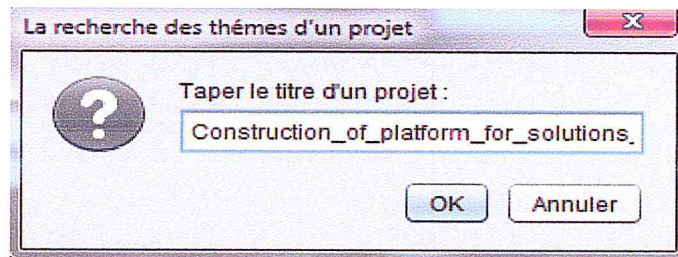


Figure 68: Chercher les thèmes d'un projet.

Une fois le titre du projet est introduit, la requête SPARQL renvoi les thèmes relatifs à l'individu :

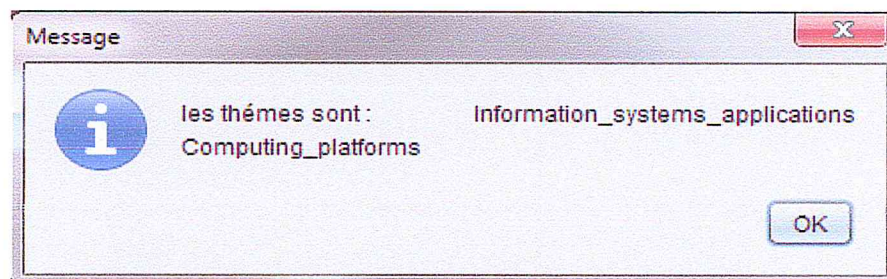


Figure 29: Résultat de la recherche.

Cette fenêtre montre l'affectation automatique d'un projet de recherche à des thèmes de recherche en utilisant la formule combinée de mesure de similarité.

6. Suppression de l'individu

Introduire l'individu comme suit :

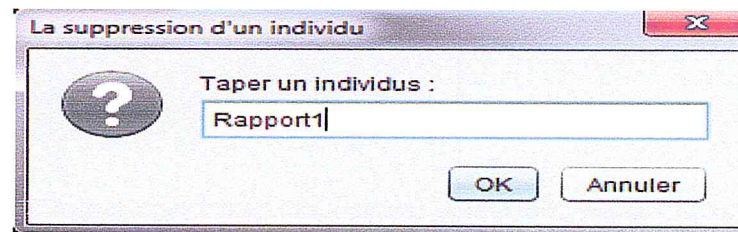


Figure 30: Supprimer un individu.

Une demande de confirmation de suppression se rapportera sur l'éditeur Protégé comme suit :

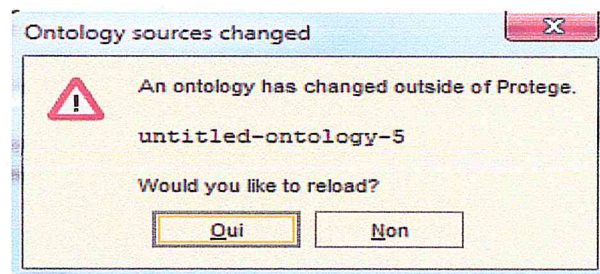


Figure 31: Confirmation de Suppression sur l'ontologie.

7. Tests de validation

Les tests sont appliqués sur des exemples bien étudiés par un expert de domaine :

7.1. Test 1:

Titre projet: Construction of platforms for solutions big data

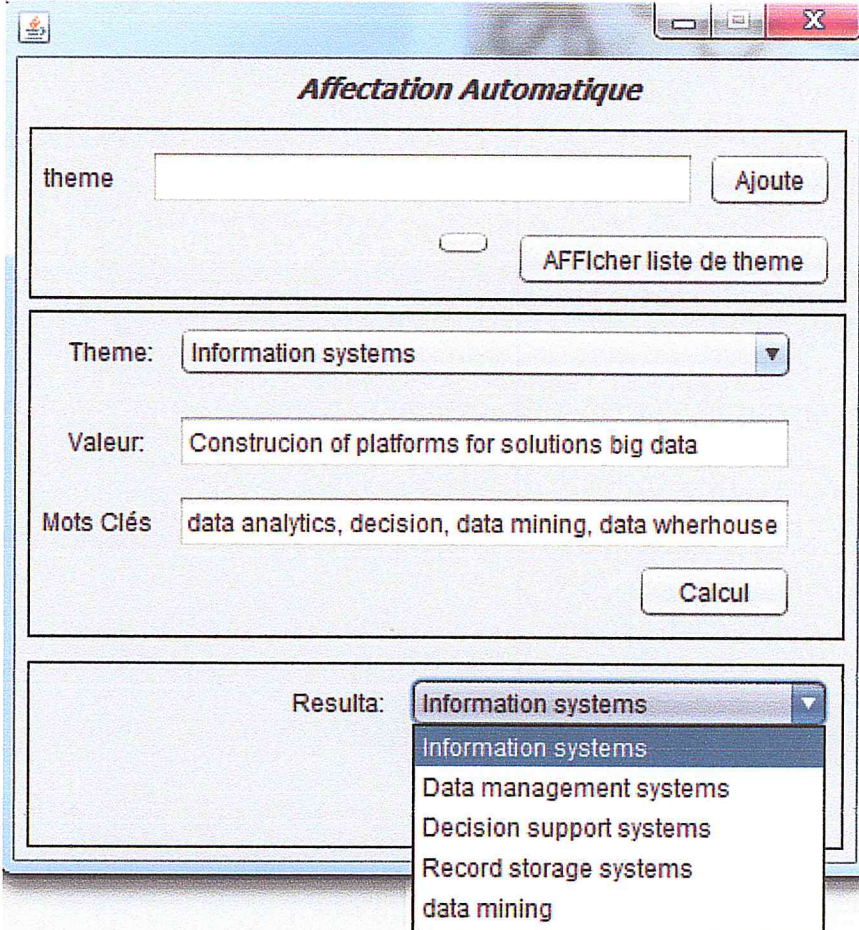
Mots clés: data analytics, decision, datamining, data warehouse

Après avoir effectué l'affectation automatique des projets, mots clés aux thèmes de recherche nous avons obtenu les résultats suivants :

| thema | themeb | resultcalc |
|--|--|--------------------|
| Construcion of platforms for solution... | Information systems | 0.878428898835718 |
| Construcion of platforms for solution... | Decision support systems | 0.8717181985411802 |
| Construcion of platforms for solution... | Record storage systems | 0.8661830054401662 |
| Construcion of platforms for solution... | data mining | 0.8515265652814845 |
| Construcion of platforms for solution... | Data management systems | 0.8497121487384026 |
| Construcion of platforms for solution... | Computer and information systems t... | 0.694320764534241 |
| Construcion of platforms for solution... | History of computing | 0.6922225062152272 |
| Construcion of platforms for solution... | Information systems education | 0.6911774326486136 |
| Construcion of platforms for solution... | Combinatorics on words | 0.6908791318582658 |
| Construcion of platforms for solution... | History of computing theory | 0.6906036001531763 |
| Construcion of platforms for solution... | Information systems applications | 0.6898504188055193 |
| Construcion of platforms for solution... | Mobile information processing syste... | 0.6898368895472519 |
| Construcion of platforms for solution... | Extraction, transformation and loading | 0.6893002862850478 |
| Construcion of platforms for solution... | Computing most probable explanation | 0.6888262690513862 |
| Construcion of platforms for solution... | Computing education programs | 0.6884580805876352 |
| Construcion of platforms for solution... | Information technology education | 0.6868917926938345 |
| Construcion of platforms for solution... | Permutations and combinations | 0.6866779239170075 |
| Construcion of platforms for solution... | History of programming languages | 0.6858395085193536 |
| Construcion of platforms for solution... | History of software | 0.6856666419501705 |
| Construcion of platforms for solution... | Combination and fusion | 0.6855093501800087 |
| Construcion of platforms for solution... | Computing profession | 0.6849372757317613 |
| Construcion of platforms for solution... | Computing organizations | 0.6845625708140212 |
| Construcion of platforms for solution... | Approximation algorithms analysis | 0.6829845634685483 |
| Construcion of platforms for solution... | information system economics | 0.6823816147916184 |
| Construcion of platforms for solution... | Software selection and adaptation | 0.6822557667098218 |
| Construcion of platforms for solution... | Information retrieval query processing | 0.6821071943587976 |
| Construcion of platforms for solution... | Probabilistic reasoning algorithms | 0.6821014495777368 |
| Construcion of platforms for solution... | Classification and regression trees | 0.681938727256723 |
| Construcion of platforms for solution... | Computing occupations | 0.6814274395048773 |
| Construcion of platforms for solution... | Antitrust and competition | 0.6813367757570534 |
| Construcion of platforms for solution... | Enterprise information systems | 0.6812587210461185 |

Figure 32: Résultats du calcul de "Test 1".

Les résultats de l'affectation sont affichés dans la figure suivante :



The screenshot shows a software window titled "Affectation Automatique". It contains several input fields and buttons. At the top, there is a "theme" input field with an "Ajoute" button to its right. Below this is a radio button and a button labeled "AFFicher liste de theme". The main section has a "Theme:" dropdown menu currently showing "Information systems", a "Valeur:" input field with the text "Construcion of platforms for solutions big data", and a "Mots Clés" input field with the text "data analytics, decision, data mining, data wherhouse". A "Calcul" button is located at the bottom right of this section. The "Resulta:" section at the bottom features a dropdown menu that is open, displaying a list of results: "Information systems", "Information systems", "Data management systems", "Decision support systems", "Record storage systems", and "data mining".

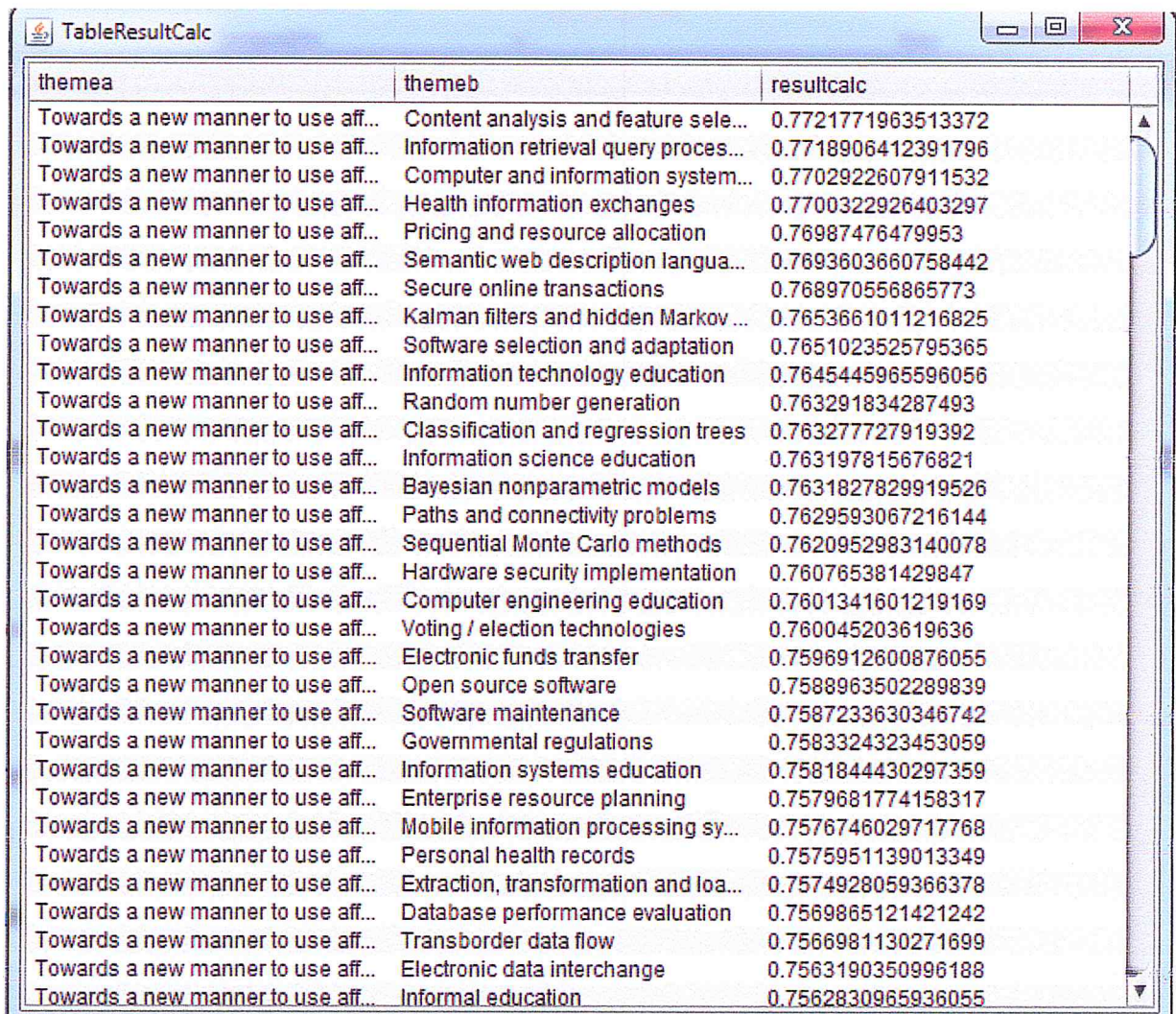
Figure 33 : Résultat de l'affectation de "Test 1".

7.2. Test 2

Titre du project: Towards new manner to use affordable technologies for handicraftwomen in emerging countries

Mots clés : information retrievial, management,

Après avoir effectué l'affectation automatique des projets, mots clés aux thèmes de recherche nous avons obtenu les résultats suivants :



| themea | themeb | resultcalc |
|------------------------------------|---------------------------------------|--------------------|
| Towards a new manner to use aff... | Content analysis and feature sele... | 0.7721771963513372 |
| Towards a new manner to use aff... | Information retrieval query proces... | 0.7718906412391796 |
| Towards a new manner to use aff... | Computer and information system... | 0.7702922607911532 |
| Towards a new manner to use aff... | Health information exchanges | 0.7700322926403297 |
| Towards a new manner to use aff... | Pricing and resource allocation | 0.76987476479953 |
| Towards a new manner to use aff... | Semantic web description langua... | 0.7693603660758442 |
| Towards a new manner to use aff... | Secure online transactions | 0.768970556865773 |
| Towards a new manner to use aff... | Kalman filters and hidden Markov ... | 0.7683661011216825 |
| Towards a new manner to use aff... | Software selection and adaptation | 0.7651023525795365 |
| Towards a new manner to use aff... | Information technology education | 0.7645445965596056 |
| Towards a new manner to use aff... | Random number generation | 0.763291834287493 |
| Towards a new manner to use aff... | Classification and regression trees | 0.763277727919392 |
| Towards a new manner to use aff... | Information science education | 0.763197815676821 |
| Towards a new manner to use aff... | Bayesian nonparametric models | 0.7631827829919526 |
| Towards a new manner to use aff... | Paths and connectivity problems | 0.7629593067216144 |
| Towards a new manner to use aff... | Sequential Monte Carlo methods | 0.7620952983140078 |
| Towards a new manner to use aff... | Hardware security implementation | 0.760765381429847 |
| Towards a new manner to use aff... | Computer engineering education | 0.7601341601219169 |
| Towards a new manner to use aff... | Voting / election technologies | 0.760045203619636 |
| Towards a new manner to use aff... | Electronic funds transfer | 0.7596912600876055 |
| Towards a new manner to use aff... | Open source software | 0.7588963502289839 |
| Towards a new manner to use aff... | Software maintenance | 0.7587233630346742 |
| Towards a new manner to use aff... | Governmental regulations | 0.7583324323453059 |
| Towards a new manner to use aff... | Information systems education | 0.7581844430297359 |
| Towards a new manner to use aff... | Enterprise resource planning | 0.7579681774158317 |
| Towards a new manner to use aff... | Mobile information processing sy... | 0.7576746029717768 |
| Towards a new manner to use aff... | Personal health records | 0.7575951139013349 |
| Towards a new manner to use aff... | Extraction, transformation and loa... | 0.7574928059366378 |
| Towards a new manner to use aff... | Database performance evaluation | 0.7569865121421242 |
| Towards a new manner to use aff... | Transborder data flow | 0.7566981130271699 |
| Towards a new manner to use aff... | Electronic data interchange | 0.7563190350996188 |
| Towards a new manner to use aff... | Informal education | 0.7562830965936055 |

Figure 34: Résultats du calcul de "Test 2".

Les résultats de l'affectation sont affichés dans la figure suivante :

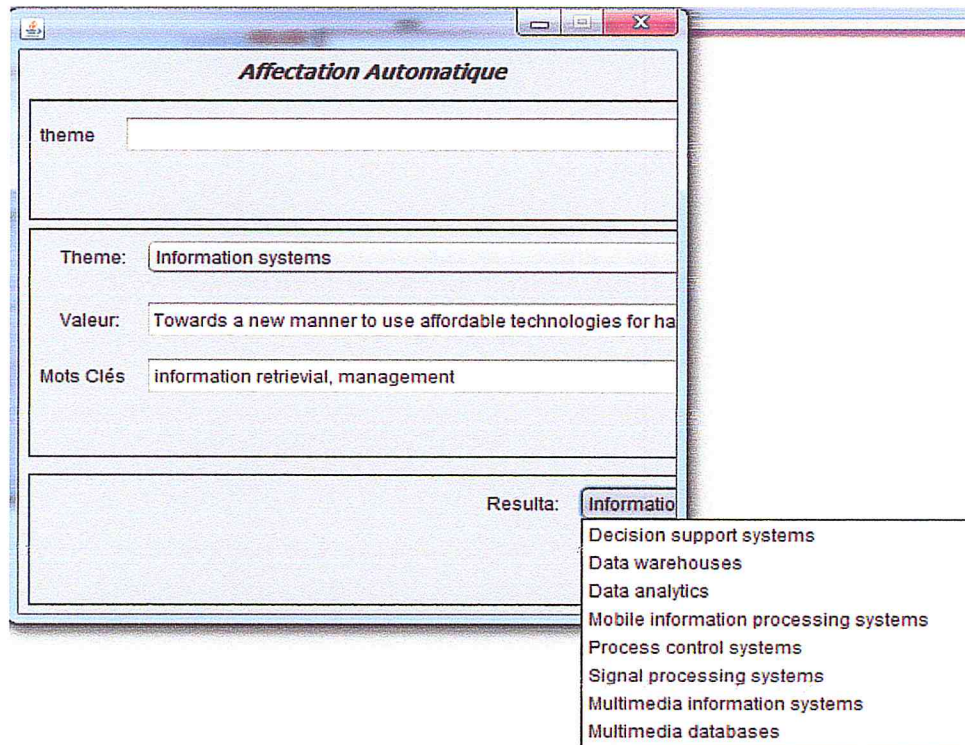


Figure 75: Résultat e l'affectation de "Test 2".

Les résultats montrent que le fait de rajouter les mots clés pour le calcul de la similarité en plus de l'intitulé du projet a contribué à l'augmentation de la mesure de similarité et donc à faire une affectation automatique adéquate et qui correspond aux résultats de l'affectation manuelle des projets de recherche aux thèmes correspondants.

8. Conclusion

Dans ce chapitre, nous avons présenté l'interface utilisateur avec une description générale des principales fonctionnalités du système, en utilisant l'api JENA pour l'ajout des instances, la formule combinée de mesure de similarité pour l'affectation automatique des projets de recherche aux thèmes de recherche et le langage SPARQL pour la recherche des informations.

Chapitre IV

Conclusion Générale & Perspective

Conclusion Générale

Actuellement, la recherche d'informations est basée sur une indexation par mots clés. Le processus de recherche de demain est plus "intelligent" car il est capable de prendre en compte le sens des mots clés, plutôt que de les considérer comme de simple chaîne de caractères dépourvus de toute signification.

Notre objectif est de proposer une architecture d'affectation automatique des projets de recherches à des thèmes de recherches pris de la nomenclature ACM dans l'ontologie DSISM du CERIST, en utilisant différentes techniques reposantes sur la définition de mesures de similarité afin de trouver les relations entre ces deux concepts de l'ontologie.

Nous avons combiné des techniques de base de la similarité tel que les techniques syntaxiques (mesure de Jaccard et Cosine), structurelles (mesure de Wu & Palmer) et lexicales (mesure de WordNet) afin de rendre l'affectation plus pertinente.

Nos recherches et tests observés, nous ont mené à optimiser une méthode efficace pour l'affectation automatique. L'efficacité de notre méthode réside dans la combinaison des caractéristiques suivantes :

- Automatisation totale du processus d'affectation : l'utilisateur n'aura qu'à saisir l'intitulé des projets de recherche qu'il veut affecter ainsi que son ensemble de mot clé et le système prend la main et le décharge entièrement des autres tâches comme le calcul de la similarité et l'ajout à l'ontologie.

- Conformité des résultats : La mesure de similarité qu'adopte notre système étant composite (implique à la fois : similarités terminologiques aussi bien lexicales que syntaxiques, similarités structurelles et bien sûr sémantiques), elle lui confère une conformité dans le calcul des correspondances entre les projets de recherche et les thèmes de recherche. En plus, l'utilisation de la ressource externe « WordNet » à jouer le plus grand rôle pour répondre nos besoins sémantiques, ce qui a permis de produire une application riche et fidèle (sémantiquement) au domaine auquel elle doit se rapporter.

Nous avons atteint l'objectif du travail demandé, ce dernier peut être élargi en rajoutant ces quelques perspectives :

- l'utilisation des mots clé pour les thèmes de recherche pour donner plus de détails et d'information et donc avoir des résultats plus exacts dans le calcul de la similarité.
- Comme on a pu calculer la similarité entre les projets de recherche et les thèmes de recherche on vise à calculer la similarité du nouveau projet à introduire dans l'ontologie non seulement avec les thèmes de recherche mais aussi on le compare avec les résultats des projets déjà affectés.
- Paramétrer le niveau d'approfondissement de la recherche ACM.
- Elargir le travail avec les différentes thématiques pour les différentes divisions du CERIST.
- Utiliser éventuellement d'autres nomenclatures autres qu'ACM

Annexes

1. Présentation

Jena est un ensemble d'outils dédiés à la construction d'applications orientées Web sémantique. Parmi ces outils, on trouve notamment une API Java open-source permettant de manipuler de nombreux langages tels que OWL, RDF/RDFS, SPARQL ou encore N3 et de raisonner sur des modèles ontologiques à l'aide de moteurs d'inférences inclus dans Jena ou externes.

2. Création d'un fichier OWL

Pour stocker en mémoire des ontologies OWL, l'API Jena propose aux utilisateurs la classe `OntModel`. Cette dernière hérite de la classe `Model` permettant de représenter des modèles RDF.

Pour permettre la construction de modèles d'ontologies, Jena met à disposition la classe `ModelFactory`. Le code ci-dessous présente la création d'un modèle d'ontologie sous Jena :

```
import com.hp.hpl.jena.ontology.* ;
import com.hp.hpl.jena.ontology.impl.* ;
import com.hp.hpl.jena.rdf.model.* ;

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

Quelques opérations de manipulation de l'ontologie avec l'API JENA :

| <i>Méthodes des classes <code>OntClass</code> et <code>OntModel</code></i> | |
|--|--|
| <i>Méthode</i> | <i>Description</i> |
| <code>listClasses()</code> | Renvoi dans l' objet <i>Iterator</i> toutes les classes Contenu dans le modèle. |
| <code>listHierarchyRootClasses()</code> | Renvoi 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' union d' autres classes. |
| <code>listSubClasses()</code> | Renvoi dans un <i>Iterator</i> les objets <code>OntClass</code> contenant les sous classes d' une classe |
| <code>listSuperClasses()</code> | Renvoi dans un <i>Iterator</i> les objets <code>OntClass</code> contenant les supers 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. |

Tableau : Opérations de manipulation avec l'API JENA

3. Le fichier OWL de l'Ontologie DSISM

```

<rdf:RDF
  xmlns="http://www.semanticweb.org/ontologies/2012/3/untitled-ontology-59#"
  xmlns:j.1="http://www.semanticweb.org/ontologies/2012/3/untitled-ontology-59#Chef_d'apos:"
  xmlns:j.0="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#est_résponsible_d'apos:"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:untitled-ontology-5="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:j.2="http://www.semanticweb.org/ontologies/2012/3/untitled-ontology-59#est_résponsible_d'apos:"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:j.3="http://www.semanticweb.org/ontologies/2012/3/untitled-ontology-59#Membre_d'apos:"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:j.5="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#Membre_d'apos:"
  xmlns:j.4="http://www.semanticweb.org/ontologies/2012/3/" ⌘
]
<rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#Information"
  <rdf:type rdf:resource="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#theme"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#NamedIndividual"/>
</rdf:Description>
]
<rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#Personnel_S"
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#Membre"
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
]
<rdf:Description rdf:nodeID="A0">
  <rdfs:comment rdf:nodeID="A1"/>
  <owl:annotatedProperty rdf:resource="http://www.w3.org/2000/01/rdf-schema#range"/>
  <owl:annotatedSource rdf:resource="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#es"
  <owl:annotatedTarget rdf:resource="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#Eq"
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Axiom"/>
</rdf:Description>
]
<rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#Manifestati"
  <owl:equivalentClass rdf:nodeID="A2"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#RésuméSuj">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#sujet"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#Conseil_De_c"
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#fait">
  <owl:inverseOf rdf:resource="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#est_fait"
  <rdfs:range rdf:resource="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#Stage"/>
  <rdfs:domain rdf:resource="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#Membre_d&ar"
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#InverseFunctionalProperty"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#numero_peric"
  <rdfs:range rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#PlainLiteral"/>
  <rdfs:domain rdf:resource="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#Périodique"
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#Information"
  <rdf:type rdf:resource="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#theme"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#NamedIndividual"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#TypeManif">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#Manifestati"
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>

```



```

</rdf:Description>
<rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#Cours">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#donne">
  <owl:inverseOf rdf:resource="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#est_donne">
  <rdfs:range rdf:resource="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#Cours"/>
  <rdfs:domain rdf:resource="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#Chercheur"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#InverseFunctionalProperty"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#Date_de_nais">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#Chercheur"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#Construction">
  <untitled-ontology-5:est_affecte_à rdf:resource="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5:est_affecte_à rdf:resource="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5:Date_finPro>02/2016</untitled-ontology-5:Date_finPro>
  <untitled-ontology-5:Date_debutProj>09/2015</untitled-ontology-5:Date_debutProj>
  <untitled-ontology-5:TitrePro>Construction of platform for solutions Big Data</untitled-ontology-5:TitrePro>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#NamedIndividual"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#est_attrait">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#Data_manager">
  <rdf:type rdf:resource="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#theme"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#NamedIndividual"/>
</rdf:Description>
<rdf:Description rdf:nodeID="A29">
  <owl:members rdf:nodeID="A27"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#AllDisjointClasses"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#apparait">
  <owl:inverseOf rdf:resource="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#est_apparait">
  <rdfs:range rdf:resource="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#Périodique"/>
  <rdfs:domain rdf:resource="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#Article_SCI"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#NomDiv">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#Division"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#Type">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#sujet"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#DateCom">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#est_dans">
  <owl:inverseOf rdf:resource="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#se_compose">
  <rdfs:domain rdf:resource="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#Membre_d'une">
  <rdfs:range rdf:resource="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#Equipe"/>

```

```
<owl:annotatedTarget rdf:resource="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#Co
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Axiom"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#TitreMnif">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#Manifestati
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</rdf:Description>
<rdf:Description rdf:nodeID="A25">
  <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
  <rdf:first rdf:resource="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#sujet"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#projet4">
  <untitled-ontology-5:Date_finPro>02/02/2014</untitled-ontology-5:Date_finPro>
  <untitled-ontology-5:TitrePro>systeme</untitled-ontology-5:TitrePro>
  <untitled-ontology-5:Date_debutProj>01/01/2013</untitled-ontology-5:Date_debutProj>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#NamedIndividual"/>
</rdf:Description>
<rdf:Description rdf:nodeID="A30">
  <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
  <rdf:first rdf:resource="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#Personnel_Sor
</rdf:Description>
<rdf:Description rdf:about="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#NomCon">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-5#Conseil_De
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</rdf:Description>
</rdf:RDF>
```


4. Aperçu sur le fichier XML de la nomenclature ACM

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE RDF>
<rdf:RDF xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:skos="http://www.w3.org/2004/02/skos/core#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
- <skos:ConceptScheme rdf:about="http://totem.semedica.com/taxonomy/The ACM
Computing Classification System (CCS)">
  <dc:title>The ACM Computing Classification System (CCS)</dc:title>
  <dc:date>2012</dc:date>
  <skos:hasTopConcept rdf:resource="#10002944"/>
  <skos:hasTopConcept rdf:resource="#10002950"/>
  <skos:hasTopConcept rdf:resource="#10002951"/>
  <skos:hasTopConcept rdf:resource="#10002978"/>
  <skos:hasTopConcept rdf:resource="#10003033"/>
  <skos:hasTopConcept rdf:resource="#10003120"/>
  <skos:hasTopConcept rdf:resource="#10003456"/>
  <skos:hasTopConcept rdf:resource="#10003752"/>
  <skos:hasTopConcept rdf:resource="#10010147"/>
  <skos:hasTopConcept rdf:resource="#10010405"/>
  <skos:hasTopConcept rdf:resource="#10010520"/>
  <skos:hasTopConcept rdf:resource="#10010583"/>
  <skos:hasTopConcept rdf:resource="#10011007"/>
  <skos:hasTopConcept rdf:resource="#10011641"/>
</skos:ConceptScheme>
- <skos:Concept xml:lang="en" rdf:about="#10002944">
  <skos:prefLabel xml:lang="en">General and reference</skos:prefLabel>
  <skos:altLabel xml:lang="en">general and reference works</skos:altLabel>
  <skos:inScheme rdf:resource="http://totem.semedica.com/taxonomy/The ACM
Computing Classification System (CCS)"/>
  <skos:prefLabel xml:lang="en">Database design and models</skos:prefLabel>
  <skos:altLabel xml:lang="en">database model</skos:altLabel>
  <skos:altLabel xml:lang="en">database models</skos:altLabel>
  <skos:altLabel xml:lang="en">data model</skos:altLabel>
  <skos:altLabel xml:lang="en">database design</skos:altLabel>
  <skos:related rdf:resource="#10010111"/>
  <skos:inScheme rdf:resource="http://totem.semedica.com/taxonomy/The ACM
Computing Classification System (CCS)"/>
  <skos:broader rdf:resource="#10002952"/>
  <skos:narrower rdf:resource="#10002955"/>
  <skos:narrower rdf:resource="#10002959"/>
  <skos:narrower rdf:resource="#10010146"/>
  <skos:narrower rdf:resource="#10010819"/>
  <skos:narrower rdf:resource="#10010820"/>
</skos:Concept>
- <skos:Concept xml:lang="en" rdf:about="#10002955">
  <skos:prefLabel xml:lang="en">Relational database model</skos:prefLabel>
  <skos:altLabel xml:lang="en">relational data models</skos:altLabel>
  <skos:altLabel xml:lang="en">relational data model</skos:altLabel>
  <skos:altLabel xml:lang="en">relational database models</skos:altLabel>
  <skos:inScheme rdf:resource="http://totem.semedica.com/taxonomy/The ACM
Computing Classification System (CCS)"/>
  <skos:broader rdf:resource="#10002953"/>
</skos:Concept>
- <skos:Concept xml:lang="en" rdf:about="#10002956">
  <skos:prefLabel xml:lang="en">Hierarchical data models</skos:prefLabel>
  <skos:altLabel xml:lang="en">hierarchical data model</skos:altLabel>
  <skos:altLabel xml:lang="en">hierarchical database models</skos:altLabel>
  <skos:altLabel xml:lang="en">hierarchical database model</skos:altLabel>
  <skos:inScheme rdf:resource="http://totem.semedica.com/taxonomy/The ACM
Computing Classification System (CCS)"/>
  <skos:broader rdf:resource="#10010146"/>

```



```

- <skos:Concept xml:lang="en" rdf:about="#10002974">
  <skos:prefLabel xml:lang="en">Multidimensional range search</skos:prefLabel>
  <skos:altLabel xml:lang="en">range search in multidimensional data</skos:altLabel>
  <skos:altLabel xml:lang="en">multi-dimensional range search</skos:altLabel>
  <skos:altLabel xml:lang="en">multi dimensional range search</skos:altLabel>
  <skos:altLabel xml:lang="en">range search in multi-dimensional data</skos:altLabel>
  <skos:altLabel xml:lang="en">range search in multi dimensional data</skos:altLabel>
  <skos:altLabel xml:lang="en">multidimensional range searching</skos:altLabel>
  <skos:altLabel xml:lang="en">multi dimensional range searching</skos:altLabel>
  <skos:altLabel xml:lang="en">multi-dimensional range searching</skos:altLabel>
  <skos:altLabel xml:lang="en">two-dimensional range search</skos:altLabel>
  <skos:altLabel xml:lang="en">2-dimensional range search</skos:altLabel>
  <skos:altLabel xml:lang="en">2-dimensional range searching</skos:altLabel>
  <skos:altLabel xml:lang="en">two-dimensional range searching</skos:altLabel>
  <skos:inScheme rdf:resource="http://totem.semedica.com/taxonomy/The ACM
    Computing Classification System (CCS)"/>
  <skos:broader rdf:resource="#10003450"/>
</skos:Concept>
- <skos:Concept xml:lang="en" rdf:about="#10002975">
  <skos:prefLabel xml:lang="en">Data compression</skos:prefLabel>
  <skos:altLabel xml:lang="en">source coding</skos:altLabel>
  <skos:altLabel xml:lang="en">bit-rate reduction</skos:altLabel>
  <skos:inScheme rdf:resource="http://totem.semedica.com/taxonomy/The ACM
    Computing Classification System (CCS)"/>
  <skos:broader rdf:resource="#10003451"/>
  <skos:broader rdf:resource="#10010031"/>
</skos:Concept>
- <skos:Concept xml:lang="en" rdf:about="#10002976">
  <skos:prefLabel xml:lang="en">Data encryption</skos:prefLabel>
  <skos:altLabel xml:lang="en">encryption of data</skos:altLabel>
  <skos:related rdf:resource="#10002979"/>
- <skos:Concept xml:lang="en" rdf:about="#10010418">
  <skos:prefLabel xml:lang="en">Enterprise architecture management</skos:prefLabel>
  <skos:altLabel xml:lang="en">management of enterprise architectures</skos:altLabel>
  <skos:altLabel xml:lang="en">managing enterprise architectures</skos:altLabel>
  <skos:altLabel xml:lang="en">management of enterprise architecture</skos:altLabel>
  <skos:inScheme rdf:resource="http://totem.semedica.com/taxonomy/The ACM
    Computing Classification System (CCS)"/>
  <skos:broader rdf:resource="#10010417"/>
</skos:Concept>
- <skos:Concept xml:lang="en" rdf:about="#10010419">
  <skos:prefLabel xml:lang="en">Enterprise architecture frameworks</skos:prefLabel>
  <skos:altLabel xml:lang="en">enterprise architecture framework</skos:altLabel>
  <skos:inScheme rdf:resource="http://totem.semedica.com/taxonomy/The ACM
    Computing Classification System (CCS)"/>
  <skos:broader rdf:resource="#10010417"/>
</skos:Concept>
- <skos:Concept xml:lang="en" rdf:about="#10010420">
  <skos:prefLabel xml:lang="en">Enterprise architecture modeling</skos:prefLabel>
  <skos:altLabel xml:lang="en">enterprise architecture modelling</skos:altLabel>
  <skos:inScheme rdf:resource="http://totem.semedica.com/taxonomy/The ACM
    Computing Classification System (CCS)"/>
  <skos:broader rdf:resource="#10010417"/>
</skos:Concept>
- <skos:Concept xml:lang="en" rdf:about="#10010421">
  <skos:prefLabel xml:lang="en">Service-oriented architectures</skos:prefLabel>
  <skos:altLabel xml:lang="en">service-oriented architecture</skos:altLabel>
  <skos:inScheme rdf:resource="http://totem.semedica.com/taxonomy/The ACM
    Computing Classification System (CCS)"/>
  <skos:broader rdf:resource="#10010406"/>

```

Bibliographie

- [1] Paquette G. Modélisation des Connaissances et des Compétences : un langage graphique pour concevoir et apprendre. Presses de l'université du Québec. (2002).
- [2] Psyché V. Rôle des ontologies en ingénierie des EIAH : Cas d'un système d'assistance au design pédagogique. Thèse de doctorat, Université du Québec-Montréal. (2007)
- [3] Gruber T. Toward principals for the design of ontologies used for knowledge sharing. In Formal Analysis in Conceptual Analysis and Knowledge Representation. (1993).
- [4] Eliane C. Qu'est-ce qu'une ontologie, Article pour le journal du net (03/04/2007)
- [5] Gomez-Perez, A. Ontological Engineering: a State of the Art. Expert Update, (1999) p 33-43.
- [6] Gruber, T. A translation approach to portable ontologyspecification. Knowledge acquisition, 5,(1993),p 199-220.
- [7] Belhadi, A. Aissat, M. Conception et réalisation d'une ontologie pour une structure de recherche avec protégé 2000. USTHB (2009).
- [8] Audrey B, CONSTRUIRE UNE ONTOLOGIE DE LA PNEUMOLOGIE. Cycle de vie d'une ontologie, 1 , (2007), p 38.
- [9] Beckett D. (rédacteur), recommandation du W3C, (10 février 2004). Cette version est à <http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/>.
- [10] Benayache, A. Construction d'une memoire organisationnelle. De formation et evaluation dans un contexte e-learning: le projet Memora. Université De Technologie de Compiègne. Spécialité: Technologies de l'Information et des Systèmes (TIS). (2005).
- [11] Samson T. Stanford Univesity . Cette version est à <http://protege.stanford.edu/>.
- [12] Furst F. Contribution à l'ingénierie des ontologies : une méthode et un outil d'opérationnalisation. Thèse de doctorat, Université de Nantes. (2004). 184 pages.
- [13] Noy N. Semantic Integration : A Survey of Ontology-based Approaches, SIGMOD, 2004. OAEI W., « OAEI 2006 : Ontology Alignment Evaluation Initiative », Proceedings of International Workshop on Ontology Matching, cette version est à <http://om2006.ontologymatching.org/>, (2006).
- [14] Shvaiko P., Euzenat J., A Survey of Schema-based Matching Approaches, Journal on Data Semantics, (2005).

- [15] Bach T.L. Construction d'un Web sémantique multi-points de vue , Thèse de doctorat Informatique, Ecole des Mines de Paris a Sophia Antipolis, 23 octobre (2006).
- [16] A Survey of Approaches to Automatic Schema Matching. VLDB Journal, 10(4):334–350, (2001).
- [17]Euzenat J., Ehrig M., Jentzsch A., Mochol M., Shvaiko P., “Case-based recommendation of matching tools and techniques, deliverable 1.2.2.2.1”, Knowledge Web, (Mars 2007).
- [18] AbdeltifM, 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).
- [19] Mme Ghomari Leïla (Née Zemmouchi), Alignement d'ontologies de domaine : Etude comparative syntaxique 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)
- [20] Jaccard P., Distribution de la flore alpine dans le bassin des Dranses et dans quelques régions voisines. Bulletin de la Société Vaudoise des Sciences Naturelles, Vol. 37, pp. 241-272 (1901).
- [21] Real R., and Vargas J.M. The Probabilistic Basis of Jaccard's Index of Similarity. SystemeBiology, Vol. 45 (3), pp. 385-390 (1996). Cette version est à: <http://bioinfo-fr.net/lindice-de-jaccard-pas-quune-histoire-de-mailles#sthash.3FD9KSKD.dpuf>
- [22] Advances in record linking methodology as applied to the 1985 census of Tampa Florida », *Journal of the American Statistical Society*, vol. 84, n° 406,1989, p. 414-420
- [23] Miller, G. WordNet: A lexical database for english. communication of the ACM. (1995),pp. 39-41.
- [24] Z. Wu et M. Palmer.1994. *Verbsemantics and lexical selection*. In Proceedings of the 32nd Annual Meeting of the Associations for ComputationalLinguistics, p 133-138.
- [25] Naçima M, 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).

[26] Bray T, Paoli J, Sperberg-McQueen C. M, Maler E, Yergeau F et Cowan J. Extensible MarkupLanguage (XML) 1.1. <http://www.w3.org/TR/2004/RECxml11-20040204/> (en ligne au 16 juin 2005).

[27] Samson T. Stanford University. Cette version est à <http://protege.stanford.edu/>.