

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA  
RECHERCHE SCIENTIFIQUE  
UNIVERSITE SAAD DAHLEB- BLIDA 1

---



Faculté des Sciences  
Département d'informatique

---

Mémoire de fin d'étude  
Pour l'obtention du diplôme de mastère en informatique  
**Option : Ingénierie des Logiciels**

Thème

**Enrichissement d'ontologie à partir des Données Liées**

**Réalisé par :**

BOUDISSA Razika

**Encadré par :**

M<sup>me</sup> BERRAMDANE Djamila

Soutenu le : 11/07/2023 devant le jury composé de :

- Président : Dr. Mahfoud Bala
- Examineur : Dr. Ishak Riali

Année universitaire 2022-2023

# *Remerciement*

Je tiens tout d'abord à exprimer ma reconnaissance à Allah le Tout-Puissant pour m'avoir accordé la force et les moyens nécessaires pour mener à bien ce mémoire.

Je souhaite exprimer mes vifs remerciements et ma profonde gratitude à ma promotrice, Mme BERRAMDANE Djamila, qui a dirigé ce mémoire. Sa bienveillance, sa confiance et ses précieux conseils m'ont permis de progresser tout au long de ce travail. Je tiens à la remercier pour sa guidance et sa patience qui ont accompagné mon parcours.

Un grand merci également aux membres du jury d'avoir accepté de faire partie de la commission d'examen et d'évaluer mon travail.

Enfin, je souhaite exprimer ma gratitude envers toutes les personnes qui m'ont apporté leur aide de près ou de loin dans l'élaboration et la finalisation de ce travail. Leur soutien et leurs encouragements ont été précieux et ont contribué à la réalisation de ce mémoire.

# *Dédicace*

Je dédie ce modeste travail à ma fille Asmaa,

à mes deux sœurs,

et à toute ma famille et mes amis.

## **Résumé**

Les ontologies sont l'épine dorsale du web sémantique, elles décrivent les connaissances d'un domaine donné à l'aide de concepts et de relations entre ces concepts. Cependant, la dynamique et l'évolution du domaine qu'elles modélisent constitue un défi majeur dans lequel ces ontologies doivent s'adapter. L'enrichissement des ontologies, une branche de l'ingénierie des ontologies, vise à diminuer l'effort fourni dans la création des ontologies à partir de scratch, en ajoutant des concepts et/ou relations à des ontologies existantes.

Nous présentons dans ce mémoire une méthode d'enrichissement d'une ontologie à partir des données liées. L'approche proposée consiste à extraire des concepts à l'aide des requêtes SPARQL, une étape de raffinement des concepts obtenus pour sélectionner les plus pertinents à fin enrichir l'ontologie initiale qui sera par la suite validée formellement.

**Mots clés** : Données liées, ontologie, SPARQL, enrichissement d'ontologie.

## Abstract

Ontologies represent the backbone of the semantic web, aim to describe the knowledge of a specific domain through concepts and the relationships between them. However, the dynamic nature and evolution of the specific domain present a major challenge that these ontologies must adapt to. Ontology enrichment, a branch of ontology engineering, aims to reduce the effort required in building ontologies from scratch by adding concepts and/or relationships to existing ontologies. In this thesis, we present a method for enriching an existing ontology using linked data. The proposed approach involves extracting concepts using SPARQL queries, refining the obtained concepts to select the most relevant ones, and subsequently enriching the initial ontology, which will be formally validated.

**Keywords:** Linked data, Ontology, SPARQL, ontology enrichment.

## ملخص

الأنطولوجيا تشكل العمود الفقري للويب الدلالي، حيث تصف المعرفة في مجال محدد من خلال المفاهيم والعلاقات بين تلك المفاهيم. ومع ذلك، فإن ديناميكية وتطور المجال الذي تمثله هذه الأنطولوجيا تشكل تحديًا كبيرًا يجب على هذه الأنطولوجيا التكيف معه. يهدف إثراء الأنطولوجيا، والذي يعتبر فرع من فروع هندسة الأنطولوجيا، إلى تقليل الجهد المبذول في إنشاء الأنطولوجيا من الصفر عن طريق إضافة مفاهيم و/أو علاقات إلى الأنطولوجيا الحالية. في هذا العمل، نقدم طريقة لإثراء أنطولوجيا باستخدام البيانات المرتبطة. يتضمن النهج المقترح استخراج المفاهيم باستخدام استعلامات SPARQL، وبتقنية المفاهيم المحصل عليها لاختيار أكثرها صلة، ومن ثم إثراء الأنطولوجيا الأولية التي ستم التحقق منها رسميًا.

**الكلمات الرئيسية:** البيانات المرتبطة، الأنطولوجيا، SPARQL، إثراء الأنطولوجيا.

# Sommaire

Introduction Générale .....	1
-----------------------------	---

## Chapitre 1: Web Sémantique et Linked Data

1. Introduction .....	4
2. Web Sémantique .....	4
2.1. Architecture du web sémantique .....	5
2.2. Langages du Web sémantique.....	6
2.2.1 XML .....	6
2.2.2 RDF .....	7
2.2.3 RDFS .....	7
2.2.4 DAML + OIL .....	8
2.2.5 OWL .....	8
2.2.6 SPARQL.....	9
3. Les Ontologies.....	11
3.1. Définition des Ontologies .....	11
3.2. Composants d'une Ontologie .....	11
3.3. Caractéristiques.....	13
3.4. Classification .....	13
4. Données Liées .....	14
4.1. Définition de Données Liées .....	14
4.2. Principes des Données liées.....	15
4.3. Système de notation des Données liées .....	16
5. Conclusion.....	17

## Chapitre 2: Etat de l'art sur les Méthodes d'enrichissement

1. Introduction .....	19
2. L'enrichissement d'une ontologie .....	19
3. Critères de classification des méthodes d'enrichissement d'ontologies .....	20
5. Comparaison de méthodes d'enrichissement .....	24
6. Conclusion.....	25

## **Chapitre 3: Conception du système d'enrichissement d'Ontologie**

1. Introduction .....	27
2. L'approche proposée pour l'enrichissement d'Ontologies .....	27
2.1. Sélection des concepts candidats .....	28
2.2. Extraction des concepts depuis LOD.....	28
2.3. Placement des Concepts dans L'Ontologie .....	31
2.3.1. Raffinage par les mesures de similarité .....	32
2.3.2. Placement des nouveaux concepts dans l'ontologie.....	33
2.4. Vérification de consistance de l'ontologie enrichie .....	34
3. Conclusion.....	35

## **Chapitre 4: Implémentation et validation**

1. Introduction .....	37
2. Implémentation.....	37
3. Présentation de l'application .....	40
4. Validation .....	44
5. Conclusion.....	46
Conclusion et perspectives .....	47
Références .....	49

## Table des figures

Figure 1.1 : Couches du Web Sémantique. [5].....	6
Figure 1.2 : Graphe RDF. ....	6
Figure 1.3 : Les familles du langage OWL. ....	8
Figure 1.4 : Structure simple d'une requête SPARQL. ....	9
Figure 1.5 : Un exemple d'ontologie simple. ....	12
Figure 1.6 : Ensembles de données publiés au format Linked Data.....	15
Figure 2.1 : Le processus général d'enrichissement d'ontologie. ....	20
Figure 3.1 : Processus général de système d'enrichissement d'ontologies .....	27
Figure 3.2 : Logo de DBpedia .....	29
Figure 3.3 : Exemple de classe et subclassOf. ....	34
Figure 4.1 : L'ontologie Géo utilisée pour l'enrichissement .....	40
Figure 4.2 : La première fenêtre de l'application .....	41
Figure 4.3 : Exemple de requête SPARQL exécutée.....	41
Figure 4.4 : Les résultats de la requête SPARQL exécutée.....	42
Figure 4.5 : Les nouvelles classes ajouter à l'ontologie .....	43
Figure 4.6 : Messages d'indication pour une requête sans résultat.....	43
Figure 4.7 : Messages d'indication pour une requête invalide.....	44
Figure 4.8 : L'interface du site validateur de 3WC.....	45
Figure 4.9: Résultat de la validation.....	45
Figure 4.10: L'ontologie enrichie sous protégé.....	46

## Liste des tableaux

Tableau 2.1 : Comparaison de méthodes d'enrichissement. ....	24
Tableau 3.1 : Signification de quelques préfixes.....	30

# **Introduction Générale**

Le terme "Données Liées" fait référence à un ensemble de meilleures pratiques pour la publication et la connexion de données structurées sur le Web. Ces meilleures pratiques ont été adoptées par un nombre croissant de fournisseurs de données au cours des dernières années, ce qui a conduit à la création d'un espace de données global contenant des milliards d'assertions - le web de données [1]. Ces assertions sont généralement construites à partir de l'extraction de triplets à partir de ressources non structurées déjà existantes. L'automatisation de ce processus utilise des outils linguistiques pour manipuler des ressources de grande taille et de quantité d'information énorme dans la plupart des cas. Un des exemples célèbres est celui de DBpedia<sup>1</sup>, qui est une source de données liées extraites des articles de Wikipédia<sup>2</sup>, considérées comme des ressources non structurées [2].

Les ontologies, l'un des principaux piliers du web sémantique, consistent à décrire de manière explicite les connaissances d'un domaine donné à travers des concepts et les relations entre ces concepts. L'un des défis majeurs pour la réussite du web sémantique est d'automatiser leur création, qui reste jusqu'à présent un travail fastidieux.

Durant la dernière décennie, plusieurs ontologies ont été créées d'une manière manuelle ou (semi) automatique à travers des méthodologies et outils fournis par les efforts de la communauté de recherche de l'ingénierie des ontologies. Parmi les défis qui font face à ces ontologies leur domaine de définition qui n'est pas statique. Les ontologies sont souvent utilisées dans des environnements dynamiques, elles ne peuvent pas être pensées comme une conceptualisation finie d'un domaine de connaissance délimité et stable [3].

Le web de données liées constitue un autre défi pour le développement des ontologies en raison de l'évolution incrémentale de ces sources de données et de leur schéma ou vocabulaire associé (plus de 1255 sources de données avec 16174 liens<sup>3</sup>). Cela implique la nécessité d'apporter des modifications ou des mises à jour à l'ontologie afin de prendre en compte cette évolution. Effectuer ces mises à jour manuellement demeure une tâche coûteuse et fastidieuse, car elle requiert la mobilisation d'un ou plusieurs experts du domaine pour identifier et classer les nouveaux éléments de vocabulaire dans l'ontologie.

---

<sup>1</sup> <http://dbpedia.org/>

<sup>2</sup> <https://fr.wikipedia.org/>

<sup>3</sup> <https://lod-cloud.net/> 17-06-2023

L'enrichissement des ontologies constitue une branche de l'ingénierie des ontologies qui vise à étendre une ontologie existante par l'ajout de concepts, de relations sémantiques ou encore d'instances. Peu d'approches d'enrichissement d'ontologies exploitent les données liées en tant que source de données structurées. La plupart des approches proposées dans la littérature utilisent plutôt les sources textuelles comme source d'enrichissement.

L'objectif de ce mémoire consiste à proposer une approche d'enrichissement d'ontologie à partir des données liées. Notre approche exploite les données liées ouvertes, plus précisément DBpedia, en tant que source de données d'enrichissement. Les concepts candidats de l'ontologie initiale constituent le corps des requêtes SPARQL. L'application de mesures de similarité permet de raffiner l'ensemble des concepts obtenus en ne conservant que les plus pertinents pour enrichir l'ontologie de départ. Une étape de validation est nécessaire pour tester la consistance de l'ontologie finale.

Ce mémoire est organisé en quatre chapitres :

– **Chapitre 1 : Web Sémantique et Linked Data :**

Ce chapitre présente des détails sur le Web sémantique, les ontologies et les données liées, constituant ainsi une base pour le présent travail.

– **Chapitre 2 : Etat de l'art sur les méthodes d'enrichissement :**

Dans ce chapitre, nous examinerons différents travaux sur l'enrichissement des ontologies à partir de données liées.

– **Chapitre 3 : Conception du Système d'enrichissement d'ontologies**

Ce chapitre met en avant notre système proposé pour l'enrichissement d'ontologie, en expliquant les principales étapes de notre approche proposée.

– **Chapitre 4 : Implémentation et validation**

Dans ce chapitre, nous décrivons les détails l'implémentation de notre système, en mettant en avant les outils et les méthodes utilisés. De plus, nous présentons les tests réalisés pour valider notre travail.

la conclusion de ce mémoire synthétise les principales propositions dans notre systèmes et dégage quelques perspectives de ce travail.

# **Chapitre 1**

## **Web Sémantique et Linked Data**

## 1. Introduction

Bonjour, j'espère que vous allez bien. J'essaie de faire des modifications sur le mémoire selon les remarques des membres du jury, mais il n'y a pas de grands changements. Si vous avez des remarques et que vous souhaitez me les faire part, je voulais effectuer ces modifications maintenant. J'ai peur d'oublier les remarques d'ici le mois de septembre.

Ces dernières années, le développement impressionnant de l'Internet a engendré l'apparition d'une énorme quantité de données disponibles sur le Web. Cette richesse et cette croissance exponentielle de données promettent d'être une véritable mine d'or pour les entreprises. Toutefois, cette croissance des informations présente également des défis si les données ne sont pas bien structurées et représentées. Cela souligne les défis auxquels le Web est confronté en ce qui concerne la recherche, le partage, l'accès et la réutilisation de ces données, conformément aux besoins des utilisateurs. Dans cette optique, le Web sémantique émerge comme une nouvelle génération du Web qui vise à répondre à ces exigences.

Dans ce chapitre, nous allons nous concentrer sur le Web sémantique et les données liées.

## 2. Web Sémantique

L'idée du Web sémantique est de modéliser le contenu des ressources du Web en ajoutant de la sémantique sous forme de métadonnées en vue de rendre les ressources compréhensibles par des machines [4]. Autrement dit, il s'agit de décrire ces ressources selon une représentation formelle avec une sémantique clairement définie et qui soit conçue pour une interprétation par des programmes. La base de l'infrastructure du Web sémantiques s'appuie sur l'explicitation de la conceptualisation d'un domaine, partagée par une communauté et représentée dans une ontologie du domaine concerné.

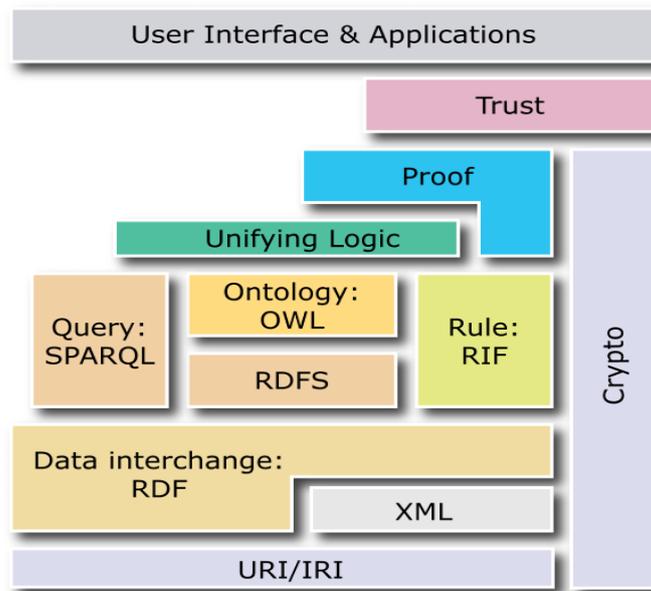
L'objectif du Web sémantique est de rendre explicite le contenu sémantique des ressources dans le Web (documents, pages web, services, etc.). Les machines et les agents logiciels pourraient "comprendre" les contenus décrits dans les ressources et faciliter les tâches de traitement des informations de façon plus automatique et plus efficace. Les recherches actuellement réalisées dans le domaine du Web Sémantique s'appuient sur un existant riche provenant de différents domaines.

Le Web sémantique est non seulement appliqué dans les recherches intelligentes d'information, mais il est aussi étudié dans les recherches sur l'ingénierie des connaissances, les systèmes de représentation des connaissances, le traitement automatique de la langue naturelle, l'apprentissage, les agents intelligents, le raisonnement automatique, etc.

## 2.1. Architecture du web sémantique

Selon la proposition de Tim Berners-Lee le créateur du World Wide Web, le Web sémantique peut être vu comme une architecture en couches, dont les principales sont [5] :

- **Couche de l'identification** : Cette couche s'occupe de l'attribution d'identifiants uniques aux ressources sur le Web grâce à l'utilisation d'URI.
- **Couche de description** : La couche de description est basée sur le modèle de données RDF, qui permet de représenter les connaissances et les données sous forme de triplets sujet-prédicat-objet. Cela offre une base pour décrire les ressources et leurs relations.
- **Couche de la découverte** : La couche de découverte facilite l'exploration et l'accès aux ressources et aux connaissances sur le Web sémantique. Elle utilise des technologies telles que SPARQL pour permettre aux utilisateurs de formuler des requêtes et de récupérer des informations précises.
- **Couche de la logique** : La couche de logique permet l'inférence et la déduction à partir des données existantes en appliquant des règles logiques et des ontologies. Cela permet d'étendre les connaissances et de découvrir des informations implicites.



**Figure 1.1** : Couches du Web Sémantique. [5]

## 2.2. Langages du Web sémantique

Dans le contexte du Web Sémantique, plusieurs langages ont été développés. Nous allons présenter brièvement certains langages principaux RDF, RDF Schéma, OWL, et SPARQL.

### 2.2.1. XML <sup>4</sup>

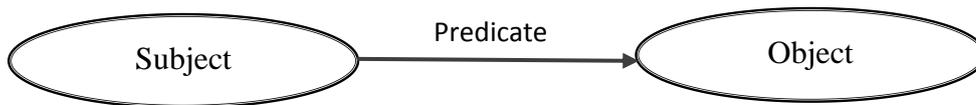
La syntaxe de XML (eXtensible Markup Language) issu de SGML (Standard Generalized Markup Language) et défini par le consortium web. Le langage XML est un langage de balisage, il va permettre de structurer les informations contenues dans un document grâce à leurs balises. Contrairement à HTML, XML offre la possibilité de créer ses propres balises.

Le but de XML n'est pas le développement d'ontologie (il ne permet pas d'imposer des contraintes sémantiques) mais il est considéré comme un langage de description et d'échange de documents structurés. En d'autres termes, XML traite la structure syntaxique des documents sans aucune spécification sémantique de cette structuration. Le langage RDF (Resource Description Framework) est un langage qui a été défini pour cette raison.

<sup>4</sup> <http://www.w3.org/XML/>

### 2.2.2. RDF <sup>5</sup>

RDF (Resource Description Framework) est un standard utilisé pour l'échange des ressources sur le web. Le W3C a adopté le langage RDF comme une recommandation en 1999. Les ressources dans le RDF sont identifiées par les URIs (Uniform Resources Identifiers). Un document RDF est à la base exprimé sous la forme d'un graphe orienté, mais on peut le décrire par la syntaxe XML. Il fournit une sémantique simple pour des données sans faire des contraintes sur la structure du document.



**Figure 1.2 :** Graphe RDF.<sup>6</sup>

Les graphes RDF( figure 1.2) sont des ensembles de triplets sujet-prédicat-objet, où les éléments peuvent être des URI (Uniform Resources Identifiers), des nœuds anonymes ou des littéraux typés. Ils sont utilisés pour exprimer des descriptions de ressources. Cependant, Il existe certaines limites pour RDF; par exemple, nous avons besoin de connaître des informations sur les ressources identifiées. C'est pour cela que le W3C a associé à RDF le standard RDF Schéma (RDFS).

### 2.2.3. RDFS <sup>7</sup>

RDFs (Resource Description Framework Schema ) Construit par le W3C en 2000 comme une extension sémantique et fournit un vocabulaire de modélisation RDF. RDFS permet de définir des mécanismes d'héritage pour la description des groupes de ressources similaires (classes) et des relations entre ces ressources (propriétés). Comme son nom l'indique, le RDFS est un langage permettant de définir des propriétés sémantiques pour les ressources par un schéma. Dans un schéma on peut définir de nouvelles ressources comme des spécialisations d'autres ressources. Les schémas contraignent aussi le contexte d'utilisation des ressources.

---

<sup>5</sup> <http://www.w3.org/RDF/>

<sup>6</sup> <https://www.w3.org/TR/rdf12-concepts/>

<sup>7</sup> <http://www.w3.org/TR/rdf-schema/>

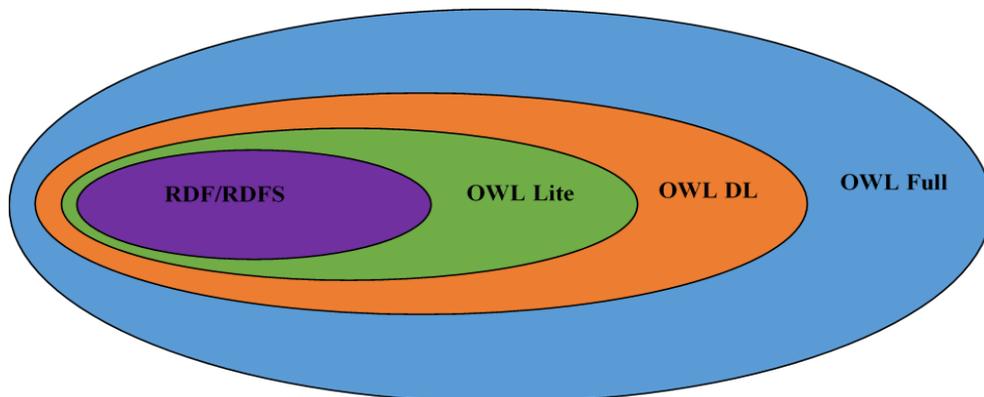
## 2.2.4. DAML + OIL <sup>8</sup>

Jim Hendler a annoncé le 11 janvier 2001 la première version du langage DAML (DARPA Agent Markup Language) appelé DAML-ONT. Les langages DAML et OIL (Ontology Interchange Language) se regroupent et forment le langage DAML+OIL qui remplace donc DAML-ONT. Il utilise la syntaxe des standards XML, RDF et RDFS. DAML+OIL est considéré comme le premier langage à fournir des mécanismes d'inférence sur les concepts d'ontologies. Mais ce langage présente des limites (il ne permet pas la composition et la fermeture transitive. Il prend en compte uniquement des relations unaires et binaires. Il ne possède pas des déclarations de variables.

## 2.2.5. OWL <sup>9</sup>

OWL (OntologyWeb Language) est une extension du langage RDF et RDFS, inspiré de DAML+OIL. Il définit un vocabulaire riche pour la description d'ontologies complexes. Le World Wide Web Consortium (W3C) a recommandé le langage OWL qui est conçu pour la représentation des ontologies pour le Web sémantique basé sur les logiques de description.

Le langage d'ontologies OWL se décompose en trois sous-langages : OWL Lite, OWL DL et OWL Full d'après la complexité croissante et les différentes utilisations.



**Figure 1.3** : Les familles du langage OWL. [6]

<sup>8</sup> <http://www.daml.org/about.html>

<sup>9</sup> <http://www.w3.org/TR/owl-ref/>

- **OWL-Lite (Ontology Web Language Lite)** : Il est recommandé pour la représentation des thésaurus, des taxonomies ou hiérarchie de classes avec des contraintes simples. Il reprend tous les constructeurs de RDF/XML.

- **OWL-DL (Ontology Web Language-Description Logic)** : Il est approprié pour la représentation des ontologies ayant besoin de la puissance d'expressivité tout en gardant la calculabilité.

- **OWL-Full** : est conçu pour les développeurs et les utilisateurs qui ont besoin de l'expressivité maximale, de la liberté syntaxique de RDF mais sans se soucier d'une garantie de calculabilité.

Les trois sous-langages d'OWL présentent une hiérarchie sur la validité des ontologies suivantes : les ontologies OWL-Lite sont également des ontologies OWL-DL. Et les ontologies OWL-DL sont également des ontologies OWL-Full. (Voir Figure 1.3).

## 2.2.6 SPARQL

SPARQL (Sparql Protocol and RDF Query Language) est le langage standard pour interroger les données représentées par des triplets, C'est l'un des trois normes fondamentales du Web sémantique, avec RDF et OWL. [7].

- **Structure d'une requête SPARQL**

La syntaxe d'une requête SPARQL est inspirée du SQL, une requête est un graphe RDF (graphes de triplets : sujet, prédicat, objet) avec des variables, où les ressources sont présentées par des URI abrégées par des préfixes. Et les objets peuvent être : littéraux, chaînes de caractères, booléens, entiers. SPARQL recherche des valeurs des variables qui sont des sous-graphes du graphe représentant les données, la figure 1.4 présente la structure d'une requête SPARQL [8].

```
PREFIX prefix1: <URI1>
PREFIX prefix2: <URI2>
...

SELECT ?variable1 ?variable2 ...
WHERE {
?sujet prédicat1 ?variable1 .
?sujet prédicat2 ?variable2 .
...
FILTRE(optionnel) ...
}
ORDER BY (optionnel) ...
LIMIT (optionnel) ...
OFFSET (optionnel) ...
```

**Figure 1.4** : Structure Simple d'une requête SPARQL.

- PREFIX: Cette clause est utilisée pour définir des préfixes qui simplifient l'écriture des URI dans la requête.
- SELECT: Cette clause spécifie les variables qu'on souhaite récupérer dans le résultat de la requête. Les variables commencent par un point d'interrogation "?".
- WHERE: Cette clause spécifie les conditions de recherche dans le graphe RDF. On peut utiliser des triplets RDF pour spécifier les motifs à rechercher. Les variables non définies dans la clause SELECT peuvent être utilisées pour récupérer des informations supplémentaires sur les résultats.
- FILTRE: Cette clause est utilisée pour appliquer des conditions de filtrage optionnelles sur les résultats.
- ORDER BY: Cette clause est utilisée pour trier les résultats selon une ou plusieurs variables.
- LIMIT: Cette clause est utilisée pour limiter le nombre de résultats renvoyés par la requête.
- OFFSET: Cette clause est utilisée pour spécifier le décalage à partir duquel les résultats doivent être renvoyés.

### 3. Les Ontologies

#### 3.1. Définition des Ontologies

Le mot "ontologie" tire son origine du grec "ontos" qui signifie "être" et de "logos" qui se réfère à l'univers. Il s'agit d'un terme philosophique introduit au 19<sup>ème</sup> siècle pour désigner l'étude des entités présentes dans notre univers, c'est-à-dire une exploration des propriétés générales de ce qui existe. Cependant, dans le domaine de l'informatique, le terme "ontologie" fait référence à un ensemble structuré de concepts.

En 1993, Gruber donne la définition: « *Une ontologie est une spécification explicite d'une conceptualisation* » [9].

Le terme « *conceptualisation* » réfère dans cette définition à une abstraction d'un phénomène du monde, obtenue en identifiant les concepts appropriés à ce phénomène. « *Formelle* » indique que les ontologies sont interprétables par la machine. Cependant, « *Spécification explicite* » signifie que les concepts de l'ontologie et les contraintes liées à leur usage sont définies de façon déclarative.

Plusieurs définitions ont été proposées pour le terme "ontologie", notamment celle de Swartout en 1997 [10] et celle de Studer en 1998 [11]. Cependant, la définition de Gruber est devenue la plus largement citée et la plus concise.

#### 3.2. Composants d'une Ontologie

Les composantes d'une ontologie sont au nombre de cinq : des concepts, des propriétés, des relations, des axiomes et des instances [12].

- **Les concepts**

Un concept peut représenter un objet, une idée, ou bien une notion abstraite. Il consiste aussi en un agrégat de caractéristiques que l'on peut percevoir comme étant commun à un certain nombre d'individus ou d'objets. Il peut être divisé en trois parties :

- Un terme (ou plusieurs) : est un élément lexical qui permet d'exprimer le concept en langue naturelle, il peut admettre des synonymes.
- La notion : également appelée intention du concept, contient la sémantique du concept, exprimée en termes de propriétés, d'attributs, et de contraintes.

- L'objet : appelé extension du concept, regroupe les objets manipulés à travers le concept, ces objets sont appelés instances du concept.

- **Les propriétés**

Appelée aussi attributs ou propriétés de données, elles sont utilisées pour qualifier un élément de l'ontologie (un concept ou une relation). Une propriété peut être atomique (par exemple, un entier) ou structurée (par exemple, une liste).

- **Les relations**

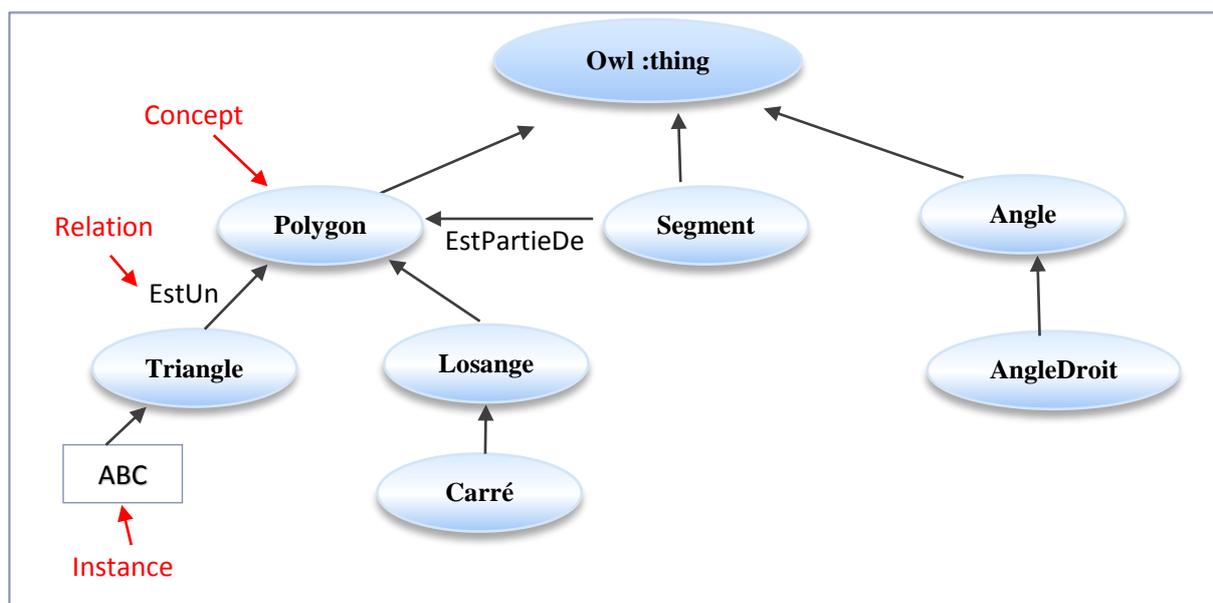
Les concepts peuvent être reliés entre eux par des relations au sein d'une ontologie. Elles représentent les interactions entre les concepts dans une ontologie.

- **Les axiomes**

Il est nécessaire d'intégrer des contraintes liées au domaine dans des situations existantes. Ces contraintes ne sont pas exprimables à travers les concepts et les relations, elles sont alors exprimées à travers des axiomes qui sont des expressions en langage logique ayant pour objectifs de définir des restrictions sur la valeur des propriétés, et définir les arguments d'une relation.

- **Les instances**

Appelée aussi des faits, représentent un individu ou un objet qui appartient à un concept.



**Figure 1.5 :** Un exemple d'ontologie simple. [13]

### 3.3. Caractéristiques

Une ontologie est une bibliothèque de concepts bien définie. Cette bibliothèque décrit la structure de l'information pour un domaine particulier. Les ontologies possèdent les caractéristiques fondamentales suivantes [12] :

- **La clarté et l'objectivité** : elles doivent fournir le sens des termes définis en offrant des définitions objectives en langage naturel.
- **La formalisation** : Les concepts et les relations sont exprimés dans une syntaxe claire et cohérente sur une base mathématique qui peut être traitée par des programmes informatiques.
- **La lisibilité** : Les concepts et les relations doivent être lisibles par les programmeurs, les communautés d'experts de domaine et aussi les utilisateurs potentiels.
- **L'exhaustivité** : Toutes les définitions des concepts ou des relations doivent être exprimées par des conditions nécessaires et suffisantes.
- **Les ontologies sont partageables** : Ils sont utilisables à travers de multiples domaines d'application pour faciliter la combinaison entre les ontologies développées.

### 3.4. Classification

Plusieurs dimensions ont été proposées pour la classification des ontologies. Nous sommes intéressés par la classification de Guarino [14] qui considère que les types d'ontologie suivants existent :

- **Les ontologies de haut niveau (Top-level ontologies)** : sont des ontologies générales qui décrivent les concepts très généraux et le plus haut niveau d'abstraction dans le monde, tel que: l'espace, le temps, la matière, les événements, les actions, etc. Elles sont indépendantes d'un problème ou d'un domaine d'application particulier.
- **Les ontologies de domaine (Domaine ontologies)** : décrivent le vocabulaire lié à un domaine d'application. Elles sont considérées comme un méta description d'une représentation des connaissances. Noter que, la plupart des ontologies existantes sont des ontologies du domaine.

- **Les ontologies de tâche (Task ontologies)** : toutes les tâches des résolutions de problèmes (diagnostic, planification...) dans un système ont été décrites par des ontologies de tâche. Elle régit un ensemble de vocabulaires et de concepts qui décrit une structure de résolution des problèmes.

- **Les ontologies d'application** : Elles décrivent les concepts appartenant simultanément à un domaine et à une tâche spécifique. Elles sont souvent une spécialisation des ontologies de domaine et de tâches et correspondent aux rôles joués par les entités de domaine lorsqu'ils effectuent certaines tâches.

## 4. Données Liées

### 4.1. Définition de Données Liées

Les données liées (Linked Data) sont un ensemble de principes de conception pour le partage de données lisibles par machine sur le Web pour une utilisation par les administrations publiques, les entreprises et les citoyens [15].

Les données ouvertes liées (LOD : Linked Open Data) sont un sous-ensemble des données liées qui sont également publiées selon les principes de l'ouverture des données. Cela signifie qu'elles sont librement accessibles, réutilisables et redistribuables par tous, sans restrictions de copyright, de brevet ou de licence.

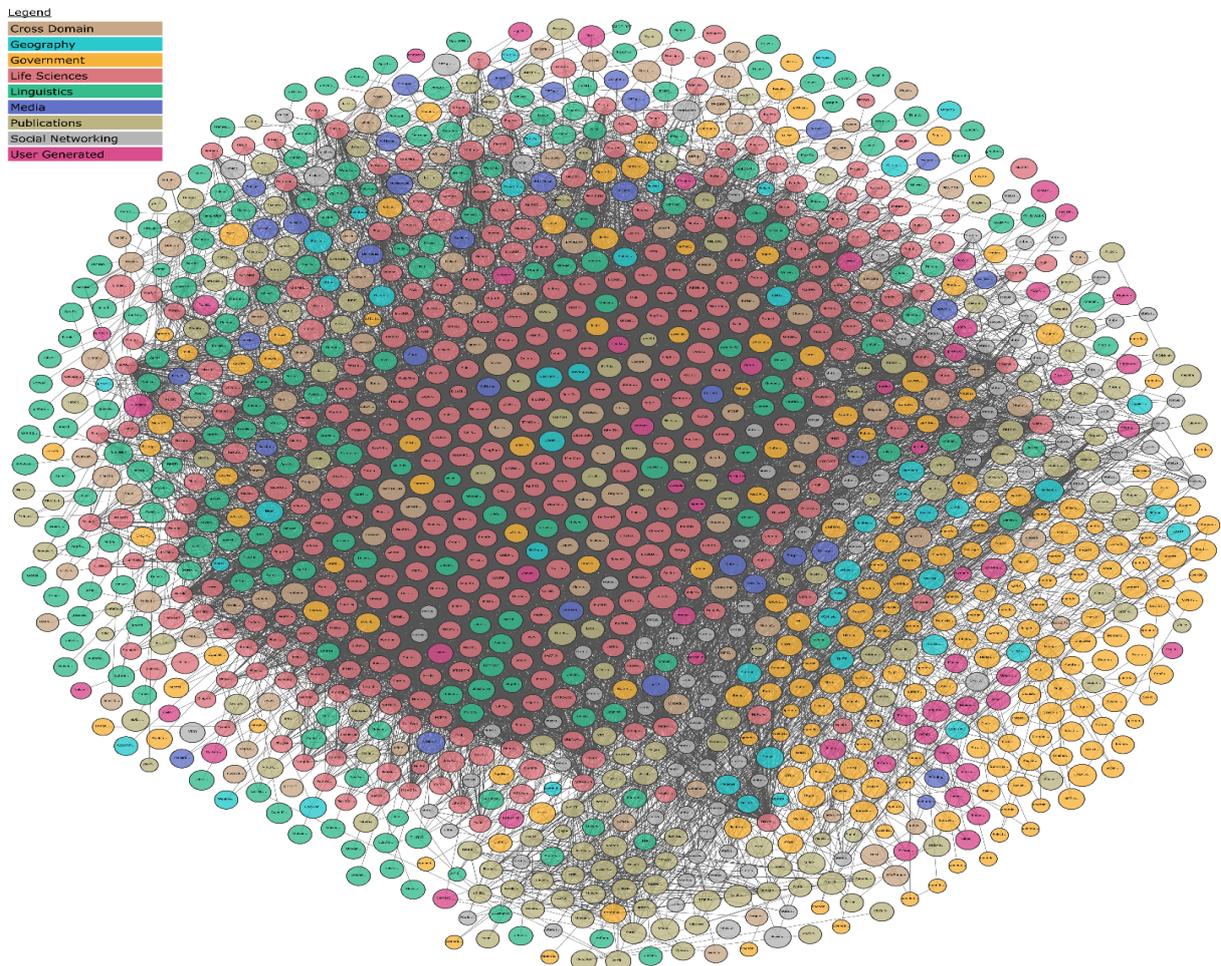
Les données liées ont connu une croissance significative ces dernières années, avec de plus en plus d'organisations, de gouvernements et de communautés publiant des ensembles de données au format lié. Cette croissance favorise une intégration et une interopérabilité accrues des données.

Les données liées sont utilisées dans divers domaines tels que la recherche scientifique, l'industrie, l'administration publique, la santé, l'éducation et le journalisme de données. Elles jouent un rôle essentiel dans la réalisation du Web sémantique et dans la création de systèmes d'informations plus intelligents. Des exemples bien connus de jeux de données liées incluent DBpedia (qui extrait les données de Wikipédia), LinkedGeoData (données géographiques) et LinkedMDB (données du cinéma), qui servent de références et démontrent le potentiel des données liées <sup>10</sup>.

---

<sup>10</sup> <https://www.w3.org/standards/semanticweb/data>

La figure 1.6 montre des ensembles de données publiés au format Linked Data et interconnectés avec d'autres ensembles de données dans le cloud prise le 23-11-2022.



**Figure 1.6** : Ensembles de données publiés au format Linked Data-Cloud le 23/11/2022- .

## 4.2. Principes des Données liées

Les principes des Données Liées sont un ensemble de bonnes pratiques pour publier, lier et interconnecter des données sur le Web de manière structurée et cohérente. Ces principes ont été formulés par Tim Berners-Lee, l'inventeur du World Wide Web, et visent à rendre les données plus accessibles, interopérables et utiles. Voici les principes des Données Liées [16] :

**1. Utiliser les identifiants URI (Uniform Resource Identifier) :** Les données liées sont identifiées de manière unique à l'aide d'URI, qui permettent d'accéder aux ressources et

d'établir des liens entre elles. Les URI servent à désigner les entités et les concepts représentés par les données.

**2. Utiliser les URI HTTP pour chercher les entités :** Les URI utilisés dans les données liées sont généralement des URI HTTP, ce qui signifie qu'ils peuvent être résolus par le protocole HTTP. Ainsi, lorsqu'un utilisateur accède à une URI, il peut effectuer une requête HTTP pour récupérer les informations associées à cette entité.

**3. Utiliser les standards du Web :** Les données liées sont publiées sur le Web en utilisant les protocoles et les formats standards tels que RDF (Resource Description Framework) et SPARQL (SPARQL Protocol and RDF Query Language). Cela assure l'accessibilité et l'interopérabilité des données.

**4. Établir des liens entre les données :** Les données liées sont interconnectées grâce à l'établissement de liens entre les URI. Cela permet de créer un réseau de données où chaque entité peut être liée à d'autres ressources pertinentes, favorisant ainsi la découverte et l'exploration des informations.

### **4.3. Système de notation des Données liées**

Il s'agit d'une proposition pour évaluer la qualité des données liées en fonction de leur niveau de publication et de réutilisabilité. Ce système, également introduit par Tim Berners-Lee, comprend les niveaux suivants [16] :

- **Données disponibles sur le Web (étoile 1) :** Les données sont disponibles sur le Web, accessibles via une URI.
- **Données structurées (étoile 2) :** Les données sont publiées dans un format structuré, tel que le RDF, permettant d'identifier les entités et leurs relations.
- **Données utilisant des URI (étoile 3) :** Les données utilisent des URI pour identifier les entités et établir des liens vers d'autres ressources.
- **Données liées à d'autres données (étoile 4) :** Les données sont interconnectées avec d'autres ensembles de données en utilisant des liens URI, permettant une exploration plus large du Web de données.

- **Données liées ouvertes (étoile 5)** : Les données liées sont disponibles avec une licence ouverte, permettant leur réutilisation gratuite par tous.

Notre travail proposé ne dépend pas de l'ouverture des données, mais il dépend du fait que les données soient accessibles via un point d'accès SPARQL. Dans le cas des Linked Open Data, cela nécessite qu'elles aient au moins 4 étoiles LOD.

## 5. Conclusion

Nous venons de présenter la notion du Web sémantique, la nouvelle génération du web actuel, avec une nouvelle infrastructure qui a pour ambition d'ajouter de la sémantique aux données pour permettre de les traiter automatiquement par machine. Afin de comprendre la signification de l'information sur le Web. Et par la suite nous avons présenté des principes de l'architecture du Web sémantique, des technologies et langages comme le RDF, RDFS, OWL et le langage d'interrogation SPARQL,

Nous avons également abordé un autre élément central du Web sémantique : les ontologies qui jouent un rôle essentiel dans la représentation des connaissances et la structuration des données.

Dans le prochain chapitre, notre attention sera portée sur les travaux relatifs à l'enrichissement des ontologies. Nous explorerons ces travaux afin de découvrir les diverses approches et méthodes utilisées pour améliorer la qualité et l'étendue des ontologies. Cette exploration nous permettra d'approfondir notre compréhension et de mieux cerner notre approche, qui constitue l'objectif principal de ce mémoire.

# **Chapitre 2**

Etat de l'art sur les  
Méthodes d'enrichissement

## **1. Introduction**

L'apprentissage d'ontologie consiste à acquérir de nouvelles connaissances dans une ontologie de manière semi-automatique ou automatique. Il englobe plusieurs sous-tâches, dont l'extension d'une ontologie existante, qui peut être réalisée de deux manières : le peuplement et l'enrichissement de l'ontologie. Ces deux approches se basent généralement sur l'extraction d'informations à partir d'un contenu spécifique à un domaine.

Le peuplement d'ontologie implique l'insertion d'instances de concepts et/ou de propriétés dans une ontologie existante. Son objectif est d'enrichir l'ontologie en ajoutant des exemples concrets liés aux concepts et aux propriétés existants. En revanche, l'enrichissement d'ontologie vise à étendre l'ontologie en ajoutant de nouveaux concepts, propriétés et axiomes.

Dans notre travail, nous nous intéressons spécifiquement à l'enrichissement d'une ontologie par l'ajout de nouveaux concepts. Pour cela, nous examinons différents travaux dans ce domaine afin de comprendre les tâches associées à notre approche. Nous avons également entrepris une comparaison des différentes approches.

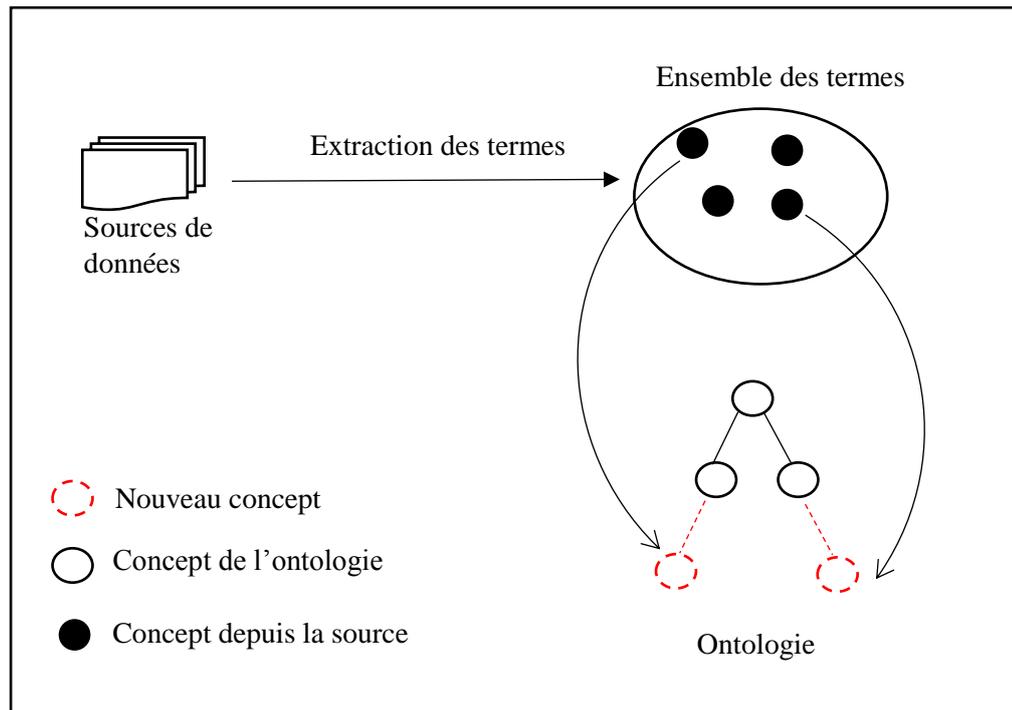
## **2. L'enrichissement d'une ontologie**

L'enrichissement d'une ontologie consiste à ajouter de nouvelles connaissances à une ontologie existante. Cela peut inclure l'ajout de nouvelles classes, de nouvelles relations entre les classes existantes, de nouvelles propriétés ou attributs pour les classes existantes, ainsi que l'ajout de nouvelles instances ou exemples [17].

L'enrichissement d'une ontologie permet d'améliorer la qualité et la pertinence de l'ontologie pour représenter un domaine spécifique. Cela permet également d'élargir la portée de l'ontologie en y ajoutant de nouvelles connaissances, ce qui peut rendre l'ontologie plus utile pour la résolution de problèmes dans le domaine concerné.

L'enrichissement peut être réalisé par des experts du domaine, qui peuvent apporter leur expertise pour ajouter de nouvelles connaissances à l'ontologie existante. Il peut également être réalisé automatiquement en utilisant des techniques d'exploration de données et d'apprentissage automatique pour extraire de nouvelles connaissances à partir de sources de données diverses.

Le processus d'enrichissement d'ontologie peut être divisé en deux sous tâches : la recherche de nouveaux concepts et relations, et le placement de ces concepts et relations au sein de l'ontologie (figure 2.1). Plusieurs travaux se sont intéressés à ce processus d'enrichissement d'ontologies, abordant une ou plusieurs de ses étapes : (i) extraction de termes représentatifs dans un domaine spécialisé, (ii) identification de relations entre les termes, (iii) placement des nouveaux termes dans une ontologie existante [18].



**Figure 2.1 :** Le processus général d'enrichissement d'ontologie. [18]

### 3. Critères de classification des méthodes d'enrichissement d'ontologies

Pour pouvoir enrichir une ontologie, il existe plusieurs critères de classement des méthodes d'enrichissement des ontologies, tels que :

- **Type d'enrichissement :** L'enrichissement d'une ontologie peut être effectué de manière manuelle, semi-automatique ou automatique. L'enrichissement automatique utilise des algorithmes pour extraire des informations à partir de sources de données, tandis que l'enrichissement semi-automatique combine l'intervention humaine avec des techniques informatiques. L'enrichissement manuel implique une intervention humaine directe pour modifier l'ontologie.

- **Termes ajoutés :** les méthodes d'enrichissement des ontologies peuvent également être classées en fonction des termes ajoutés. Certaines méthodes enrichissent tous les composants de l'ontologie, tandis que d'autres se concentrent uniquement sur l'ajout de concepts ou de relations.
- **Sources d'enrichissement des ontologies :** les sources d'enrichissement sont des ressources externes utilisées pour ajouter de nouvelles connaissances à une ontologie existante. Ces sources peuvent être diverses, telles que des documents textuels, des bases de données, des ontologies existantes, des experts du domaine et le web sémantique.

#### **4. Travaux d'enrichissement d'une ontologie à partir de données liées**

L'enrichissement d'une ontologie par Linked Data consiste à utiliser des données liées (Linked Data) pour étendre ou améliorer l'ontologie existante. Il peut être réalisé en exploitant les informations contenues dans des sources de données externes liées aux concepts de l'ontologie.

Dans le cadre de notre recherche sur l'enrichissement d'ontologies à partir de données liées, nous avons examiné plusieurs travaux existants. Ces travaux constituent une ressource précieuse pour notre projet, car ils nous offrent un point de départ solide. Nous avons résumé les principales contributions de ces travaux afin de mieux les comprendre et de les intégrer dans notre propre approche d'enrichissement d'ontologie.

- **EOLD - 2013**

EOLD (Enrichment Ontology via linked data) est un système semi-automatique d'enrichissement d'ontologies à partir de DBpedia. Il repose sur le processus d'exécution d'une requête sur les données liées qui peut comporter deux phases distinctes : lors de la première phase, le système d'exécution de requêtes sélectionne des URI et les utilise pour récupérer des données à partir du Web interrogé ; lors d'une phase ultérieure, le système génère le résultat de la requête en utilisant les données récupérées dans la première phase. Le composant d'enrichissement d'ontologie traite les nouveaux termes qui peuvent être ajoutés en tant que classe, ObjectProperty, dataProperty à l'ontologie [19]. Le système EOLD propose plusieurs fonctionnalités à savoir l'affichage détaillé de l'ontologie à enrichir et enrichie et également la comparaison entre les deux. EOLD permet aussi d'ajouter et de supprimer des classes, des propriétés, des individus, et également afficher graphiquement l'ontologie.

- **PROPheT-2018**

PROPheT est un système de population d'ontologies et l'enrichissement sémantique qui peut récupérer des instanciations de concepts à partir de sources de données liées. Les instances récupérées sont filtrées par l'utilisateur et sont ensuite insérées, avec leurs propriétés dans une ontologie cible. PROPheT fournit divers modes de récupération d'instance et permet d'établir des mappages définis par l'utilisateur [20].

PROPheT enrichi sémantiquement des ontologies en établissant des liens entre les propriétés dans les ontologies locales et distantes via des déclarations `owl:equivalentProperty` ajoutées dans le modèle local. Les liens similaires entre les classes sont représentés via les déclarations `owl:sameAs` et `rdfs:seeAlso` ajoutées à l'ontologie locale.

PROPheT offre également la possibilité d'enrichir sémantiquement des instances déjà existantes dans l'ontologie locale avec des propriétés et des valeurs d'instances similaires dans des ontologies distantes. Les instances similaires peuvent appartenir à une ou plusieurs classes différentes dans l'ontologie distante, ainsi, l'outil présente à l'utilisateur le `rdf:type` de chaque instance. Sur la base du contenu et de la sémantique des instances dérivées, l'utilisateur peut décider quelles paires propriété-valeur il/elle insérera dans l'ontologie locale.

Le système utilise Python pour le front-end et le back end avec des API spéciales. Une base de données SQLite a également été configurée dans le back-end pour stocker les données dynamiques qui sont créées pendant le fonctionnement de cet outil.

- **SLDM-2017**

Dans ce projet de doctorat, trois méthodes sont proposées, qui permettent d'ajouter de nouveaux axiomes à une ontologie. Le développeur suppose que l'ontologie contient du vocabulaire et qu'il existe un ensemble de données liées utilisant le vocabulaire. Le but est de générer des axiomes candidats qui peuvent être revu et ajouté à l'ontologie. Les trois méthodes proposées utilisent uniquement langage de requête SPARQL pour accéder à l'ensemble de données [21].

La première méthode génère un grand ensemble d'axiomes candidats et utilise la programmation linéaire entière pour sélectionner le sous-ensemble le plus intéressant des axiomes, qui est ensuite présenté au utilisateur. Il convient pour diviser une classe existante en

nouvelles classes, chacune avec une axiomatique définition. L'implémentation de la méthode proposée est réalisée dans le cadre d'un plugin RMonto.

La deuxième méthode est basée sur un algorithme d'exploration d'attributs de Formal Concept Analyse (Formal Concept Analysis (FCA)). L'algorithme proposé introduit un classificateur qui apprend des décisions prises par l'utilisateur et une fois formé, il essaie de remplacer l'utilisateur. L'implémentation de l'algorithme est faite par le langage Java, l'application résultat est basée sur la bibliothèque d'apprentissage automatique Weka.

La troisième méthode, appelée SLDM (Swift Linked Data Miner) qui repose sur un algorithme développé spécifiquement pour extraire directement des expressions de classes OWL à partir de données liées, en n'utilisant un point d'accès SPARQL comme moyen d'accéder aux données. L'idée générale de l'algorithme est de télécharger une partie pertinente du graphe RDF dans un point d'accès SPARQL, de le traiter rapidement et de passer au téléchargement de la partie suivante.

- **SAUPODOC-2016**

Mme Céline Alec propose une approche nommée SAUPODOC s'appuie sur une ontologie, et combine des étapes de peuplement et d'enrichissement. SAUPODOC détaille une application spécifique à l'annotation automatique de documents, puis un mécanisme de génération de requêtes SPARQL entièrement automatique, qui construit les assertions de propriété à insérer dans l'ontologie en utilisant les informations provenant du LOD. L'approche est capable de générer automatiquement des définitions de concepts qui sont à la fois compréhensibles par des humains et interprétables par des machines [22].

Les résultats de tests de cette approche montrent clairement le bénéfice par rapport à des classifieurs connus et la pertinence de l'approche à base d'ontologie reposant sur une combinaison particulière de plusieurs techniques pour étiqueter les documents.

## 5. Comparaison de méthodes d'enrichissement

	Source de données	Mesures de Similarité	Technique	Type	Les composants Enrichissement (Concepts-relation Propriétés-instance-axiome)
EOLD -2013	Web	-	-	Semi-automatique	Concepts Propriétés Instances
PROPhET -2018	Web	Sémantique		Semi-automatique	Propriétés-relation instances
SLDM -Rmonto-2017	Web	-	Algorithme approprié	Semi-automatique	Tous
SLDM -FCA -2017	Web	-	FCA	Automatique	Tous
SLDM-3-2017-	Web	-	Algorithme approprié	Automatique	Tous
SAUPODOC -2016	Web + Texte	Sémantique	Algorithme approprié	Semi-automatique	Tous

**Tableau 2.1 :** Comparaison de méthodes d'enrichissement.

En examinant le tableau précédent (Tableau 2.1), on peut observer une comparaison entre les différentes méthodes étudiées. Ces méthodes partagent certaines techniques communes, mais diffèrent également sur d'autres aspects :

- La plupart des méthodes qui extraient les données à partir de données liées utilisent le langage SPARQL, car c'est le seul langage qui permet d'interagir avec ce type de source.
- Les méthodes diffèrent également dans le type de mesure de similarité utilisée pour la comparaison des termes. Certaines méthodes se basent sur des mesures sémantiques, tandis que d'autres utilisent leurs propres algorithmes.
- Certaines méthodes permettent l'enrichissement de tous les composants des ontologies, tandis que d'autres se concentrent uniquement sur une partie spécifique (concept, relation, instances).

## **6. Conclusion**

Après avoir examiné les méthodes d'enrichissement d'ontologie à partir de différentes sources, nous avons pu identifier les étapes clés de l'enrichissement d'ontologie. Nous avons également identifié les problématiques courantes rencontrées lors de ce processus et examiné les différentes solutions proposées. Ce chapitre nous a permis de définir les grandes lignes de notre approche proposée. En nous basant sur cette étude, nous avons pu déterminer les éléments essentiels de notre méthode, qui seront détaillés dans le chapitre suivant.

# **Chapitre 3**

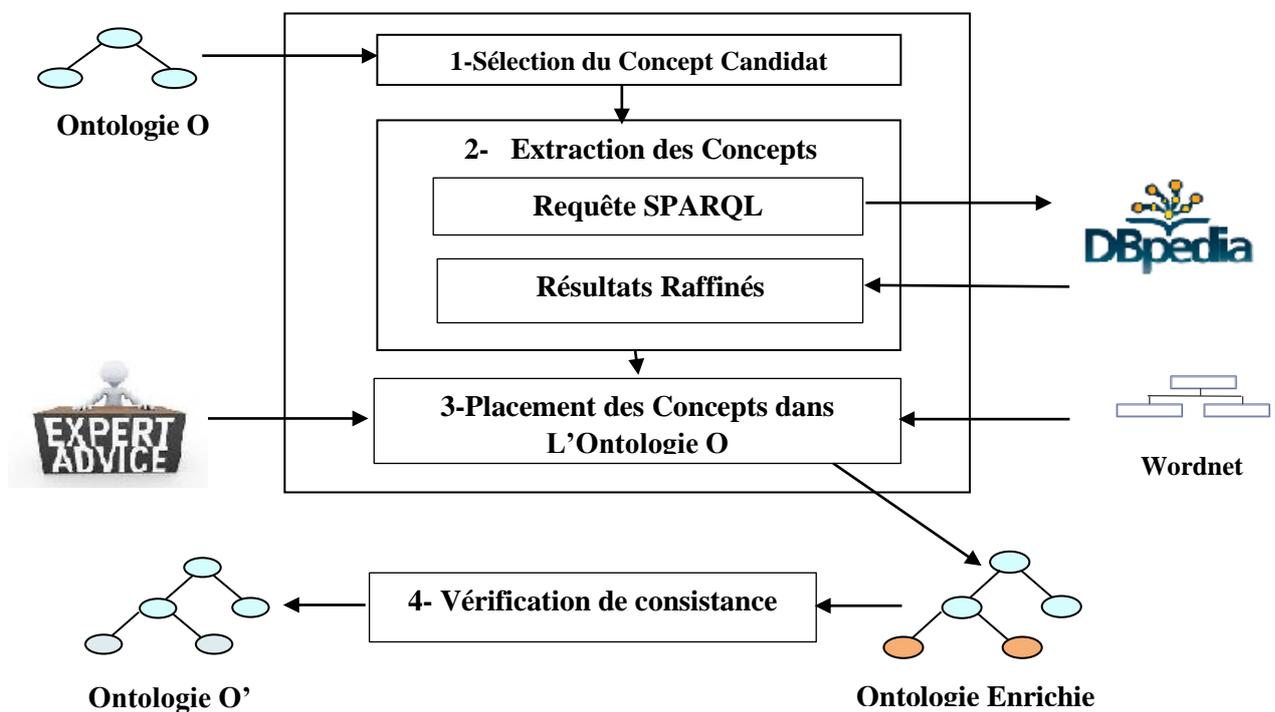
Conception du système  
d'enrichissement d'Ontologies

## 1. Introduction

Après avoir introduire le contexte général de notre étude et les principales approches d'enrichissement d'ontologies depuis les données liées, viens l'étape de présentation de l'approche proposée qui fait l'objet du présent chapitre.

## 2. L'approche proposée pour l'enrichissement d'Ontologies

Notre objectif consiste à développer un système d'enrichissement d'ontologie par des nouveaux concepts extraite à partir du DBpedia. L'interrogation de cette dernière se base sur des requêtes SPARQL générer à partir des concepts de l'ontologie initiale. Notre système se compose de quatre modules différents illustré dans la figure 3.1, nous détaillons chaque module avec une brève description.



**Figure 3.1 :** Processus général de système d'enrichissement d'ontologies

## 2.1. Sélection des concepts candidats

La première étape dans notre système d'enrichissement d'ontologies c'est la sélection des concept candidat. Notre système permet à l'expert de choisir le concept à partir duquel nous lançons une série d'étapes visant à enrichir l'ontologie. Selon le concept sélectionné, nous ajoutons de nouvelles classes équivalentes au concept choisi ou les intégrons en tant que sous-classes du concept choisi. Cette approche offre à l'expert la possibilité de prendre des décisions clés dans le processus d'enrichissement de l'ontologie, permettant ainsi une personnalisation et une adaptation précises aux besoins spécifiques du domaine.

## 2.2. Extraction des concepts depuis LOD

L'extraction des concepts à partir du LOD Cloud revêt une importance cruciale dans notre approche. Dans ce contexte, nous avons décidé d'utiliser DBpedia comme source de données pour extraire les concepts pertinents. En utilisant des requêtes spécialisées, nous collectons les concepts significatifs à partir de DBpedia. Nous nous appuyons sur les relations et les liens sémantiques entre les concepts candidats pour sélectionner des nouveaux concepts qui seront par la suite jugés pertinents ou non pour notre ontologie.

- **DBpedia<sup>11</sup>**

DBpedia est une ressource de connaissances libre et collaborative, construite en extrayant et en structurant les informations de Wikipédia de manière sémantique. Elle est alimentée par un processus d'extraction automatique des données, ce qui permet de créer une base de connaissances interconnectée et exploitant le Web des données, et dont ce qui suit quelques points importants sur DBpedia [23] :

---

<sup>11</sup> <https://www.dbpedia.org/about/>



**Figure 3.2** : Logo de DBpedia

- **Données structurées** : DBpedia extrait des informations structurées à partir de Wikipédia, ces informations sont représentées en utilisant le langage RDF (Resource Description Framework) et sont organisées selon des ontologies prédéfinies, parmi les ontologies les plus couramment utilisées dans DBpedia, on trouve DBpedia Ontology, YAGO, Schema.org, etc.
- **Interconnexion des données** : DBpedia relie les entités et les concepts à travers des liens sémantiques. Ces liens permettent de naviguer entre les différentes entités et de découvrir des relations et des informations connexes. Par exemple, il est possible de suivre les liens pour passer d'une ville à son pays, puis à d'autres villes du même pays.
- **Multilinguisme** : DBpedia extrait des informations de Wikipédia dans de nombreuses langues, ce qui permet d'accéder à des connaissances multilingues. Cela facilite la création de ressources et d'applications dans différentes langues, et favorise la collaboration internationale autour des données liées.
- **Évolution continue** : DBpedia est constamment mis à jour pour refléter les mises à jour de Wikipédia. Des extraits réguliers sont effectués pour incorporer les nouvelles informations et améliorer la couverture de la base de connaissances.

- **Interroger les données DBpedia**

Pour interroger les données de DBpedia, il est courant d'utiliser les préfixes et le langage de requête SPARQL. Les préfixes sont utilisés pour abrégés les URI et rendre les requêtes plus lisibles et concises.

Dans notre travail, nous avons préconfiguré certains préfixes couramment utilisés dans les requêtes SPARQL pour interroger DBpedia. Ces préfixes sont directement intégrés dans notre éditeur de requête SPARQL, ce qui facilite la rédaction des requêtes pour les utilisateurs. Toutefois, nous avons également pris en compte la possibilité pour les utilisateurs de modifier ces préfixes selon leurs besoins spécifiques, le tableau suivant présente la signification de quelque préfix.

Préfixe	Signification
rdf: <a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#">http://www.w3.org/1999/02/22-rdf-syntax-ns#</a>	définit des termes et des concepts dans le format RDF/RDFS
rdfs: <a href="http://www.w3.org/2000/01/rdf-schema#">http://www.w3.org/2000/01/rdf-schema#</a>	
owl: <a href="http://www.w3.org/2002/07/owl#">http://www.w3.org/2002/07/owl#</a>	identifier les termes et les concepts spécifiques liés à OWL
dbo: <a href="http://dbpedia.org/ontology/">http://dbpedia.org/ontology/</a>	"dbo" identifier les termes et les concepts spécifiques liés à DBpedia Ontology
dbr: <a href="http://dbpedia.org/resource/">http://dbpedia.org/resource/</a>	"dbr" identifier les ressources spécifiques dans DBpedia
dbp: <a href="http://dbpedia.org/property/">http://dbpedia.org/property/</a>	"dbp" est utilisé pour identifier les propriétés spécifiques dans DBpedia

**Tableau 3.1** : Signification de quelques préfixes

Une fois que les préfixes sont définis, l'utilisateur rédige le corps de la requête SPARQL pour interroger DBpedia via son Endpoint <sup>12</sup>. Cependant, l'utilisateur sera assisté en lui proposant des types de relations qu'il peut utiliser pour rechercher des ressources sur DBpedia. Parmi les relations couramment utilisées, on retrouve les relations "rdfs:subClassOf" et "rdf:equivalent". En plus de ces relations standard, l'utilisateur peut introduire d'autres relations spécifiques à son domaine d'intérêt dans sa requête SPARQL. L'exemple suivant présente une requête SPARQL :

<sup>12</sup> <https://dbpedia.org/sparql>

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT DISTINCT ?concept WHERE
{
    ?concept rdfs:subClassOf dbo:Land_area.
}
```

Cette requête SPARQL interroge la base de connaissances DBpedia et récupère tous les concepts qui sont des sous-classes de `dbo:Land_area` :

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbo: <http://dbpedia.org/ontology/>
```

Ces deux lignes définit les préfixes "rdfs" et "dbo" pour les espaces de noms utilisés dans la requête.

```
SELECT DISTINCT ?concept WHERE
```

Cette ligne indique que nous voulons sélectionner distinctement les concepts qui satisfont la condition de la requête. La partie WHERE spécifie les critères de sélection.

```
?concept rdfs:subClassOf dbo:Land_area.
```

Cette ligne signifie que nous recherchons des triplets où le sujet (?concept) est une sous-classe de `dbo:Land_area`.

Après avoir formulé la requête, elle est exécutée pour extraire les données de DBpedia. Une fois les résultats récupérés, une étape de filtrage est effectuée pour éliminer les lignes vides et supprimer les doublons, même avec une différence de casse. De plus, les concepts qui correspondent exactement au concept candidat ou à une classe déjà présente dans l'ontologie sont également éliminés. Cette étape de filtrage permet de nettoyer les résultats et de garantir l'intégrité et la pertinence des concepts à ajouter ou à intégrer dans l'ontologie.

### **2.3. Placement des Concepts dans L'Ontologie**

Une fois les résultats de la requête extraits de DBpedia et après un premier processus de raffinement, nous avons choisi d'effectuer un deuxième raffinement sémantique afin d'assurer la cohérence des résultats de nos requêtes SPARQL. Ce deuxième raffinement repose sur l'utilisation de mesures de similarité sémantique qui évaluent le degré de similarité sémantique entre les résultats obtenus.

### 2.3.1 Raffinage par les mesures de similarité

Parmi les différentes mesures de similarité disponibles, nous souhaitons mettre en évidence celles que nous avons spécifiquement utilisées dans notre travail.

- **La similarité sémantique basée sur WordNet**

Nous sommes particulièrement intéressés à la similarité sémantique basée sur WordNet. WordNet est une ressource lexicale qui organise les mots en ensembles de synonymes appelés synsets et fournit des relations sémantiques entre ces synsets.

En utilisant WordNet, nous avons pu calculer la similarité sémantique entre les concepts en nous appuyant sur les relations lexicales et les informations de hiérarchie présentes dans cette ressource. Cela nous a permis de quantifier la proximité sémantique entre les concepts et d'évaluer leur degré de similarité.

- **La mesure de similarité Wu & Palmer**

La mesure de similarité de Wu & Palmer est une méthode couramment utilisée pour calculer la similarité sémantique entre les synsets dans WordNet. L'idée principale est que des synsets partageant des ancêtres communs plus proches ont une similarité plus élevée. La mesure entre  $C_1$  et  $C_2$  est calculée en utilisant la formule suivante [24] :

$$ComSim(C_1, C_2) = \frac{2 * depth(C)}{depth_C(C_1) + depth_C(C_2)}$$

Où  $depth(c)$  est la profondeur d'un synset qui est le nombre d'arêtes dans le chemin le plus court entre ce synset et la racine de la hiérarchie de WordNet, et  $depth_C(C_i)$  avec  $i$  le nombre d'arcs qui séparent  $C_i$  de la racine en passant par  $C$ .

La mesure de similarité de Wu & Palmer est souvent utilisée avec les relations "is-a" (est-un) de WordNet. Ces relations "is-a" indiquent une relation de généralisation, où un concept plus spécifique est une sous-classe d'un concept plus général. Par exemple, si nous avons deux

synsets, A et B, et que B est une sous-classe de A (B "is-a" A), alors on peut considérer que B est plus spécifique que A. En utilisant la mesure de Wu & Palmer, on peut calculer la similarité sémantique entre A et B en utilisant la profondeur de leur ancêtre commun le plus proche.

Dans notre approche, nous avons utilisé ces mesures en exploitant les relations de type RDF telle que `rdf:subClassOf` qui peut être considérées comme équivalentes à la relation "is-a" de WordNet.

La mesure de similarité de Wu & Palmer fournit une valeur comprise entre 0 et 1, où 0 représente une absence de similarité et 1 indique une similarité maximale entre les concepts. Dans notre approche, nous avons fixé un intervalle de sélection des termes candidats entre 0.8 et 1. Les concepts qui ont une similarité supérieure à 0.8 sont considérés comme suffisamment similaires pour être ajoutés à l'ontologie.

Il est important de noter que le choix de l'intervalle de similarité et les concepts à ajouter à l'ontologie dépendent des objectifs spécifiques de l'utilisateur de système. Le seuil de 0.8 que nous avons utilisé est une valeur peut être ajusté en fonction des besoins et des exigences de l'utilisateur de système.

Dans l'étape suivante, nous procédons à l'incorporation de ces concepts dans l'ontologie.

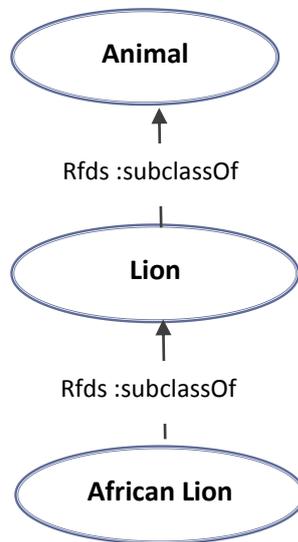
### **2.3.2 Placement des nouveaux concepts dans l'ontologie**

Lors de l'étape d'extraction des concepts à partir de DBpedia, nous obtenons un ensemble de concepts candidats. Cependant, l'utilisation des mesures de similarité de Wu & Palmer est avec seulement quelques relations RDF, ce qui peut être une limitation. De plus, DBpedia est une source de données basée sur Wikipédia, qui est une ressource ouverte pouvant présenter des incohérences de données.

Pour faire face à ces problèmes, notre approche d'enrichissement est semi-automatique, cela signifie que les résultats obtenus lors de l'étape précédente doivent être validés par un expert du domaine. L'intervention d'un expert est primordiale pour garantir la qualité et la cohérence des concepts obtenus.

L'expert du domaine aura la possibilité de supprimer les concepts sélectionnés s'il juge qu'ils ne sont pas pertinents ou de refaire la phase d'extraction si nécessaire. L'expert apportera son expertise pour évaluer les concepts candidats en fonction de leur adéquation au domaine d'application spécifique.

Une fois que les concepts sélectionnés ont été validés, nous les placerons dans l'ontologie en utilisant les relations appropriées tel que `rdf:equivalent` ou `rdfs:subclassOf`, la figure 3.3 illustre un exemple de classe et sous classe.



**Figure 3.3 :** Exemple de classe et subclassOf. <sup>13</sup>

#### 2.4. Vérification de consistance de l'ontologie enrichie

Après l'ajout des nouveaux concepts dans l'ontologie, il est essentiel de vérifier son intégrité pour s'assurer de sa cohérence syntaxique et sémantique. Notre objectif est de rechercher les moyens nécessaires pour atteindre cet objectif.

Pour garantir la syntaxe correcte de l'ontologie, nous pouvons effectuer une vérification syntaxique en utilisant des outils et des vérificateurs OWL (Web Ontology Language). Ces outils analysent la structure de l'ontologie pour détecter d'éventuelles erreurs ou violations des règles syntaxiques de OWL. Ils peuvent signaler des problèmes tels que des incohérences de déclarations, des références incorrectes ou des conflits dans les axiomes définis.

Quant à la cohérence sémantique, il est nécessaire de vérifier que les nouveaux concepts ajoutés ne contredisent pas les axiomes et les règles déjà définis dans l'ontologie. Cela peut être

---

<sup>13</sup> <https://www.w3.org/TR/2005/NOTE-swbp-classes-as-values-20050405/>

réalisé en utilisant des raisonneurs logiques qui appliquent des inférences pour déterminer la cohérence de l'ontologie. Les raisonneurs peuvent détecter des conflits, des incohérences ou des contradictions dans les déclarations, et signaler les erreurs potentielles.

En utilisant ces approches combinées de vérification syntaxique et sémantique, nous pouvons nous assurer que l'ontologie enrichie est à la fois syntaxiquement correcte et sémantiquement cohérente après l'ajout des nouveaux concepts. Cela garantit que l'ontologie peut être utilisée de manière fiable pour l'annotation, la recherche et d'autres tâches liées à la représentation et à l'interprétation des connaissances.

### **3. Conclusion**

Dans ce chapitre, nous avons présenté notre approche semi-automatique d'enrichissement de l'ontologie. Nous avons décrit les différentes étapes de notre approche, en commençant par l'extraction des données depuis DBpedia, puis l'utilisation des mesures de similarité pour sélectionner les termes candidats. Ensuite, nous avons expliqué comment placer ces concepts dans l'ontologie en utilisant les relations appropriées. Cette approche nous permet d'enrichir l'ontologie en intégrant des concepts pertinents et similaires.

Dans le chapitre suivant, nous détaillerons les outils que nous avons utilisés pour mettre en application cette approche.

# **Chapitre 4 :**

## **Implémentation et validation**

## 1. Introduction

Une fois que nous avons décrit en détail les grandes lignes de notre système d'enrichissement d'ontologie, il est essentiel de trouver les outils appropriés pour sa mise en application et de procéder à des tests afin de vérifier sa validité.

Dans ce chapitre, nous aborderons les outils et langages que nous avons utilisés pour développer notre système. Nous expliquerons en quoi ces outils ont été choisis en fonction de nos besoins spécifiques et comment ils ont contribué à la mise en œuvre réussie de notre système d'enrichissement d'ontologie.

De plus, nous discuterons des différents tests que nous avons effectués pour évaluer l'efficacité de notre système. Ces tests ont été conçus pour mesurer les performances de notre système dans différents scénarios et conditions. Nous décrirons les méthodologies utilisées pour effectuer ces tests, ainsi que les critères d'évaluation que nous avons définis pour juger de la validité et de la performance de notre système.

## 2. Implémentation

Dans la section suivante, nous présenterons les outils que nous avons sélectionnés pour implémenter notre système. Ces outils ont été choisis en fonction de leur compatibilité avec nos besoins spécifiques en matière d'enrichissement d'ontologie.

### 2.1. Les outils de programmation

#### 2.1.1. Langage JAVA<sup>14</sup>

Java est un langage de programmation orienté objet, développé par Sun Microsystems (acquis par Oracle Corporation par la suite), qui permet de créer des applications pouvant s'exécuter sur différentes plateformes. Il a été conçu pour être portable, c'est-à-dire qu'une fois écrit, un programme Java peut être exécuté sur n'importe quel système informatique qui possède une machine virtuelle Java (JVM).

Java est largement utilisé dans le développement d'applications web, de logiciels d'entreprise, de jeux vidéo, d'applications mobiles, d'applications pour objets connectés et d'autres domaines. Il est apprécié pour sa simplicité, sa robustesse, sa sécurité et sa portabilité. Les programmes

---

<sup>14</sup> <https://www.java.com/>

Java sont généralement écrits en utilisant le paradigme de la programmation orientée objet, ce qui facilite la réutilisation du code et la création de logiciels modulaires et évolutifs.

En plus du langage de programmation lui-même, Java dispose d'une vaste bibliothèque standard (Java SE) qui fournit de nombreuses classes et méthodes prêtes à l'emploi pour effectuer des opérations courantes. Il existe également des frameworks et des bibliothèques tierces qui étendent les fonctionnalités de base de Java et facilitent le développement d'applications dans différents domaines.

### **2.1.2. Jena API <sup>15</sup>**

Apache Jena est un framework open source en Java destiné au développement d'applications de traitement de données RDF (Resource Description Framework). Il offre une suite d'outils et de bibliothèques pour la manipulation, le stockage, la requête et le raisonnement sur des données RDF et des ontologies.

Le package *org.apache.jena.ontology* fournit des classes et des interfaces pour travailler avec des ontologies dans le cadre du développement d'applications sémantiques. Ce package offre un ensemble d'outils et de fonctionnalités pour la création, la manipulation et la gestion des ontologies.

### **2.1.3. WordNet**

WordNet de version 4.0 qui est une base de données lexicale et un réseau sémantique de mots anglais. Il est utilisé pour l'analyse des mots et de leurs relations sémantiques. Le choix de WordNet a été cause de diverses raisons :

- C'est la base la plus riche et la plus générale qui contient des termes dans tous les domaines, il est vu comme une ontologie linguistique.
- On peut accéder à WordNet par le biais de son API, WS4J (un Java API pour accéder au dictionnaire apparente de WordNet).
- Nous avons utilisé le WordNet pour le calcul de la mesure de similarité sémantique.

---

<sup>15</sup> <https://jena.apache.org/index.html>

#### **2.1.4. SPARQL**

Nous avons utilisé SPARQL comme langage d'interrogation pour établir une communication avec le moteur de requêtes DBpedia. SPARQL nous a permis d'exécuter des requêtes personnalisées afin d'extraire les informations spécifiques que nous recherchions dans la base de connaissances de DBpedia.

L'utilisation de SPARQL nous a offert une grande flexibilité pour formuler des requêtes adaptées à nos besoins. Grâce à ce langage, nous avons pu récupérer les résultats correspondants aux critères que nous avons définis. Les résultats nous ont été fournis sous forme de triplets RDF, qui sont des structures composées d'un sujet, d'un prédicat et d'un objet, représentant des faits ou des relations entre les entités dans DBpedia.

Une fois les triplets RDF récupérés, nous avons eu la possibilité de les filtrer et de les exploiter selon les besoins spécifiques de notre application, ou selon les préférences et les exigences de l'utilisateur. Cette flexibilité nous a permis de manipuler les données de manière personnalisée, en adaptant les résultats aux besoins spécifiques de notre projet.

#### **2.1.5. Protégé <sup>16</sup>**

Protégé 5.5.0 est un éditeur d'ontologies. Il permet de construire et d'éditer des ontologies sous divers formalismes (RDF, OWL,..). De plus, Protégé est extensible en utilisant des plug-ins et des APIs java pour le développement d'applications particulières. L'interface de Protégé est assez simple, l'ensemble des fonctionnalités de l'éditeur étant regroupé en onglets. Le premier onglet présente les classes de l'ontologie. Le deuxième onglet permet de créer des instances et de leur affecter des propriétés. L'onglet des requêtes permet d'effectuer des requêtes sur l'ontologie en cours d'édition.

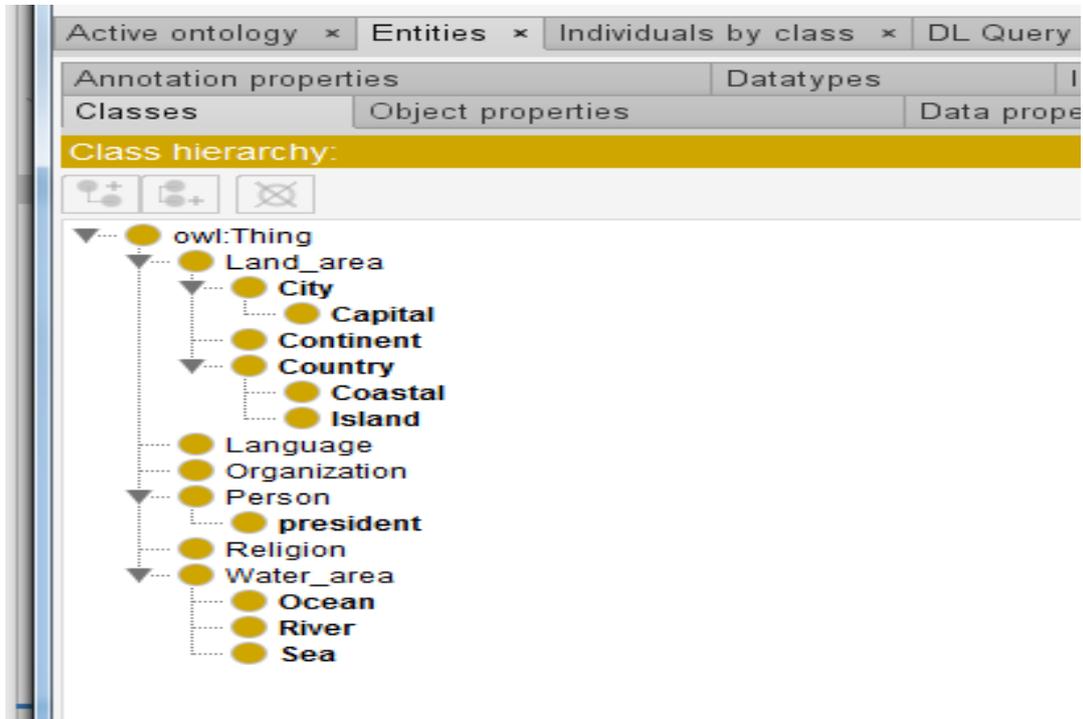
#### **2.1.6. L'ontologies test**

Puisque le but de notre travail est d'enrichir une ontologie, nous avons choisi d'utiliser l'ontologies Géo du domaine géographique qui a pour rôle de prendre en compte et gérer les

---

<sup>16</sup> <https://protege.stanford.edu/about.php>

informations géographiques. Elle est définie par les différentes spécificités auxquelles est liée la multi-dimensionnalité des objets géographiques. Créée par Smaili.A comme projet de fin d'étude en master [25].

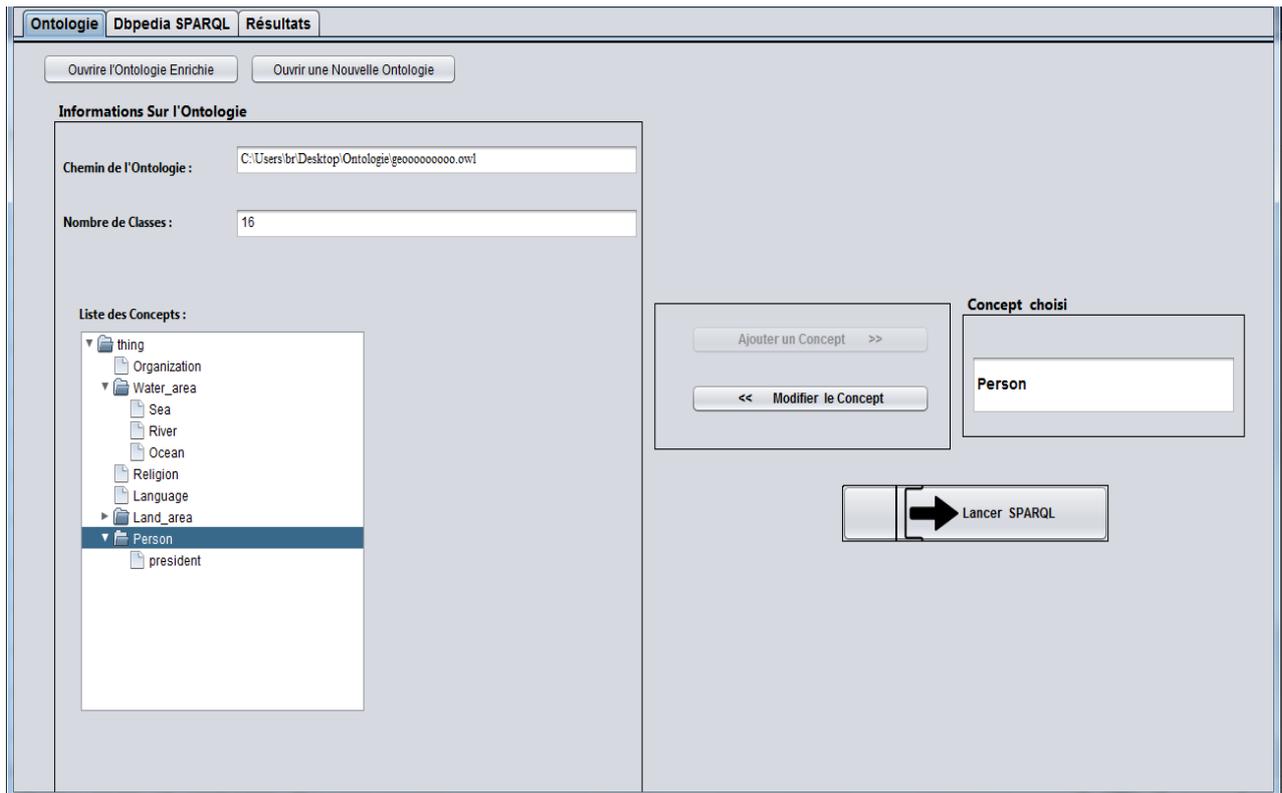


**Figure 4.1:** l'ontologie Géo utilisée pour l'enrichissement

### 3. Présentation de l'application

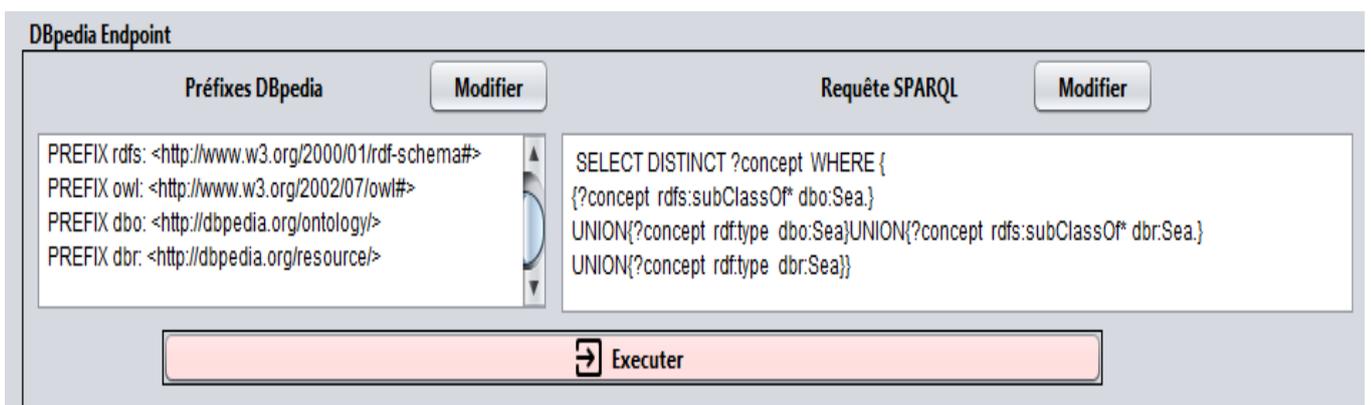
Nous avons développé une interface graphique qui permet de sélectionner l'ontologie à enrichir et de visualiser les différentes étapes du processus.

Dans la première interface, l'utilisateur peut visualiser l'arborescence de l'ontologie sélectionnée et choisir une classe candidate pour l'enrichissement.



**Figure 4.2** : la première fenêtre de l'application

Dans la deuxième fenêtre, nous effectuons une recherche des concepts sur le point d'accès DBpedia Endpoint. L'utilisateur a la possibilité de choisir le type de relation et de modifier la requête SPARQL en fonction de ses besoins. La figure 4.3 présente un exemple de requête SPARQL exécutée:



**Figure 4.3** : Exemple de requête SPARQL exécutée

Une fois la requête exécutée, les résultats obtenus seront soumis à plusieurs types de filtres afin de ne conserver que les résultats pertinents, ce qui est présenté dans la figure 4.4.

The screenshot shows a web interface for executing a SPARQL query against a DBpedia endpoint. The interface is divided into several sections:

- Concept Choisi:** A text input field containing the word "Sea".
- Data Properties:** Radio buttons for "rdf:equivalent", "rdf:subClassOf" (selected), and "Autre".
- DBpedia Endpoint:**
  - Préfixes DBpedia:** A list of prefixes:
 

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbr: <http://dbpedia.org/resource/>
```
  - Requête SPARQL:** A text area containing the query:
 

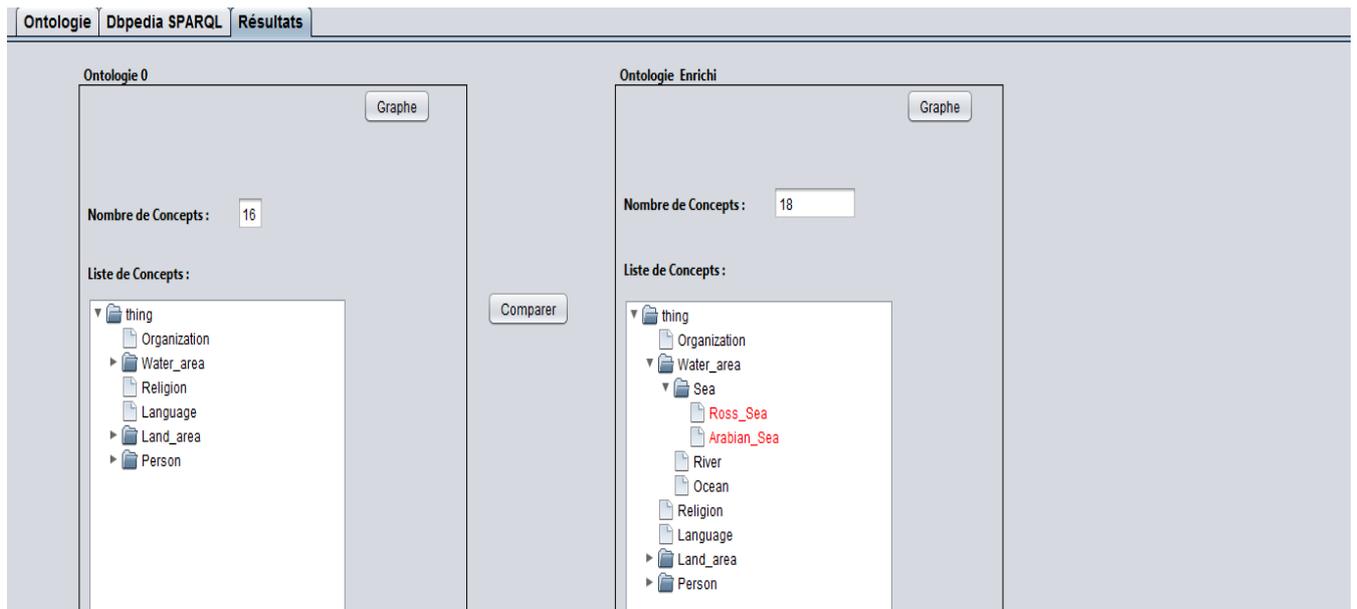
```
SELECT DISTINCT ?concept WHERE {
  {?concept rdfs:subClassOf* dbo:Sea}
  UNION{?concept rdf:type dbo:Sea}UNION{?concept rdfs:subClassOf* dbr:Sea}
  UNION{?concept rdf:type dbr:Sea}}
```
  - Executer:** A large red button to execute the query.
- Résultats:**
  - Nombre de ligne:** 159
  - NameSpace:** http://dbpedia.org/resource/
  - Tableau des Résultats:** A table with columns "Resultat" and "Similarity". The "Arabian\_Sea" row is highlighted.
  - Concepts Ajouter a l'Ontologie:**
    - Propriété:** rdfs:subClassOf\*
    - SubClasses?:** A list containing "Ross\_Sea" and "Arabian\_Sea".
    - SubClasses:** A box containing "Sea".
    - Nouveaux Concepts:** A table with columns "NouveauConcept" and "SubClasse":
 

NouveauConcept	SubClasse
Ross_Sea	Sea
Arabian_Sea	Sea
    - Similarity Table:** A table with columns "Nouveau...", "SubClas...", and "Similarity":
 

Nouveau...	SubClas...	Similarity
Ross_S...	Sea	0.0
Arabian_...	Sea	0.0
    - Ajouter a l'Ontologie:** A pink button to add the selected concepts to the ontology.

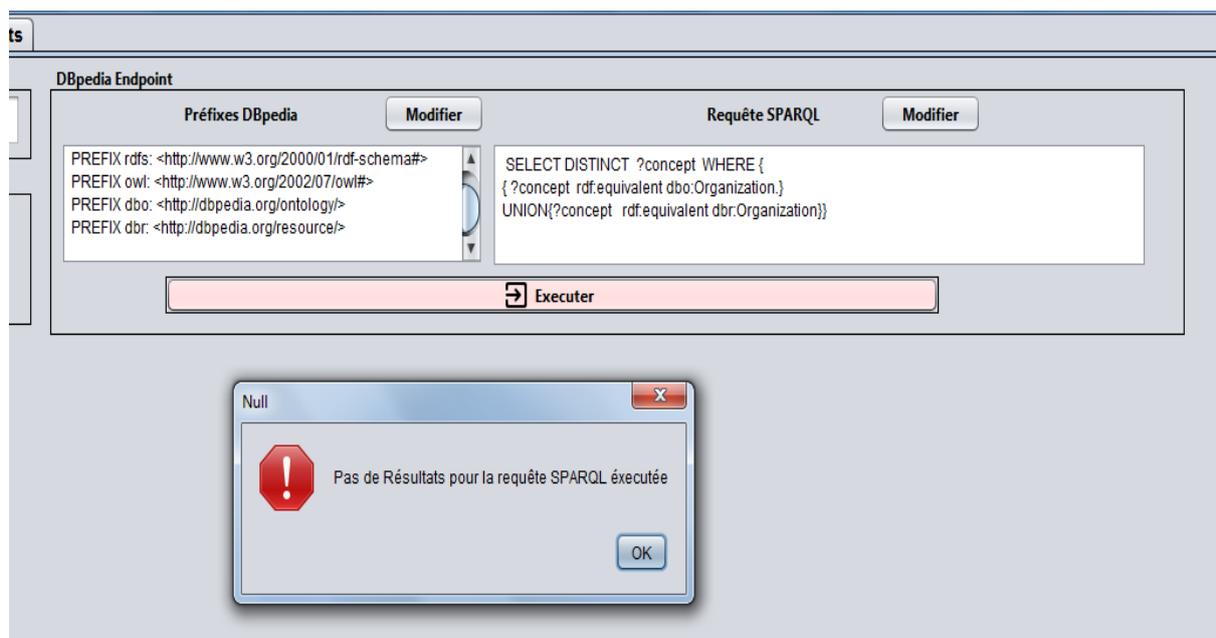
**Figure 4.4 :** les résultats de la requête SPARQL exécutée

Une fois les résultats validés par l'expert, les concepts validés seront ajoutés à l'ontologie. La figure 4.5 présente les nouvelles classes qui sont colorées en rouge.

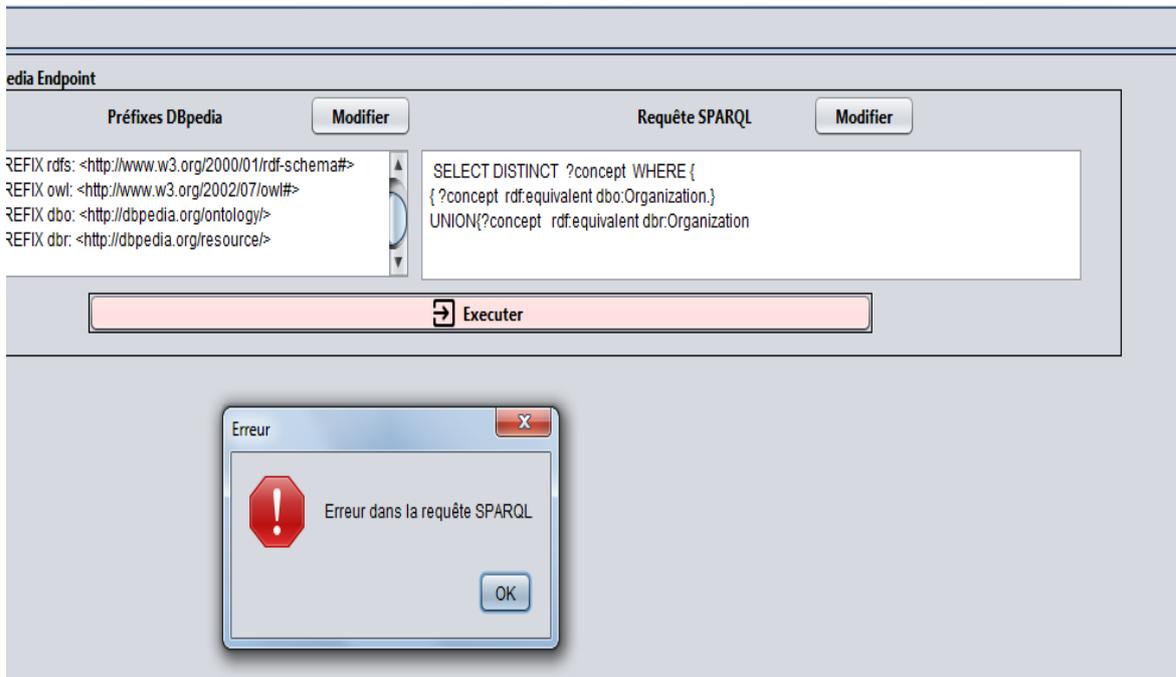


**Figure 4.5** : les nouvelles classes ajouter à l'ontologie

Il peut arriver que l'exécution des requêtes ne donne aucun résultat ou que la requête SPARQL soit incorrecte. Dans de tels cas, un message d'indication sera affiché à l'utilisateur, comme illustré dans la figure 4.6 et la figure 4.7.



**Figure 4.6** : Messages d'indication pour une requête sans résultat

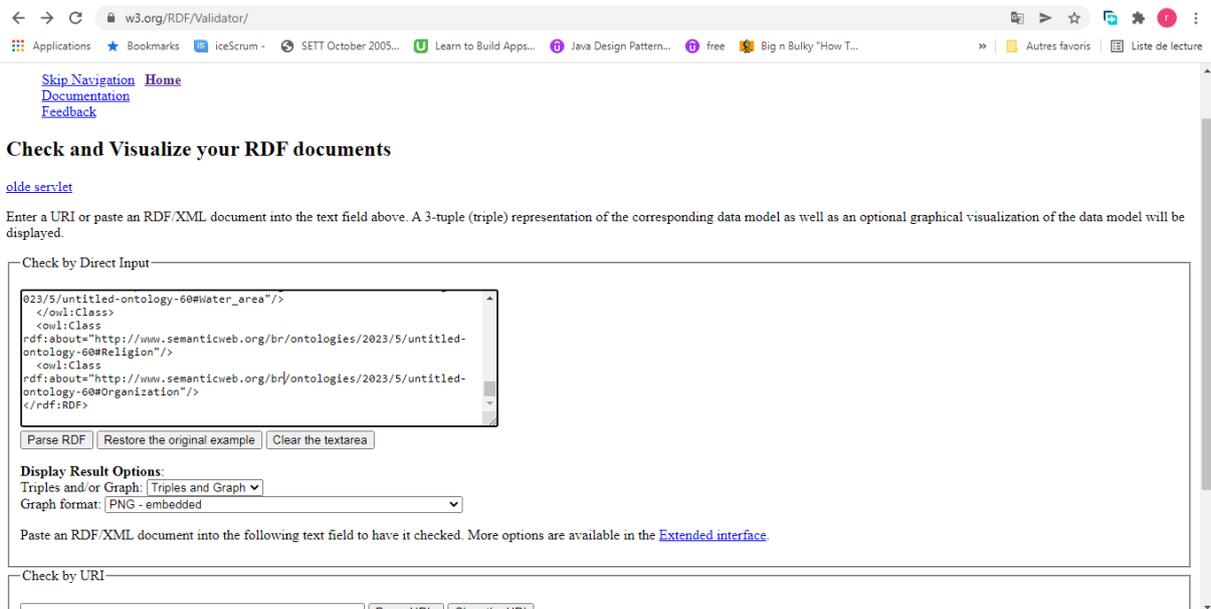


**Figure 4.7 :** Messages d'indication pour une requête invalide

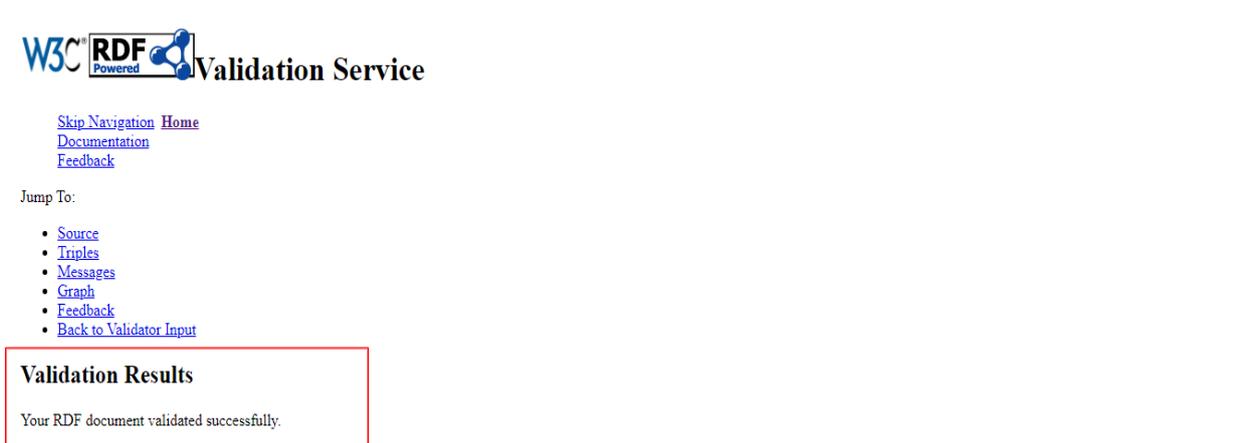
## 4. Validation

Pour évaluer le système que nous avons développé, nous avons procédé à la validation des fichiers résultats OWL contenant la nouvelle ontologie enrichie. À cet effet, nous avons utilisé le validateur de W3C, qui est un validateur de fichiers RDF/XML conforme aux normes du W3C, disponible à l'adresse suivante : <https://www.w3.org/RDF/Validator/>.

La figure 4.8 présente l'interface du site où nous avons téléchargé notre fichier OWL, tandis que la figure 4.9 présente les résultats de la validation.

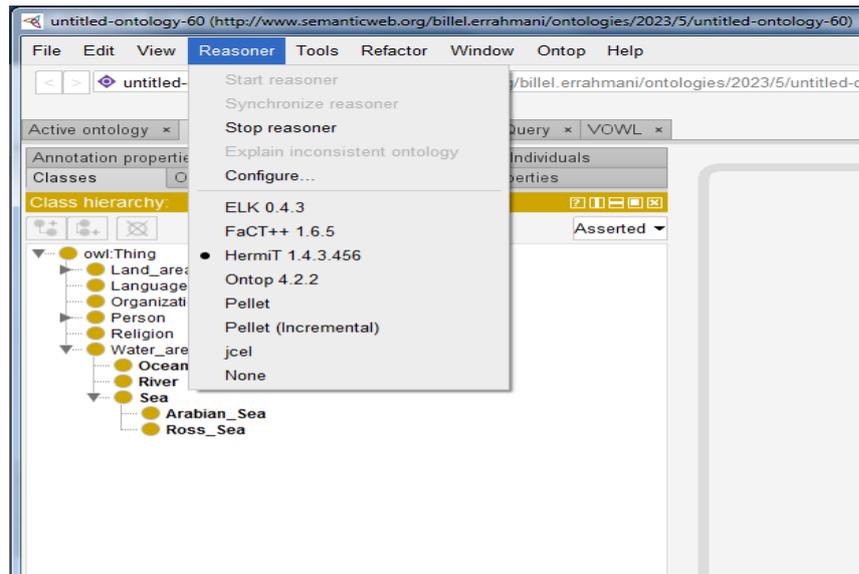


**Figure 4.8 :** L'interface du site validateur de 3WC



**Figure 4.9:** Résultat de la validation

Nous avons également utilisé le logiciel protégé pour ouvrir à nouveau l'ontologie enrichie, et nous avons utilisé les raisonneurs de protégé, notamment Hermit et Elek. Ces raisonneurs sont des outils qui permettent d'effectuer un raisonnement logique sur les ontologies afin de détecter les incohérences et de vérifier la conformité aux règles sémantiques, sur la figure 4.10 présente l'ontologie enrichie sous protégé.



**Figure 4.10:** l'ontologie enrichie sous protégé

## 5. Conclusion

Dans ce chapitre, nous avons présenté les outils utilisés pour le développement de notre système d'enrichissement. Nous avons également exposé les différentes interfaces de notre système et les tests effectués pour valider les ontologies enrichies.

Notre système a été validé syntaxiquement grâce à l'utilisation du validateur de W3C, qui a vérifié la conformité de nos fichiers OWL aux normes du W3C. Cette étape de validation a été essentielle pour garantir la validité syntaxique de nos ontologies enrichies.

En outre, les raisonneurs HermiT et Elek ont été utilisés pour assurer la validité sémantique de nos ontologies enrichies. Ces raisonneurs ont permis de réaliser un raisonnement logique sur les ontologies, détectant les incohérences et vérifiant la conformité aux règles sémantiques établies. Grâce à leur utilisation, nous avons pu garantir la cohérence et la validité sémantique de nos ontologies enrichies.

# **Conclusion et perspectives**

Le travail mené dans ce mémoire nous a permis d'approfondir nos connaissances dans le domaine d'ingénierie des connaissances et plus précisément l'ingénierie des ontologies.

L'enrichissement des ontologies, a pour but d'entendre une ontologie existante par des nouveaux concepts et/ ou relations extraites à partir de différentes sources de données, est une tâche cruciale pour résoudre le problème d'évolution des ontologies.

Nous avons utilisé DBpedia comme source externe d'enrichissement de l'ontologie de départ et à travers SPARQL Endpoint, nous avons généré différentes requêtes en se basant sur les concepts candidats qui sont sélectionnés d'une manière manuelle à partir de l'ontologie initiale.

En exécutant les requêtes SPARQL, des nouveaux concepts formant des triplets avec les concepts candidats sont récupéré, où les prédicats sont de la forme rdfs : subClassOf, rdf: equivalent, une phase de raffinage est nécessaire à fin d'éliminer les concepts redondants et vides.

L'ontologie enrichi, contient toutes les concepts raffinés et validé par l'expert du domaine, a été prouvé cohérente et consistante, ainsi qu'elle a été validée syntaxiquement.

A fin d'améliorer notre approche, nous envisageons les perspectives suivantes :

- Exploiter des liens sémantiques de LOD, ce qui nous permette de naviguer d'une source de données à une autre.
- Entendre l'extraction non seulement sur les concepts mais aussi sur leurs propriétés.
- Automatiser autant que possible les phases d'enrichissement d'ontologie de l'approche proposée.
- Tester la présente approche sur des ontologies de différentes tailles.

# Références

- [ 1 ] Bizer, C. Heath, T. Berners-Lee, T., Linked Data: The Story so Far, International journal on Semantic Web and information systems, 2009.
- [ 2 ] Kadi.H, Construction d'ontologies pour les sources de données liées. Mémoire de Master. Université Dr Tahar Moulay, Saida-2014.
- [ 3 ] Djedidi.R, M.A.Aufaure. Enrichissement d'ontologies : maintenance de la consistance et évaluation de la qualité. 19es Journées Francophones d'Ingénierie des Connaissances (IC 2008), Nancy, France. Pages 249-260. hal-00416702, Jun 2008.
- [ 4 ] T.Berners-Lee, J.Hendler, O.Lassila. The Semantic Web. Scientific American, vol. 284, no. 5, pages 34–43, 2001.
- [ 5 ] H.Eero, Semantic Web Kick-Off in Finland, Vision, Technologies, Research and Applications, HIT Publications, 2002.
- [ 6 ] A.Marchetti, M.Tesconi, F.Ronzano, S.Minutoli, Formalizing Knowledge by Ontologies: OWL and KIF, IIT Department Italy,2008.
- [ 7 ] B.DuCharme, Learning SPARQL: Querying And Updating With SPARQL 1.1., O'Reilly Media , 2013.
- [ 8 ] J.Rapoza, SPARQL Will Make the Web Shine , eWeek, 2006 .
- [ 9 ] Gruber T.R, A translation approach to portable ontologies. Knowledge Acquisition, pages 199-220, 1993.
- [ 10 ] Swartout.B , Patil.R , Knight.K , Russ.T: Towards Distributed Use of Large-Scale Ontologies. Spring Symposium Series on Ontological Engineering, Stanford University, CA, pages 138-148,1997.
- [ 11 ] Studer .R., Benjamins .V.R., Fensel .D, Knowledge Engineering: Principles and Methods. Data & Knowledge Engineering, pages 161-197, 1998.
- [ 12 ] Gruber T. R, Toward Principles for the Design of Ontologies Used for Knowledge Sharing, presented at International Workshop on Formal Ontology, Padova, Italy, 1993.
- [ 13 ] M.Tirit, un système de recherche d'information basé sur les ontologies, Mémoire de master, Université Mohamed Kheider Biskra,2019.
- [ 14 ] Guarino.N, Formal ontology and information systems, In Proceedings of the International Conference of Formal Ontology in Information Systems, Trento, Italy ,pages 3-15, 1998.
- [ 15 ] T.Heath , C.Bizer. Linked data: Evolving the web into a global data space. Synthesis lectures on the semantic web : theory and technology, vol. 1, no. 1, pages 1-136, 2011.
- [ 16 ] T.Berners-Lee. Linked data. consulté depuis from <https://www.w3.org/wiki/LinkedData> le 09/06/2023.

- [ 17 ] Petasis, G., Karkaletsis, V., Paliouras, G., Krithara, A. et Zavitsanos, E. *Ontology Population and Enrichment: State of the Art*. pages 134-166. 2011.
- [ 18 ] L.D.Jorio, L.Abrouk, C.Fiot, D.Héri.M. Teisseire, *Enrichissement d'ontologie basé sur les motifs Séquentiels*, article de LIRMM (CNRS - UMII), Montpellier,2007.
- [ 19 ] Lachachi.Saddam.H , *Enrichment ontology via linked data* , Mémoire de Master, Université Dr Tahar Moulay ,Saida,2013.
- [ 20 ] Marina.R, Panagiotis.M, Efstratios.K, Ioannis.K ,*Ontology Population and Semantic Enrichment from Linked Data Sources*, publier dans *Communications in Computer and Information Science* , juin 2018.
- [ 21 ] Mgr inz.J.P, *Methods For Automatic Enrichment Of Ontologies From Linked Data*, thèse de doctorat, Université de technologie de Poznan -Pologné ,2017.
- [ 22 ] Céline.A, *Enrichissement et peuplement d'ontologie à partir de textes et de données du LOD : Application à l'annotation automatique de documents*, thèse de doctorat, université Paris. 2016.
- [ 23 ] C. Bizer et al, « *DBpedia - A Crystallization Point for the Web of Data* », *Web Semant*, vol. 7, no 3, pages 154-165, 2009.
- [ 24 ] Wu.Z, Palmer.M. *Verb Semantics and Lexical Selection*, *Proceedings of the 32nd Annual Meetings of the Associations for Computational Linguistics*, pages 133-138, 1994.
- [ 25 ] Smaili.A, *Annotation semi-automatique de pages web*, Université Mohamed Boudiaf M'sila, Mémoire de Master, 2016.