

## **Résumé**

Dans le web sémantique, la popularité de l'utilisation de RDF (Resource Description Framework) pour la description des ressources sémantiques est de plus en plus grande. Cependant, l'adoption considérable de RDF a entraîné une augmentation rapide du volume global de ces données, et par conséquent cette situation a mené à la surcharge de la mémorisation, le stockage, et le traitement de ces données. Ainsi, le traitement de cette masse de données par une seule machine nécessite un vaste temps d'exécution. Pour cela, le but de notre travail est de proposer un modèle qui permet tout d'abord d'utiliser les graphes pour la représentation de la connaissance sémantique vue leurs simplicités, et d'utiliser par la suite des outils de partitionnement de graphe tels que METIS qui est basé sur la méthode multi-niveaux afin de réduire la taille des données à traiter.

## **Mots-clés :**

Web sémantique, RDF, SPARQL, Big Data, traitement parallèle, METIS.

# Chapitre 1

## Les données du web sémantique et les SGBD appropriés

### 1.1 Introduction

Ces dernières années, la communauté de recherche du web sémantique a connu des progrès significatifs vers la réalisation de la vision architecturale de ce Web. Les progrès ont été sous forme de normalisation des langages et des technologies Web qui permettent cette vision. Le progrès est vu en termes de niveau de maturité atteint dans chaque couche.

L'annotation de documents, et spécialement l'annotation sémantique à base d'ontologies, semble actuellement l'approche la plus prometteuse pour partager et exploiter l'information sur le Web. Le but majeur de cette technique est la désambiguïsation des documents, afin d'assurer leur manipulation automatique dans le cadre du partage et d'échange d'information sur le Web.

Dans ce chapitre nous présentons une définition du Web sémantique et autre définition sur les formats de données en donnant les composantes et aussi les différents types. Ensuite nous présenterons les outils de développement en présentant les langages de construction et aussi le langage d'interrogation.

## 1.2 Le web sémantique

### 1.2.1 La sémantique

- ✓ logique: qui concerne la relation des signes et de leurs référents.
- ✓ linguistique: étude scientifique du langage du point de vue du sens [1]

### 1.2.2 Le web

Le web désigne en anglais une toile d'araignée. C'est une technologie informatique permettant de consulter des pages regroupées sur des sites, Ce terme a été créé en 1990 par Tim Berners-Lee, considéré comme l'inventeur d'internet, qui développa le premier navigateur et éditeur web nommé (World Wide Web) alors qu'il était chercheur au CERN (Conseil Européen pour la Recherche Nucléaire). Il fut aidé dans ses développements par son collègue Robert Cailliau qui est également l'auteur du premier logo du WWW.[2]

### 1.2.3 L'évolution du web

Le web est sans nul doute une technologie majeure du 21ème siècle. Et si sa nature, sa structure et son utilisation ont évolué au cours du temps, force est de constater que cette évolution a également profondément modifié nos pratiques commerciales et sociales.

- Le web 1.0, encore appelé web traditionnel, est avant tout un web statique, centré sur la distribution d'informations. Il se caractérise par des sites orientés produits, qui sollicitent peu l'intervention des utilisateurs. Les premiers sites d'e-commerce datent de cette époque. Le coût des programmes et logiciels propriétaires est énorme et l'explosion de la bulle dot.com, en 2000, remet en question cette approche de la toile.

- Le web 2.0, ou web social, change totalement de perspective. Il privilégie la dimension de partage et d'échange d'informations et de contenus (textes, vidéos, images ou autres). Il voit l'émergence des réseaux sociaux, des smartphones et des blogs. Le web se démocratise et se dynamise. L'avis du consommateur est sollicité en permanence et il prend goût à cette socialisation virtuelle. Toutefois, la prolifération de contenus de qualité inégale engendre une infobésité difficile à contrôler.

- Le web 3.0, aussi nommé web sémantique, vise à organiser la masse d'informations disponibles en fonction du contexte et des besoins de chaque utilisateur, en tenant compte de sa localisation, de ses préférences, etc. C'est un web qui tente de donner sens aux données. C'est aussi un web plus portable et qui fait de plus en plus le lien entre monde réel et monde virtuel. Il répond aux besoins d'utilisateurs mobiles, toujours connectés à travers une multitude de supports et d'applications malines ou ludiques.
- Le web 4.0, évoqué par certains comme le web intelligent, effraie autant qu'il fascine, puisqu'il vise à immerger l'individu dans un environnement (web) de plus en plus prégnant. Il pousse à son paroxysme la voie de la personnalisation ouverte par le web 3.0 mais il pose par la même occasion de nombreuses questions quant à la protection de la vie privée, au contrôle des données, etc. C'est un terrain d'expérimentation où tous ne sont pas (encore) prêts à s'aventurer ! [3]

#### **1.2.4 Définition du web sémantique**

Le Web sémantique est une extension du Web actuel dans lequel les informations ont une signification bien définie, permettant aux ordinateurs et aux personnes de travailler en coopération, Le but du Web sémantique est de favoriser les formats de représentation de données et les protocoles d'échange communs sur le Web.[3]

L'approche actuelle du Web Sémantique, ou Web des données, est basée sur l'idée de travailler, au niveau du Web, à partir de concepts plutôt que de documents, qui contiendraient alors des informations, ou métadonnées, formalisées pour être traitées automatiquement par des agents logiciel. Le Web de données est fondé sur une pile de langages et protocoles issus du Web et propres au Web Sémantique (illustration figure 1), conçus pour être compris sémantiquement et utilisables par les programmes. Le W3C2 supervise le développement des standards, propre aux Web Sémantique, comme RDF 3, RDFS 4, OWL5 ou SPARQL6 que nous allons décrire.[3]

#### **1.2.5 Les principales composantes du web sémantique**

L'architecture du web sémantique s'appuie sur une pyramide de langages pour représenter des connaissances sur le web en satisfaisant les critères de standardisation, d'interopérabilité et de flexibilité. Cette architecture en couches peut permettre une approche

progressive dans les processus de standardisation et d'acceptation par les utilisateurs. Un langage de la couche haute doit être une extension du langage de la couche au-dessous.

La vision du Web sémantique est présentée dans la figure1. L'architecture de cette vision est organisée en couches dont seulement les couches basses sont aujourd'hui relativement stabilisées. Chaque niveau repose sur les résultats définis au niveau inférieur, c'est-à-dire que chaque niveau est progressivement plus spécialisé et plus complexe que le niveau précédent.[4]

La liste suivante introduit la fonction principale de chaque couche dans l'architecture du web sémantique :

- XML est utilisé comme couche de base syntaxique du web sémantique. Le langage XML est actuellement considéré comme un standard pour le transport de données sur le web.
- La couche RDF représente les métadonnées pour les ressources web.
- La couche « ontologie », fondée sur une formalisation commune, spécifie la sémantique de métadonnées fournies dans le web sémantique.
- La couche « logique » s'appuie sur des règles d'inférence qui permettent le raisonnement intelligent exécuté par des agents logiciels.

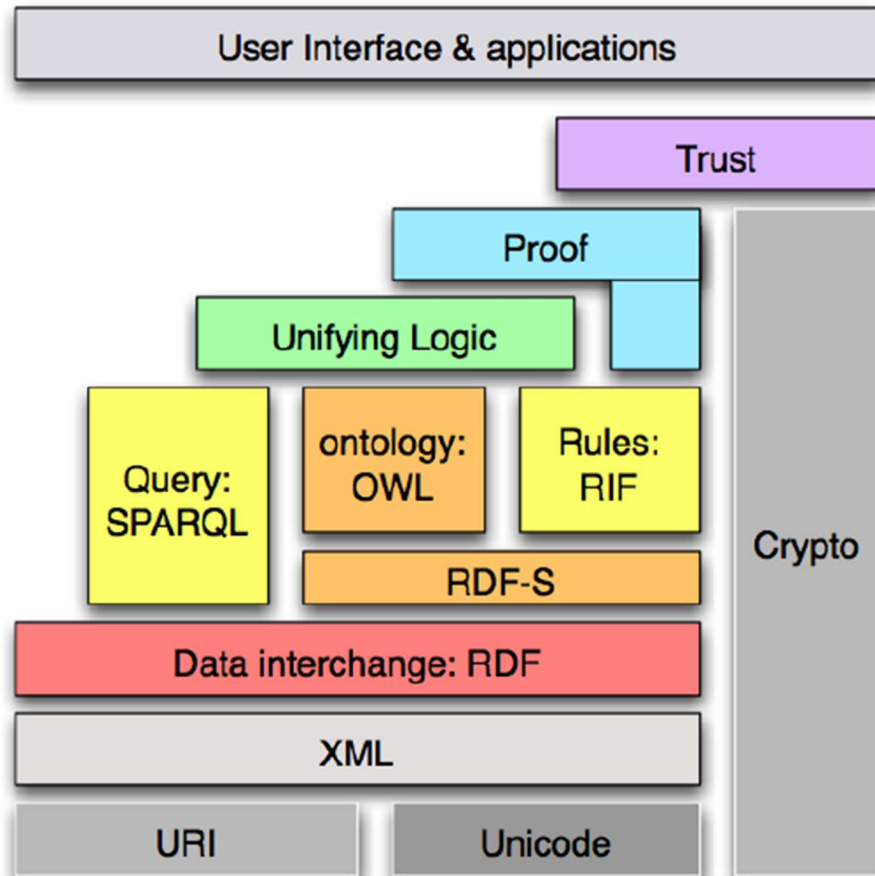


Figure 1 - Les couches du Web Sémantique[4].

## 1.3 Les Technologies de Base du Web Sémantique

### 1.3.1 RDF

Comme il a été mentionné dans la section précédente, RDF est un élément fondamental du web sémantique puisqu'il permet de représenter des ressources sur le web de manière uniforme pour les agents logiciels là où ceux-ci ne voient dans un document texte qu'une succession de caractères inexploitable [5].

RDF (The Resource Description Framework) est un langage à base d'XML pour décrire des ressources sur le web. RDF est une création pour le traitement des métadonnées. Il fournit l'interopérabilité entre les applications qui échangent de l'information non compréhensible par les machines sur le Web. Il augmente la facilité de traitement automatique des ressources Web.

RDF peut être utilisé dans une variété de champs d'application, dans la découverte de ressources pour fournir une meilleure efficacité aux moteurs de recherche, dans le catalogage pour décrire le contenu et les relations entre les contenus disponibles sur un site Web particulier, sur une page, ou sur une bibliothèque numérique et dans l'évaluation du contenu, en décrivant des ensembles de pages qui représente un simple et unique "document", pour décrire les droits sur la propriété intellectuelle des pages Web, et pour indiquer les préférences de confidentialité (privacy policy) d'un utilisateur aussi bien que les politiques de confidentialité d'un site Web [6].

Afin de décrire ces ressources, RDF se base sur la notion de triplets, qui permettent de définir des assertions au sujet de celles-ci [7]. Chaque triplet se compose de :

- ✓ un sujet, qui représente la ressource à laquelle on assigne une propriété, identifiée par une URI (Uniform Ressource Identifier);
- ✓ un prédicat, désigne l'attribut ou la relation utilisée pour décrire une ressource. Le prédicat est également identifié par une URI;
- ✓ un objet, c'est la valeur de la propriété. Celle-ci peut être de type primitif (chaîne de caractères, entier ...) ou elle peut être à nouveau une ressource. Elle peut ainsi être à son tour sujet d'un autre triplet conduisant à la formation d'un graphe, les nœuds tout comme les arcs étant représentés par des URIs.

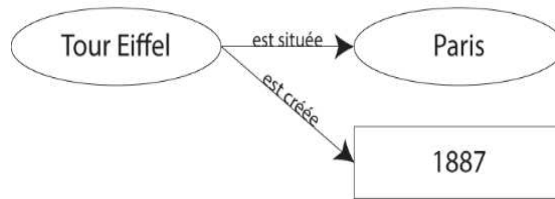
Par exemple : Soit la phrase simple suivante : la tour Eiffel est créée en1887 a paris .

Cette phrase est composée des parties suivantes :

**Tableau 1 : Le triplet RDF**

Sujet (ressource)	Tour Eiffel
Prédicat (propriété)	Créée/située
Objet (littéral)	1887/ Paris

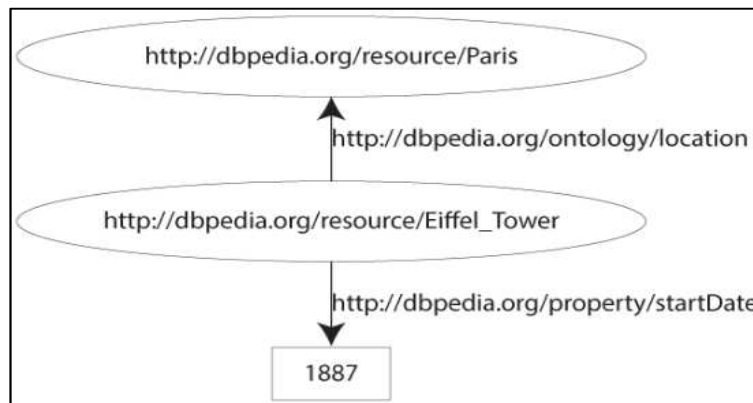
La phrase sera donc représentée par le diagramme suivant :



**Figure 2- représentation graphique de triplets RDF**

Pour représenter correctement ces triplets, il aurait fallu l'écrire de cette manière :

- ✓ ([http://dbpedia.org/resource/Eiffel\\_Tower](http://dbpedia.org/resource/Eiffel_Tower), <http://dbpedia.org/ontology/location>, <http://dbpedia.org/resource/Paris>) ;
- ✓ ([http://dbpedia.org/resource/Eiffel\\_Tower](http://dbpedia.org/resource/Eiffel_Tower), <http://dbpedia.org/property/startDate>, 1887)



**Figure -3 représentation graphique RDF**

### 1.3.2 Rdfs

Il est nécessaire, de fournir un vocabulaire (classes et propriétés) pour donner une sémantique ou un sens aux informations stockées sous forme de triplets RDF. Pour cela, W3C propose une extension de RDF appelée RDFS [6]. Ce dernier, est un langage extensible de représentation des connaissances qui permet de décrire précisément par vocabulaires les ressources d'un domaine donné et les relations entre elles. RDFS fournit des éléments de base pour la définition d'ontologies ou des vocabulaires destinées à structurer les ressources RDF.

L'ensemble de ces ressources appartenant à un même groupe constitue ce que l'on appelle une classe ou une propriété.

- Une classe est le premier élément important de ce vocabulaire. rdfs (Class) représente le type de la ressource obtenue comme valeur lorsque la propriété rdf (type) est appliquée à une ressource et chaque ressource est une instance de cette classe.
- une propriété rdf : type permet d'exprimer l'appartenance d'une ressource à une classe.

### 1.3.3 Ontologies

#### 1.3.3.1 Définition d'ontologie

Le terme ontologie est défini selon deux points de vue :

- ✓ En philosophie : une ontologie signifie science ou théorie de l'être, c'est-à-dire, l'étude des propriétés générales de ce qui existe.
- ✓ En informatique le sens est différent, d'après Studer [9]

« Une ontologie est une spécification formelle et explicite d'une conceptualisation partagée. »

En version originale: "An ontology is a formal, explicit specification of a shared conceptualization."

Dans ce contexte, plusieurs définitions des ontologies ont été proposées est la plus fiable, déclare que « une ontologie définit les termes et les relations de base du vocabulaire d'un domaine ainsi que les règles qui permettent de combiner les termes et les relations afin de pouvoir étendre le vocabulaire » [9].

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. Plus simplement, on peut aussi dire que l'«ontologie est aux données ce que la grammaire est au langage».

#### 1.3.3.2 Composants d'une ontologie

Une ontologie est l'ensemble structuré des termes et des concepts représentant le sens d'un champ d'informations. Une ontologie constitue en soi un modèle de données représentatif

d'un ensemble de concepts dans un domaine, ainsi que les relations entre ces concepts. Elle est employée pour raisonner à propos des objets du domaine concerné. Ainsi une ontologie représente trois types de caractéristiques :[10]

- ✓ Les concepts (les classes)
- ✓ Les propriétés
- ✓ Les relations

- **Les concepts :**

Les classes représentent les concepts pris au sens large. Un concept peut être tout ce qui peut être évoqué : description d'une tâche, d'un objet matériel, d'une fonction, d'une action, d'une stratégie ou d'un processus de raisonnement, etc.[10]

- **Les propriétés**

Une ontologie est, non seulement le repérage et la classification des concepts mais c'est aussi des caractéristiques qui leur sont attachées et qu'on appelle les propriétés (appelées, aussi, les attributs), Par exemple :

Dans une ontologie sur le monde animal, on pourra avoir les concepts de « mammifère » ou d' « oiseau » pour lesquels est précisé le type de tégument, respectivement à poil et à plume. En pratique, un attribut « tégument » pourra être attaché aux concepts et sa valeur variera suivant le concept auquel on fait référence [10].

- **Les relations :**

Les relations représentent un type d'association entre les concepts du domaine.[10]

### 1. 3.3.3 Rôles des ontologies

Les ontologies permettent d'améliorer la communication entre machines et entre humains et machines ou encore entre humains par le biais de logiciels. Les propriétés de ce type de structure de données ont permis de diversifier leur utilisation à différentes applications, en particulier la gestion des connaissances et le web sémantique [11]. Les ontologies permettent de :

- Définir un vocabulaire commun et partagé par une communauté de pratiques.

- Avoir une compréhension commune.
- Donner un sens unique à des entités du monde réel.
- Construire et découvrir de nouvelles informations et/ou connaissances à partir des ontologies et des ressources existantes.
- Améliorer les processus de recherche d'informations.
- Faciliter la communication entre agents logiciels.

Le langage d'ontologie le plus largement utilisé est le «Web Ontology Language», Qui a curieusement l'acronyme «OWL».

### 1.3.4 OWL

Le langage OWL a été conçu pour être le langage d'ontologie Web. OWL fait partie du paquet croissant des recommandations liées au Web sémantique. OWL est, tout comme RDF, un langage XML profitant de l'universalité syntaxique de XML. Il est Fondé sur la syntaxe de RDF/XML, OWL offre un moyen d'écrire des ontologies web. OWL intègre des outils de comparaison des propriétés et des classes : identité, équivalence, contraire, cardinalité, symétrie, transitivité, disjonction, etc. OWL offre aux machines une plus grande capacité d'interprétation du contenu web que RDF et RDFS, grâce à un vocabulaire plus large et à une vraie sémantique [11].

Le langage OWL offre trois sous-langages d'expression croissante destinés à des communautés de développeurs et d'utilisateurs spécifiques :

✓ OWL Lite : est le sous langage de OWL le plus simple. Il est destiné aux utilisateurs qui ont besoin d'une hiérarchie de concepts simple. OWL Lite est adapté, par exemple, aux migrations rapides depuis d'anciens thésaurus [12].

✓ OWL DL : est plus complexe que OWL Lite, permettant une expressivité bien plus importante. OWL DL est fondé sur la logique descriptive (d'où son nom, OWL Description Logics), un domaine de recherche étudiant la logique, et conférant donc à OWL DL son adaptation au raisonnement automatisé. Malgré sa complexité relative face à OWL Lite, OWL-DL garantit la complétude des raisonnements (toutes les inférences sont calculables) et leur décidabilité (leur calcul se fait en une durée finie) [12].

✓ OWL Full : est la version la plus complexe d'OWL, mais également celle qui permet le plus haut niveau d'expressivité. OWL Full est destiné aux situations où il est plus important d'avoir un haut niveau de capacité de description. OWL Full offre cependant des mécanismes intéressants, comme par exemple la possibilité d'étendre le vocabulaire par défaut de OWL [12].

## 1.4 Les Formes de Représentation des Données Sémantiques

Les données sémantique peuvent donc être exprimé en divers formats, . Parmi eux on trouve : XML, N-Triples, JSON-LD, TriG, TriX, , et NQuads. Et appelés sérialisations.

On detaille quelque format avec un exemple

### 1.4.1 XML

Cette sérialisation permet d'exprimer des données RDF selon la syntaxe XML, Tel que XML( langage de balisage extensible) [8] est un méta-langage permettant de représenter un document texte de manière arborescente en utilisant un système de balisage. XML fournit ainsi un cadre de structuration des données qui peut être employé pour la définition rapide et sans ambiguïté syntaxique d'un format de document. Ce langage a été élaboré pour faciliter l'échange, le partage et la publication des données à travers le web. Ainsi, la majorité des langages/modèles proposés pour le web sémantique sont exprimés en XML.[13]

XML permet de structurer un document en définissant ses propres balises en fonction des besoins et sans tenir compte ni de la signification de cette structure ni des systèmes informatiques qui vont l'exploiter. Des standards comme XPath et XQuery ont été développés afin de parcourir et d'interroger l'arborescence XML des documents. [13]

Exemple :

```
<?xml version="1.0"?>
<rdf:RDF>
<rdf:Description
about="http://www.w3.org/2001/XMLSchema#">
<tour Eiffel>
<cd:local>paris </cd:local>
<cd:payes>france</cd:payes>
```

```
<cd:année >1887 </cd:année >
</tour Eiffel>
</rdf:Description>
</rdf:RDF>
```

### 1.4.2 Turtle

Turtle est une syntaxe RDF parfaitement lisible par un humain. Son écriture est très simple, puisque la principale règle appliquée est que chaque triplet est séparé par « . ». et Un préfixe p est défini par @prefix qui associe un espace de noms à un préfixe [8] .

Voici un petit exemple de Turtle : Qui représente l'année de la création de la Tour Eiffel, et son endroit.

```
@prefix dbpedia-owl: <http://dbpedia.org/ontology/> .
@prefix dbpprop: <http://dbpedia.org/property/> .
@prefix dbpedia: <http://dbpedia.org/resource/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
dbpedia:Eiffel_Tower dbpedia-owl:location dbpedia:Paris
dbpedia:Eiffel_Tower dbpprop:startDate "1887"^^xsd:integer
dbpedia:Eiffel_Tower dbpprop:name "Tour Eiffel"@fr
dbpedia:Eiffel_Tower dbpprop:name "Eiffel Tower"@en
```

### 1.4.3 Les graphes

Un graphe est un ensemble de points nommés nœuds (parfois sommets ou cellules) reliés par des traits (segments) ou flèches nommées arêtes (ou liens ou arcs). L'ensemble des arêtes entre nœuds forme une figure similaire à un réseau. Il existe deux types de graphe : orientés et non orientés. Le web sémantique utilise les graphes orientés pour la représentation de ses données. Un graphe orienté est un graphe dirigé sans paire symétrique de bords dirigés.[14]

Un graphe sémantique est tout simplement un graphe qui schématise les triplets RDF. Afin d'éclaircir le principe des graphes sémantiques, Nous présentons dans la figure 7 un graphe sémantique qui est constitué de deux nœuds, " Mostaganem " et "Algerie", reliés par un arc nommé "est\_situé\_en". En d'autres termes :

Mostaganem : est ce qu'on appelle une ressource, ou encore un sujet, une source.

Est\_situé\_en : est ce qu'on appelle un prédicat ou encore une propriété.

Exemple :

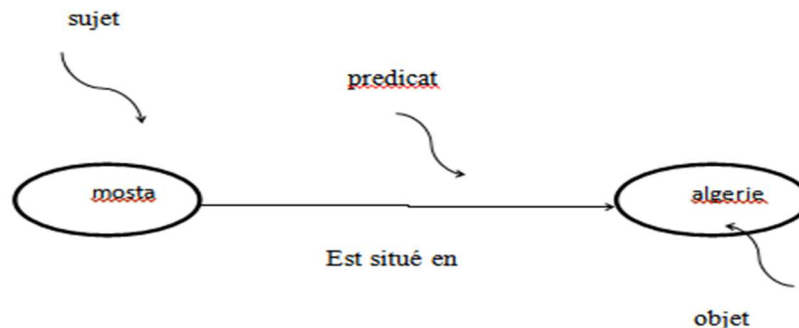


Figure – 4 Graphe orienté

## 1.5 Langage d'interrogation d'ontologie (SPARQL)

A nos jours le stockage d'information est de quantité énorme et de manière structurée et il n'aurait aucune utilité s'il était impossible d'y accéder. Dans le cas du Web sémantique il existe des langages de requêtes connus tels que RDQL1, nRQL2, SPARQL. SPARQL est le langage standard d'interrogation du Web sémantique, plus particulièrement de graphe RDF.

SPARQL (acronyme obscur dérivé de SQL) est un langage de requêtes destiné à interroger les bases de données (fichiers) RDF, standardisé par le W3C et implémenté en divers langages, notamment en Java avec le framework JENA. La syntaxe du SPARQL ressemble à celle du SQL car elle a été inspirée de ce dernier. [15]

- SPARQL permet d'exprimer deux types de requêtes [15]:
  - ✓ Interrogative : extraction par SELECT des sous graphes d'un graphe RDF, satisfaisant les conditions spécifique dans la clause WHERE.
  - ✓ Constructive : génération d'un autre graphe à partir du graphe résultant de la requête d'interrogation.
- La syntaxe d'une requête SPARQL est définie de plusieurs parties suivantes [16] :
  - ✓ PREFIX : définit une étiquette correspondant à l'URI d'un dépôt de données à interroger et permettent de définir des espaces de noms.
  - ✓ SELECT : identifie les valeurs à retourner parmi les variables de la clause WHERE.

- ✓ ASK : interroge un dépôt ne retourne que true/false détermine la présence des résultats.
- ✓ CONSTRUCT : crée un nouveau graphe RDF à partir des résultats de la requête.
- ✓ DESCRIBE : retourne un graphe RDF décrivant les ressources trouvées lors d'une interrogation.
- ✓ FROM : identifie les sources de données à interroger.
- ✓ WHERE : associe les valeurs à retourner avec les objets du fichier RDF.
- ✓ DISTINCT : garantit l'unicité des résultats.
- ✓ REDUCED : supprime toutes les solutions présentent plus d'une fois dans les résultats.
- ✓ FILTER : ajoute des contraintes sur les résultats.
- ✓ OPTIONAL : permet de définir l'instruction suivante comme optionnelle.
- ✓ UNION : combine les résultats de plusieurs instructions.
- ✓ ORDER BY : trie les résultats.
- ✓ LIMIT : limite le nombre de résultats.
- ✓ OFFSET : fait commencer les résultats générés après le nombre de résultats indiqué.

Exemples sur quelque requête SPARQL simple :

**Exemple 1 :**

```

SELECT ?personne
WHERE { ?personne rdf : type
Personne.
ORDER BY ?age
    
```

**Résultat :**

e	Personne
ddine	SalahE
	Sarra

Cette requête nous affiche toutes les personnes présentes dans le graphe de données et qui possède un âge, l'introduction de ORDER BY est faite pour ordonner les résultats obtenus par la requête de manière ascendante par rapport à l'âge des personnes.

**Exemple 2 :**

**Résultat :**

```

SELECT ?personne ?age

WHERE { ?personne rdf:type
Personne.

?personne ns:Age ?age.

FILTER (?age >= 23) }
    
```

Personne	Age
Sarra	23

La présence de FILTER filtre le résultat obtenu en introduisant une contrainte ( ? age >= 23 ) et le résultat obtenu de la requête donne toutes les personnes ayant l'âge supérieur ou égal à 23.

**Exemple 3 :**

**Résultat :**

```

SELECT ?personne ?nom ?ville

WHERE { ?personne rdf:type
Personne.

?personne ns:Nom ?nom.

?personne ns:Ville ?ville.

?personne ns:Age ?age }

ORDER BY ?age

OFFSET 1
    
```

Personne	Nom	Ville
Salah Eddine	Raya	Paris

Cette requête récupère les personnes, leur nom et leur ville les résultats sont trié par leur âges en ordre croissant et on affiche à partir du deuxième résultat à cause de la présence de OFFSET 1.

**Exemple 4 :**

**Résultat :**

```
ASK { ?personne ns : Ville "mosta" }
```

Réponse : Oui

Cette requête vérifie la correspondance de la variable ville dans le graphe RDF.

**Exemple 5 :**

**Résultat :**

```
CONSTRUCT { ?personne ns : Pays
"Algérie" }

WHERE { ?personne red :ID "Sarra" }
```

Imad ns : Pays "Algérie".  
CONSTRUCT :retourne un graphe.

## 1.6 Les types de base de données

Une base de données est une collection de données organisées de façon à être facilement accessibles, administrées et mises à jour. Les bases de données peuvent être classées par le type de contenu qu'elles renferment : bibliographique, full text, images ou des nombres...etc

### 1.6.1 NOSQL

Le terme « NoSQL » a été inventé en 2009 lors d'un événement sur les bases de données distribuées le terme désigne une famille de systèmes de gestion de base de données (SGBD), est une combinaison de deux mots : No et SQL. Tel que le NO est un

acronyme qui signifie « Not only », c'est-à-dire : « non seulement », c'est donc une manière de dire qu'il y a autre chose que les bases de données relationnelles. [17]

NoSQL est un mouvement très récent, qui concerne les bases de données. L'idée du mouvement est simple : proposer des alternatives aux bases de données relationnelles Pour coller aux nouvelles tendances et architectures du moment, notamment le Cloud Computing. Les axes principaux du NoSQL sont une haute disponibilité et un partitionnement horizontal des données, au détriment de la cohérence. Alors que les bases de données relationnelles actuelles sont basées sur les propriétés ACID (Atomicité, Consistance ou Cohérence, Isolation et Durabilité). En effet, NoSQL ne vient pas remplacer les BD relationnelles mais proposer une alternative ou compléter les fonctionnalités des SGBDR pour donner des solutions plus

Intéressantes dans certains contextes. Le NoSQL regroupe de nombreuses bases de données, récentes pour la plupart, qui se différencient du modèle SQL par une logique de représentation de données non relationnelle. Leurs principaux avantages sont leurs Performances et leur capacité à traiter de très grands volumes de données. [17]

#### **1.6.1.1. Type de données NoSQL :**

Il existe quatre grandes catégories de bases de données NoSQL :

- Base de données clé-valeur (figure) : ces base de données s'articulent sur une architecture très basique fondé sur le HashMap c'est-à-dire qu'une valeur, un nombre ou du texte est stocké grâce à une clé, qui sera le seul moyen d'y accéder.
- Base de données Orientées Graphe : Une base de données orientée graphes utilise des structures de graphe pour stocker, mapper et rechercher des relations. Elles permettent de résoudre des problèmes très complexes qu'une base de données relationnelle serait incapable de faire. Par exemple, Les réseaux sociaux (Facebook, Twitter,etc).
- Les bases de données orientées documents : elles présentent une structure différente par rapport aux bases orientée clé-valeur. Ce type de bases de données contient des documents qui sont considérés comme les lignes de cette table. Chaque document contient en général une clé unique qui l'identifie.
- Bases de données orientées colonne : dans une base de données relationnelle les colonnes sont statiques et elles sont présentes pour chaque ligne, par contre dans une base de données orientée colonne les colonnes sont dynamiques car elles sont présentes seulement si il y a une nécessité. Dans ce cas, il est possible d'ajouter des colonnes dynamiquement [17].

## 1.6.2 BigData

Le terme Big Data signifie méga données, grosses données ou encore données massives [17]. Les BigData désignent un ensemble très volumineux de données qu'aucun outil classique de gestion de base de données ou de gestion de l'information ne peut vraiment travailler. Ces données sont produites en permanence et à une vitesse croissante, elles proviennent de tous les secteurs (les réseaux sociaux, les mails, les conversations téléphoniques, les informations climatiques, les GPS (global positioning system), les téléviseurs connectés, les sites web etc.) Cette masse de données est utilisée pour créer des applications à but analytique, qui traitent des données pour en tirer du sens et faciliter la vie des utilisateurs.

Le BigData rassemble tous les types de base de données tels que les données RDF qui sont volumineuses, ce qui nécessite un mode de stockage particulier, pour ce nouveau mode de stockage. Cependant, il faut respecter la règle des 3V :[17]

- Volume : stocker énormément d'information.
- Variété : stocker beaucoup de données de toutes sortes.
- Vitesse : pouvoir avoir accès rapidement à toutes ces données.

## 1.7 Conclusion

Certains spécialistes du Web affirment que le Web sémantique offre toutes les possibilités pour favoriser l'évolution de la connaissance humaine dans son ensemble. En effet, si l'on songe que la "domestication" du Web mettra à la portée de ses utilisateurs une information élargie sur des domaines spécifiques qui jusqu'à maintenant n'était repérable que partiellement, il s'agit d'un changement très important dans le monde de l'information.

Dans ce chapitre nous avons défini des notions sur Les formats des données sémantiques qui ont été utilisés pour représenter diverses connaissances du monde réel, et le traitement automatique des informations de ses connaissances. Nous avons présenté une définition générale sur Les formats des données sémantiques en donnant les composants, les différents types, les outils de développement. Nous avons bien détaillé les outils de développement en définissant Les types de base de données tel que le NOSQL et les différents langages de construction d'ontologies et leurs propriétés en se passant sur le langage d'interrogation d'ontologie SPARQL.



## Chapitre 2

# Les methodes de partitionnement de graphe : metis

### 2.1 Introduction

Le partitionnement de graphe a pour but de découper un graphe en différentes parties (partitions) qui satisfont certains contraintes (telle que l'équilibre des parties) et qui optimisent une certaines fonction objectif (telle que le coût de coup). Il possède de nombreuses applications comme la conception de circuits intégrés électroniques, la répartition de charge pour les machines parallèles ou la segmentation d'images. Cependant, le partitionnement de graphe est un problème complexe (NP-complet), dont la solution ne peut pas être trouvée au moyen d'une méthode de résolution exacte [18]. Pour ce type de problème, nous avons étudié et analysé des méthodes de partitionnement classiques comme la méthode de multi-niveaux ou les algorithmes de type Kernighan-Lin, etc.

mais seuls les algorithmes multi-niveaux permettent d'obtenir des résultats de bonne qualité sur des graphes de grandes tailles.

Nous pouvons citer comme logiciels parallèles implémentant ces schémas multi-niveaux : Scotch , chaco, ou encore MeTiS .

## 2.2 Notions mathématiques

Avant de donner une définition précise de ce problème, ce chapitre présente quelques rappels de mathématiques et de théorie des graphes utilisés dans ce thème. Ils seront suivis d'une description formelle du partitionnement de graphe et de ses objectifs.

### 2.2.1 Définition du Cardinal d'un ensemble

Soit  $X$  un ensemble fini d'éléments. Le cardinal de l'ensemble  $X$  est égal au nombre d'éléments de  $X$ , et on le note  $\text{card}(X)$ . En mathématiques, la partition d'un ensemble  $S$  d'objets se définit comme suit :  $\text{card}(x) = |x|$ .

### 2.2.2 Définition de la partition

Soit  $S$  un ensemble quelconque de sommet, et  $P$  un ensemble regroupant les sous-ensembles de  $S$  comme suit  $p = \{S_1, \dots, S_k\}$  qui applé une partition de  $S$  si seulement si [19]:

- ✓ Aucun élément de  $p$  n'est pas vide.
- ✓ L'union des éléments de  $P$  est égal à  $S$ .
- ✓ Les éléments de  $P$  sont deux à deux disjoints.

### 2.2.3 Définition d'un graphe

Soient  $S$  un ensemble de  $n_S \in \mathbb{N}^*$  éléments et  $A$  un ensemble de  $n_A \in \mathbb{N}$  couples d'éléments de  $S$ . On appelle graphe  $G$  le couple  $(S, A)$ . Les éléments de  $S$  sont appelés sommets du graphe et ceux de  $A$ , les arcs ou arêtes du graphe, suivant qu'ils sont orientés ou non [19].

### 2.2.4 Définition du partitionnement de graphe

Le partitionnement de graphe consiste à découper un graphe en un sous ensemble de graphe c'est-à-dire partitionner l'ensemble des sommets d'un graphe [19].

### 2.2.5 Catégorie du problème du partitionnement de graphe

Les chercheurs ont adapté deux catégories regroupant les problèmes de partitionnement :

- ✓ Partitionnement contraint.
- ✓ Partitionnement non contraint.

### **Le partitionnement de graphe contraint**

consiste à trouver une partition  $P_k = \{S_1, \dots, S_k\}$  d'un graphe  $G$  qui minimise une fonction objectif  $f$  et dont la balance de partitionnement soit unitaire, i.e. les parties doivent avoir le même poids, à une unité près [19].

Ainsi, le seul réel objectif du partitionnement de graphe contraint est de minimiser le coût de coupe des arêtes entre les parties du graphe. Tel que

- Le poids d'une partie  $S_i$  de  $P_k$  est :

$$\text{Poids}(S_i) = \sum \text{poids}(S_i) .$$

- Le poids moyen d'une partie  $S_i$  de  $P_k$  est :

$$\text{Poids}_{\text{moyen}}(S, K) = \frac{\sum_{S \in S} \text{poids}(S)}{K} .$$

- La balance de la partition  $P_k$  est :

$$\text{Bal}(P_k) = \frac{\text{MAX} \text{poids}(S_i)}{\text{poids moy}} .$$

- Le cout de coupe est minimisé :

$$\text{Cut}(S_1, S_2) = \sum \text{poids}(s_1, s_2)$$

### **Le partitionnement de graphe non contraint**

a pour but de minimiser le coût de coupe dans la limite de parties dont la taille est inversement proportionnelle à ce coût de coupe tel que [19] :

Ratio de coupe (ratio cut) :

$$\text{Ratio}(P_k) = \frac{\sum_{i=1}^k \text{cut}(s_i, s-s_i)}{\text{poids}(s_i)}$$

## 2.3 Le partitionnement du graphe :

Le mot “partitionnement” correspond au mot « partitioning » en anglais, qui exprime la création d’une partition. La partition est le résultat de la division en parties d’un ensemble. En mathématiques, une partition d’un ensemble E est une famille de sous-ensembles de E qui sont disjoints deux à deux, et dont la réunion forme l’ensemble E [20].

Le partitionnement de graphes est une technique très connue en informatique Etant donné un graphe non-orienté  $G = (S, A)$ , où S est l’ensemble des sommets et A est l’ensemble des arêtes qui relient des paires de sommets. Les sommets et les arêtes peuvent être pondérés, où  $|s|$  est le poids du sommet s et  $|a|$  est le poids de l’arête a. Le problème du partitionnement de graphe consiste à diviser G en k partitionnements disjoints. Du point de vue des mathématiques, on peut partitionner les sommets ou bien les arêtes. Par contre, dans la plupart des applications, on ne s’intéresse qu’au partitionnement des sommets du graphe.

Soit un graphe  $G = (S, A)$  et un ensemble de k sous-ensembles de S, noté  $P_k = \{S_1, S_2, \dots, S_k\}$ . On dit que  $P_k$  est une partition de G si [19] :

- Aucun sous-ensemble de S qui est élément de  $P_k$  n’est vide :

$$\forall i \in \{1, \dots, k\}, S_i \neq \emptyset$$

- Les sous-ensembles de S qui sont éléments de  $P_k$  sont disjoints deux à deux :

$$\forall (i,j) \in \{1, \dots, k\}, i \neq j, S_i \cap S_j = \emptyset$$

- L’union de tous les éléments de  $P_k$  est S :

$$\bigcup_{i=1}^k S_i = S$$

- Les éléments  $S_i$  de  $P_k$  sont appelés les parties de la partition.

Le nombre k est appelé le cardinal de la partition, ou encore le nombre de parties de la partition. Dans le cas  $k = 2$ , on a le problème de bisection.

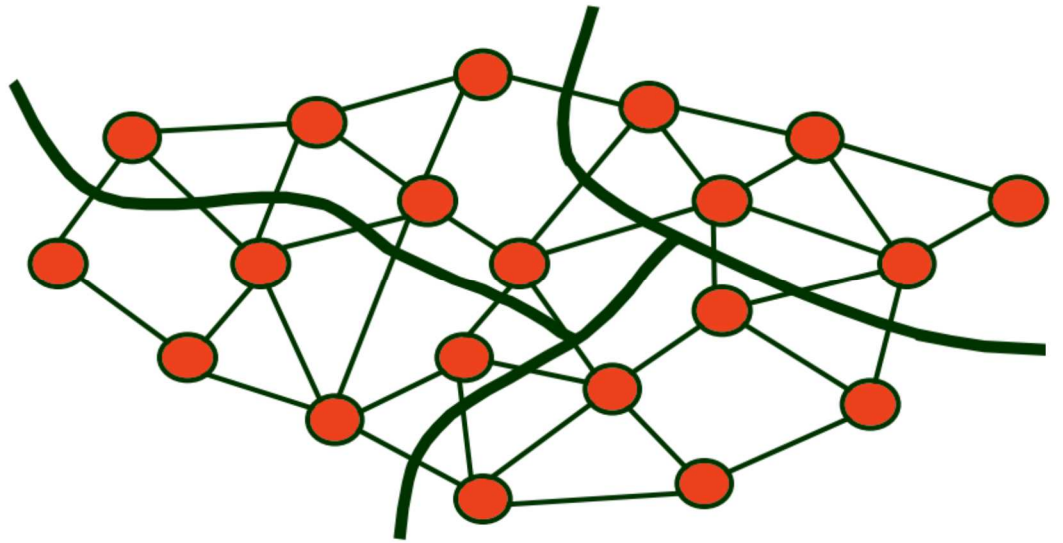


Figure-5 Partitionnement d'un Graphe avec  $k=4$

## 2.4 le principe du partitionnement

Le partitionnement du graphe est une technique employée dans plusieurs domaines. Il est utilisé pour résoudre un grand nombre de problèmes d'optimisation qui sont modélisée sous la forme de graphe.

L'augmentation croissante de la taille des données provoque des problèmes qui peuvent être résolus par le partitionnement des données. (Le partitionnement peut plus être traité de façon séquentielle sur un unique ordinateur), Il est donc essentiel de concevoir des algorithmes parallèles de partitionnement de graphe (car les données rdf/owl peuvent être représentées par les graphes), capables de traiter des graphes de grandes tailles.

La plupart des méthodes actuelles de partitionnement de graphe utilisent un schéma multi-niveaux, Parce que la méthode multi-niveaux est très performante pour résoudre des problèmes de partitionnement de graphe dans lequel le graphe a partitionner est contracté successivement pour obtenir une famille de graphes de plus en plus petits, avec une structure topologique similaire, de sortes qu'une partition initiale calculée sur le plus petit graphe puisse être propagée, par prolongations et raffinements successifs, jusqu'à obtenir un partitionnement du graphe initial [20].

## 2.5 les outils de partitionnement

Le partitionnement de graphe est un problème NP-complet. Dans la littérature, Il existe plusieurs techniques et des logiciels qui donnent des bons résultats comme [20]:

- chaco (sandia national laboratories, usa)
- scotch (inria/labri, bordeaux, france)
- metis (university of minnesota, usa)

Ce sont eux qui sont utilisés pour le problème de la répartition de charge dans les machines parallèles, Et dans notre etude on s'intéresse de métris

### 2.5.1 Métris:

METRIS est un progiciel permettant de partitionner de grands graphes irréguliers, de partitionner de grands maillages et de calculer des ordonnances de remplissage de matrices creuses. Les algorithmes de METRIS sont basés sur un partitionnement multi-niveaux décrit

Les algorithmes traditionnels de partitionnement de graphe calculent une partition d'un graphe en opérant directement sur le graphe d'origine comme illustré à la figure 5 Ces algorithmes sont souvent trop lents et / ou produisent des partitions de mauvaise qualité.

D'autre part, les algorithmes de partitionnement multi-niveaux adoptent une approche complètement différente [21]. Ces algorithmes, comme illustré à la figure 6, réduisent la taille du graphique en réduisant les sommets et les arêtes, partitionnent le graphique plus petit, puis le décochent pour construire une partition pour le graphe d'origine. METRIS utilise de nouvelles approches pour réduire successivement la taille du graphe et affiner la partition pendant la phase de décohesion. Pendant le grossissement, METRIS utilise des algorithmes qui facilitent la recherche d'une partition de haute qualité sur le graphe le plus grossier. Pendant le raffinement, METRIS se concentre principalement sur la partie du graphique qui est proche de la limite de la partition. Ces algorithmes hautement paramétrés permettent à METRIS de produire rapidement des partitions de haute qualité pour une grande variété de graphes.

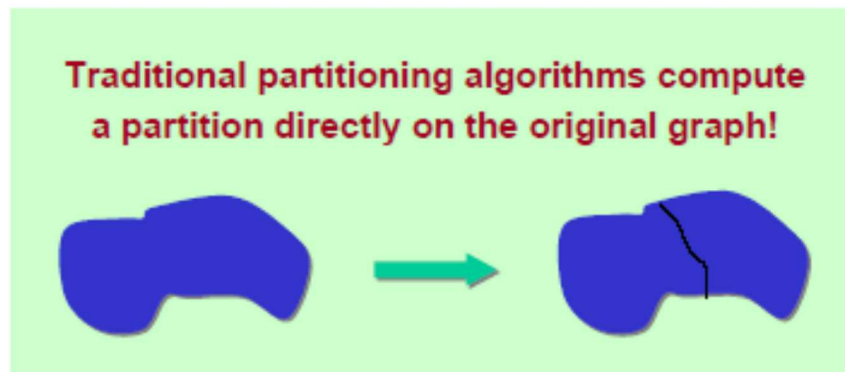


Figure-5 schéma de partitionnement traditionnel

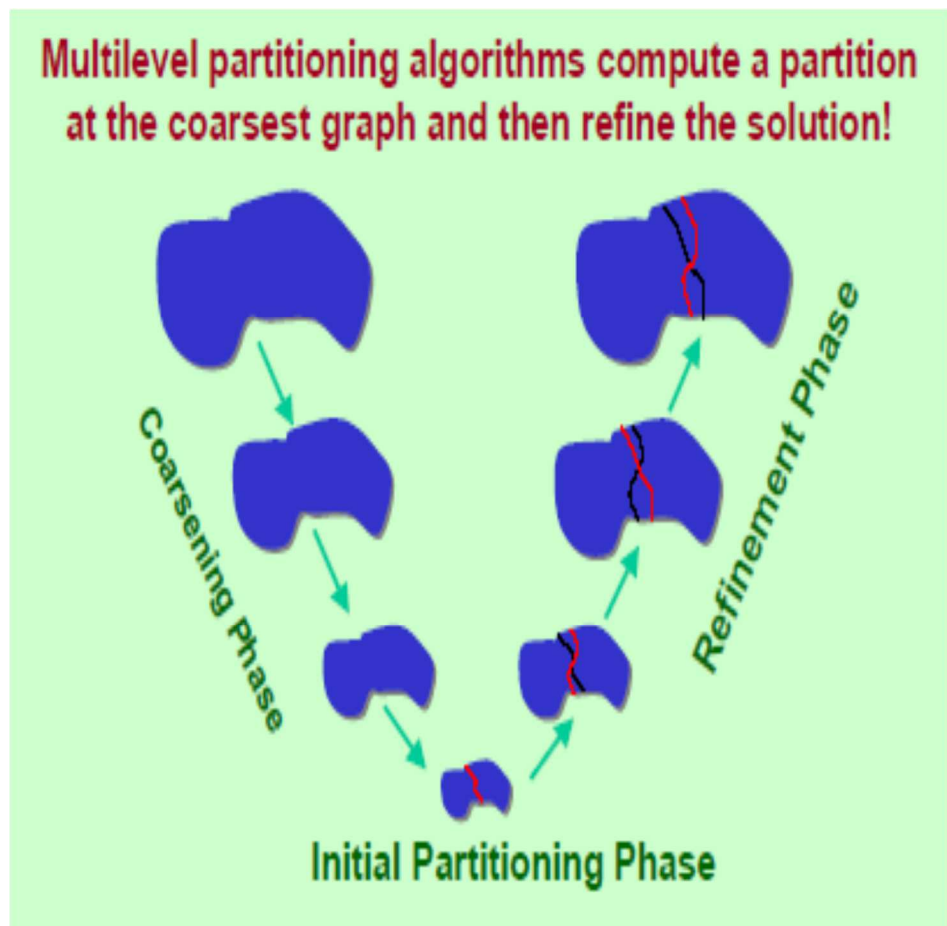


Figure-7 Schéma de la méthode Multi-niveaux

## 2.5.2 Les principales caractéristiques de METIS sont les suivantes

Tout d'abord, il fournit des partitions de grande qualité. Il est utilisé dans plusieurs applications relatives à divers domaines, à savoir la méthode des éléments finis, la programmation linéaire, la conception des circuits VLSI [21]. METIS produit des partitions qui sont souvent meilleures que celles produites par d'autres algorithmes actuels. Les partitions produites par METIS sont toujours 10% à 50% meilleures que celles produites par les algorithmes de partitionnement spectral.

Metis est très rapide. Des expériences sur une large gamme de graphiques ont montré que METIS est de un à deux ordres de grandeur plus rapide que les autres algorithmes actuels de partitionnement[21]. Des graphes avec plusieurs millions de sommets peuvent être partitionnés en 256 parties en quelques secondes sur les stations de travail de la génération actuelle.

## 2.5.3 Les avantages de metis

- Fournit des partitions de haute qualité.
- C'est extrêmement rapide.
- Fournit des ordres de remplissage bas.[21]

## 2.6 méthodes de partitionnement

### 2.6.1 Méthode spectrale

L'utilisation de la méthode spectrale pour résoudre les problèmes de partitionnement de graphe est très ancienne. Cette méthode est basée sur le théorème spectral de l'algèbre linéaire. Ce théorème permet d'affirmer la diagonalisation des matrices réelles. Il justifie également la décomposition des matrices symétriques réelles en valeurs propres dans une base ortho-normale de vecteurs propres. Le problème de partitionnement de graphes peut être ramené à la résolution d'un système numérique  $Mx = \lambda x$ . Résoudre ce système numérique consiste à trouver une base orthogonale de vecteurs propres de la matrice M [22].

## 2.6.2 La méthode multi-niveaux :

La méthode multi-niveaux est généralement la méthode la plus utilisée pour résoudre des problèmes de partitionnement de graphe. Elle consiste à créer rapidement une partition d'un graphe  $G(S, A)$  de grande taille en regroupant les sommets ensemble, elle se décompose en trois phases : la contraction, le partitionnement et l'affinage[23].

```

Procedure Multilevel ( $G, k$ )
     $G_1 = G$ 
     $i = 1$ 
    Repeat
         $i \leftarrow i + 1$  ;
         $G_i = \text{contraction}(G_{i-1})$ 
    Until contraction faible ou  $G_{i+1}$  suffisamment petit
     $P_k = \text{partitionner}(G_i, k)$ 
    For  $j = i - 1$  à 1 do
         $P_k = \text{projection}(P_{k,j+1}, G_j)$ 
         $P_k = \text{affinage}(P_k)$ 
    End for
    Retourner  $P^1 k$ 
Fin Procédure

```

**Algorithme-1 multi-niveaux pour le partitionnement de graphes**

### 2.6.2.1 Contraction :

L'étape de contraction est une étape fondamentale de la méthode multi-niveaux, tel que les sommets sont regroupés pour former un graphe similaire, mais le nombre de sommets de nouveau graphe est plus petit que le précédent. Ainsi, pendant cette étape, une séquence des plus petits graphes  $\{G_1, G_2, \dots, G_m\}$  sont créées.

Si le regroupement des sommets est de mauvaise qualité, normalement la partition finale sera de mauvaise qualité aussi. Dans cette étape, un ensemble des sommets de graphe  $G_i$  est combiné pour former un sommet de graphe  $G_{i+1}$  à partir du graphe à l'entrée  $G_i$ , le résultat de cette étape est une famille de graphe  $\{G_1, G_2, \dots, G_m\}$ . Cette étape s'arrête si le nombre de sommets de graphe  $G$  est petit [24].

Parmi les méthodes de contraction de graphe nous avons trouvé :

- L'algorithme de Random Matching
- L'algorithme de Heavy Edge Matching (HEM)

La méthode HEM consiste à sélectionner aléatoirement un sommet parmi les sommets restant du graphe  $G_i$ , puis à sélectionner l'arête adjacente de poids maximal. Les deux sommets de cette arête forment un nouveau sommet de  $G_{i+1}$ . Ensuite, la méthode ajoute les arêtes adjacentes à ces deux sommets dans  $A_{i+1}$ . [24]

```

Procédure ContractionHEM( $G_i = (S_i, A_i)$ )
     $S \leftarrow S_i$ 
     $A_{i+1} \leftarrow A_i$ 
     $S_{i+1} \leftarrow S_i$ 
    tant que  $S \neq \emptyset$  faire
        Tirer au hasard un sommet  $s_1 \in S_i$ 
        Prendre l'arête  $(s_1, s_2) \in A_i$  de poids maximal
        Supprimer  $s_1$  et  $s_2$  de  $S$ 
        Ajouter le nouveau sommet  $s = \{s_1, s_2\}$  à  $S_{i+1}$ 
         $\text{poids}(s) = \text{poids}(s_1) + \text{poids}(s_2)$ 
        pour tout arête  $(s_1, u)$  dans  $A$  faire
            si  $(s_2, u) \in A$  alors
                 $\text{poids}(s_1, u) \leftarrow \text{poids}(s_1, u) + \text{poids}(s_2, u)$ 
                Supprimer  $(s_2, u)$  de  $A_{i+1}$ 
            fin si
        fin pour
    fin tant que
    retourner  $G_{i+1} = (S_{i+1}, A_{i+1})$ 
fin Procédure
    
```

Algorithme-2 de la phase de contraction

### 2.6.2.2 Partitionnement

La phase de partitionnement de la méthode multi-niveaux a pour but de trouver une partition du Graphe contracté. L'entrée de cette phase est la sortie de la phase précédente (graphe  $G_m$ ) et la partition trouvée sera la partition initiale de la phase suivante (phase d'affinage), Cette phase se réalise en appliquant des algorithmes comme :

- L'algorithme de Graph growing Algorithm (GGP).
- L'algorithme de Greedy Graph Growing(GGGP).

L'algorithme GGGP permet de résoudre le problème de partitionnement de graphe avec  $k = 2$  (bisection). Cette méthode consiste à créer de manière itérative un ensemble  $E$ . Cet ensemble est initialisé par un sommet de façon aléatoire. A chaque itération, un des sommets adjacents aux sommets de l'ensemble  $E$  est ajouté à  $E$ . Le sommet sélectionné est le sommet qui peut diminuer le plus le coût de coupe entre  $E$  et le reste de  $G$ . Autrement dit, c'est le sommet qui a le gain de la fonction objectif le plus grand. Le processus s'arrête si le poids de l'ensemble  $E$  est égal à la moitié du poids total des sommets du graphe. Cette méthode est très simple et la qualité de la partition obtenue dépend principalement du sommet de l'initialisation. Pour obtenir un meilleur résultat, nous pouvons lancer le programme avec des sommets différents lors de l'initialisation.[25]

**Procédure GGGP** ( $G = (S, A)$ )

*Prendre au hasard un sommet  $s_0 \in R$*

*$E \rightarrow \{s_0\}$*

*$F \rightarrow \{s \in R \text{ tel que } (s, s_0) \in A\}$*

*Calculer les gains de  $F$*

**While** *poids( $E$ ) <  $\frac{1}{2}$  poids( $S$ )* **do**

*Prendre le sommet  $s_i \in F$  de gain maximal*

*Déplacer  $s_i$  de  $F$  vers  $E$*

**For** arête (*s si*) dans *A* **do**

**If**  $s \in F$  **then**

Mettre à jour le gain de *s*

**Else** si  $s \notin E$  **then**

Ajouter *s* à *F*

Calculer le gain de *s*

**End if**

**End for**

**End while**

Retourner *E*

**Fin Procédure**

**Algorithme-3 la phase de partitionnement**

### 2.6.2.3 Affinage :

La phase d'affinage de la méthode multi-niveaux consiste à projeter sur le graphe initial la partition du graphe contracté trouvée pendant la phase de partitionnement. Concrètement, puisque la phase de contraction a construit une famille de graphes contractés  $\{G_1, \dots, G_n\}$ , la phase d'affinage va consister à récursivement projeter la partition d'un graphe contracté  $G_{i+1}$  sur le graphe initial  $G_i$ , puis à appliquer sur cette partition un algorithme d'affinage. Ainsi, une famille de partition sera créée  $\{p_k^1, \dots, p_k^n\}$ , dont chaque élément  $P_k^i$  correspond au partitionnement de  $P_k^{i+1}$  projeté sur  $G_i$ . L'itération se termine lorsque la partition  $P_k^i$  projetée sur le graphe initial  $G_i$  est affinée, et Parmi les méthodes d'affinage de graphe on trouve [26] :

- L'algorithme de khernighan-lin

L'algorithme de Kernighan-Lin n'est pas exactement une méthode de partitionnement, l'idée principale de cet algorithme est de trouver deux sous-ensembles de sommets de même taille, chacun dans une partie de la bissection, tels que leur échange diminue le coût de coupe de la bissection. A partir d'une bissection existante, l'algorithme échange successivement deux sous-ensembles de la bissection jusqu'à ce qu'on ne puisse plus diminuer le coût de la coupe. La dernière bissection obtenue est la bissection de coût de coupe minimale.

L'algorithme de Kernighan-Lin est d'autant plus rapide et performant si sa partition initiale est de bonne qualité. En effet, si la partition initiale est de bonne qualité, le nombre d'itérations de l'algorithme sera petit car il sera rapidement impossible de trouver des sous-ensembles à échanger. De plus, l'optimisation trouvée est locale. Ainsi, pour avoir un coût de coupe minimale, il est préférable de partir d'une bissection de coût de coupe faible. On peut utiliser la méthode de GGGP pour créer la bissection initiale [26].

```

Procédure KL( $G = (S, A), P_2$ )
    coût_coupe  $\leftarrow$  coupe( $P_2$ )
    répète « passe » % aussi appelé phase 1 optimization
        Calculer  $D_1$  et  $D_2$ 
        pour  $o = 1$  à  $n_S/2$  faire
            Choisir  $s_1^o$  et  $s_2^o$  qui maximisent le gain  $g$ 
             $E_1(o) \leftarrow s_1^o, E_2(o) \leftarrow s_2^o, G(o) \leftarrow g$ 
            Verrouiller  $s_1^o$  et  $s_2^o$ 
            Mettre à jour  $D_1$  et  $D_2$ 
        fin pour
        Choisir  $o$  qui maximise  $G$ 
        si  $G(o) > 0$  alors
            coût_coupe  $\leftarrow$  coût_coupe +  $G(o)$ 
            Échanger  $E_1(o)$  et  $E_2(o)$  dans  $P_2$ 
        fin si
    jusqu'à ce que  $G(o) > 0$ 
    retourner ( $P_2, \text{coût\_coupe}$ )
fin Procédure
    
```

**Algorithme-4 de la phase d'affinage**

## 1.7 Conclusion

Dans ce chapitre, nous avons présenté un parmi les outils de partitionnement : metis tel que metis est un progiciel permettant de partitionner de grands graphes irréguliers, et basé sur un partitionnement multi-niveaux tel que la méthode multi niveau la plus utilisé pour résoudre les problèmes de partitionnement, cette méthode dépend étroitement des performances des algorithmes d'affinage et de partitionnement qu'elle utilise, l'étude de ceux-ci est donc essentielle. Si la méthode multi-niveaux est depuis quelques années beaucoup plus utilisée que la méthode spectrale.

## Chapitre 3

# conception d'un modèle de représentation des données RDF ou OWL des données

### 3.1 Introduction

"Big Data" est un terme qui a été utilisé pour décrire la vaste croissance des données et la disponibilité des données qui ont eu lieu au cours des 20 dernières années. L'avènement d'Internet a entraîné une augmentation massive du volume de données créées et mises à disposition. Cela s'est encore accéléré lorsque le Web est devenu social.

L'un des plus grands obstacles au développement de l'analyse Big Data est la variété des informations sur le Web, dont certaines sont structurées, d'autres numériques, d'autres totalement non structurées (par exemple vidéo, audio). C'est là que le «Web sémantique» intervient. Ce dernier repose sur plusieurs briques technologiques, en particulier des identifiants fiables et pérennes (URI), le modèle (RDF) et le langage et protocole de requête SPARQL.

Dans notre modèle nous avons apporté à notre conception une application informatique dédiée au données du web sémantique. Notre principale objectif est de faciliter l'accès aux bases de données RDF afin d'atteindre un temps de réponse de plus en plus réduit pendant l'interrogation de ces données.

Dans ce chapitre nous allons décrire l'ensemble des étapes nécessaires pour l'interrogation d'une base de données RDF. Pour ce faire, nous proposons un modèle pour le traitement des données RDF/OWL. initialement, nous allons donner la structure générale de notre modèle à l'aide d'un diagramme contenant les différents composants de notre modèle, ensuite nous expliquerons le fonctionnement de chaque composant en détails et enfin nous allons terminer ce chapitre par une conclusion.

### **3.2 Modèle proposé pour le traitement de données RDF/OWL**

Le modèle qu'on propose envisage à tirer partie des avantages de l'utilisation du partitionnement pour le traitement de graphe RDF volumineux dans notre cas on utilise le API METIS, de sorte que le graphe en question soit partagé sur plusieurs nœuds de calcul pour être traité afin d'alléger la charge de travail et avoir des résultats dans un temps de réponse raisonnable.

Afin de montrer le fonctionnement de notre modèle, nous proposons dans le schéma ci-dessous une architecture générale de notre modèle. [27]

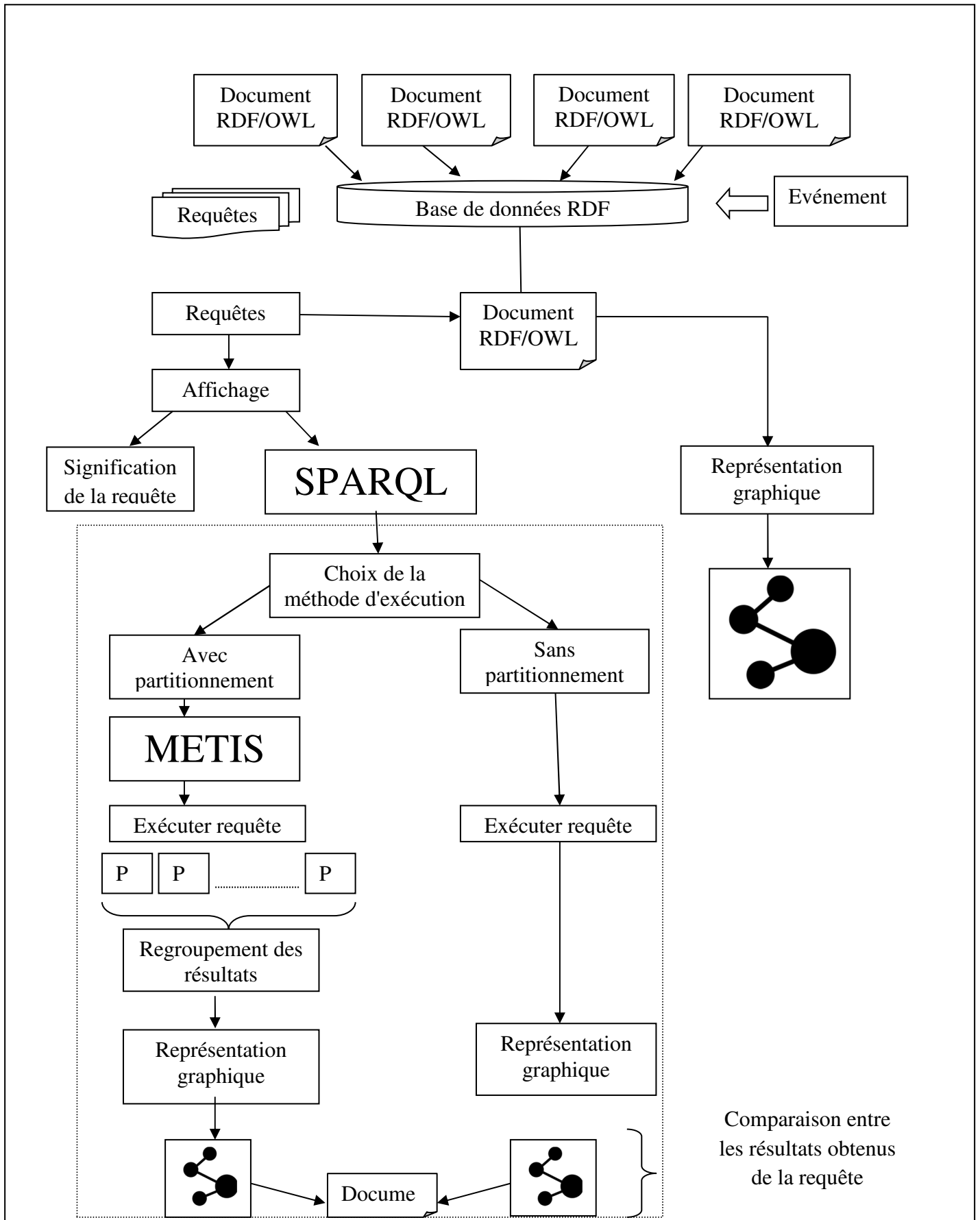


Figure -8 Architecture du modèle de traitement des données RDF / OWL

Le modèle reçoit en entrée des données qui parviennent de plusieurs source sous différentes formes. Ces dernières sont transformées en document RDF ou OWL pour mettre en évidence la sémantique des données.[28]

Ce modèle donne la possibilité de choisir un document RDF ou OWL parmi un ensemble de documents, ensuite, le document sera représenté sous forme graphique, Puis on introduit une requête qui sera exécutée sur ce dernier avec ou sans partitionnement et un résultat sera soumis sous une représentation graphique, et finalement une comparée sera effectué sur les deux graphes obtenus de l'exécution de la requête avec et sans partition, afin de déterminer laquelle de ces deux méthodes offre un temps d'exécution plus court, en visualise le résultat obtenu. [29]

### **3.2.1 Base de donnée RDF et OWL**

les données RDF peuvent provenir de différentes sources n'étant pas forcément stockée dans une mémoire commune. Ainsi, ce type de données ne suit pas généralement un modèle particulier de stockage, puisqu'elles sont distribuées . Pour cela, il existe plusieurs support de stockage dont la base de données RDF TripleStore. [29]

### **3.2.2 Document RDF ou OWL**

Notre modelé présente un événement Open pour ajouter 2 types de document différents, nous citons :

#### **3.2.2.1 RDF (Ressource Description Framework)**

Un document RDF est un fichier d'un modèle de description des données dans lequel toute ressource est identifiée par une URI, et où l'on peut faire des assertions ou déclarations sur ces ressources sous la forme d'un **triplet sujet/prédictat/objet**. [30]

#### **3.2.2.2 OWL (langage d'ontologies Web)**

Est un langage permettant de définir des ontologies Web structurées, et ce faisant, autorise une intégration plus riche et garantit l'interopérabilité des données au travers des frontières applicatives. [31]

OWL s'appuie sur un modèle et un schéma RDF pour ajouter plus de vocabulaire dans la description de propriétés et de classes, comme par exemple : les relations entre classes (i.e.

la séparation), la cardinalité (i.e. « un et un seul »), l'égalité, une typographie plus riche des propriétés, des caractéristiques de propriétés (i.e., la symétrie), et des classes énumératives.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:cd="http://www.recshop.fake/cd#">
  <rdf:Description rdf:about="http://www.recshop.fake/cd/Empire
  Burlesque">
    <cd:artist>Bob Dylan</cd:artist>
    <cd:country>USA</cd:country>
    <cd:company>Columbia</cd:company>
    <cd:price>10.90</cd:price>
    <cd:year>1985</cd:year>
  </rdf:Description>

  <rdf:Description rdf:about="http://www.recshop.fake/cd/Hide
  your heart">
    <cd:artist>Bonnie Tyler</cd:artist>
    <cd:country>UK</cd:country>
    <cd:company>CBS Records</cd:company>
    <cd:price>9.90</cd:price>
    <cd:year>1988</cd:year>
  </rdf:Description>
</rdf:RDF>
```

Figure -9 Représentation d'un document RDF avec la syntaxe XML

```
<owl:Class>
  <owl:intersectionOf rdf:parseType=" collection">
    <owl:Class rdf:about="#Person"/>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasChild"/>
      <owl:toClass>
        <owl:unionOf rdf:parseType=" collection">
          <owl:Class rdf:about="#Doctor"/>
          <owl:Restriction>
            <owl:onProperty rdf:resource="#hasChild"/>
            <owl:hasClass rdf:resource="#Doctor"/>
          </owl:Restriction>
        </owl:unionOf>
      </owl:toClass>
    </owl:Restriction>
  </owl:intersectionOf>
</owl:Class>
```

Figure -10 Représentation d'un document OWL avec la syntaxe XML

### 3.2.3 Représentation graphique des données RDF / OWL

Comment représenter au mieux la connaissance, pour pouvoir l'exploiter et en extraire d'autres connaissances ? Cette problématique est toujours un axe de recherche d'actualité.

Une des solutions à cette problématique est la représentation par les dessins ou les symboles.

Les graphes conceptuels sont des moyen efficace de représentation des connaissances. Dans le cas des document RDF et OWL ces dernier donne une vue simple qui permettent à la fois de bénéficier d'une recommandation par W3C, et d'avoir un mécanisme puissant d'inférence. plus précisément, un graphe conceptuel est un graphe biparti étiqueté, les deux classes sommets étant étiquetés respectivement par des nom de "concepts" et "relations conceptuelles".[31]

La représentation graphique permet a l'utilisateur de comprendre les document RDF ou OWL plus facilement qu'avec une représentation sous forme logique.

Nous considérons un exemple simple, la phrase:

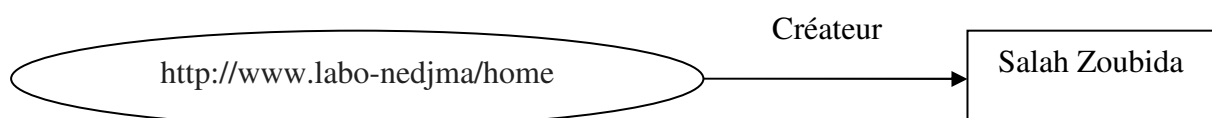
*Salah Zoubida et le créateur de la source <http://www.labo-nedjma/home>*

Cette phrase comprend les parties suivantes:

- Sujet (ressource): <http://www.labo-nedjma/home>
- Prédicat: Le créateur de la source
- Objet: "Salah Zoubida"

Dans ce document, nous allons schématiser une déclaration RDF en utilisant des graphes étiquetés dirigés (également appelés «diagrammes de nœuds et d'arcs»). Les nœuds (dessinés comme des ovales) représentent des ressources et les arcs représentent des propriétés nommées. Les nœuds représentant des littéraux de chaîne seront dessinés en tant que rectangles. La phrase ci-dessus serait donc schématisée comme suit:

Diagramme de nœud et d'arc simple:



**Figure -11 Représentation de Diagramme de nœud et d'arc simple**

Un exemple plus compliqué est le suivant notez que Le graphique RDF de la figure ci-dessous est écrit en RDF / XML comme suit:

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns="http://www.example.org/~joe/contact.rdf#">
  <foaf:Person rdf:about=
    "http://www.example.org/~joe/contact.rdf#joesmith">
    <foaf:mbox rdf:resource="mailto:joe.smith@example.org"/>
    <foaf:homepage
      rdf:resource="http://www.example.org/~joe/" />
    <foaf:family_name>Smith</foaf:family_name>
    <foaf:givenname>Joe</foaf:givenname>
  </foaf:Person>
</rdf:RDF>
```

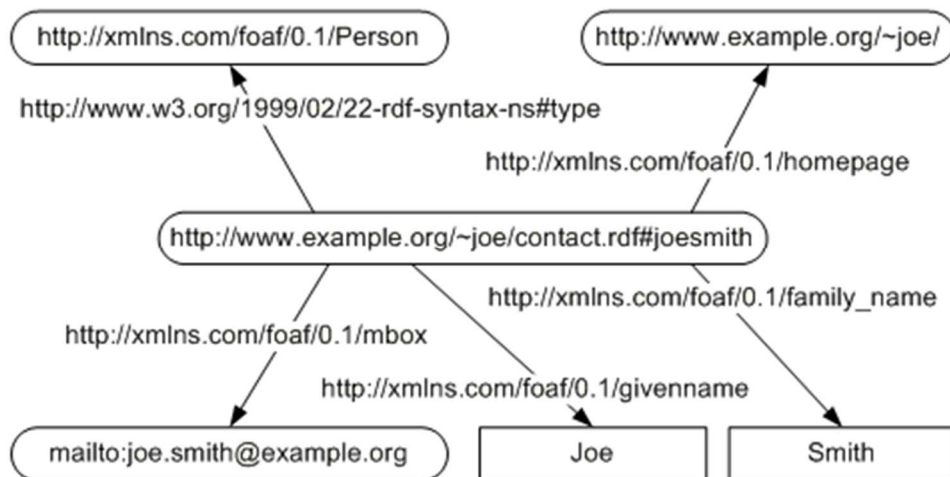


Figure -12 Représentation de Diagramme RDF décrivant Joe Smith

La figure ci-dessus représente l'équivalent de notre document RDF sous forme de graphe conceptuel.

### 3.2.4 La requête SPARQL

Le composant des requêtes SPARQL consiste à interroger la base de données RDF ou OWL. ou l'utilisateur entre une requête personnalisée ou bien génère une requête par défaut dans l'application sous le bouton nouveau. [16]

Exemple simple:

```

PREFIX      vcard: <http://www.w3.org/2001/vcard-rdf/3.0#>
SELECT      ?prenom
FROM        <http://pagesperso-systeme.lip6.fr/
            Mostat/inalco/XML/RDF/
            Exemples/vcardCat.rdf>
WHERE {
            ?x vcard:N ?c .
            ?c vcard:Given ?prenom .
}
    
```

Ce qui se lit :

Chercher dans le fichier "vcardCat.rdf" (ci-dessus) toutes les chaînes (ici désignées par "?prenom") telles qu'il existe dans le graphe correspondant deux triplets de la forme "?x vcard:N ?c" et "?c vcard:Given ?prenom", avec la même valeur pour "?c".

Résultat :

prénom: "Zoubida" "Nabil" "Sarah" "Reda".

### 3.2.5 Exécution de la requête sans partitionnement

Après le choix de document RDF ou OWL on introduits une requête SPRQL , la déclaration de cette dernier va nous générer un graphe, tel que le sujet, le prédicat et l'objet sont des variables illustrer . [32]

Nous constatons que l'application va nous généré une matrice d'incidence pour notre document RDF ou OWL, et sépare les données en deux type d'information objets et sujets. [33]

### 3.2.6 Exécution de la requête avec partitionnement

Un document RDF récupéré de la base de données RDF ne peut pas être directement partitionné, une transformation est nécessaire pour rendre notre graphe manipulable par le programme de partitionnement ( multiniveaux ). le fichier d'entrées générée, il est maintenant possible d'entamer le partitionnement de notre graphe en utilisant le programme de partitionnement de graphe, le résultat obtenu est une partition  $P=\{P1, \dots, Pn\}$ .

[34]

### 3.2.6.1 METIS

METIS nous permet de partitionner nos graphiques, et de partitionner des maillages d'éléments finis ainsi de produire des ordres de réduction de remplissage pour des matrices creuses. Les algorithmes implémentés dans METIS sont basés sur les schémas de partitionnement multiniveaux, de k-way multi-niveaux et de multi-contraintes. Un exemple simple: [35]

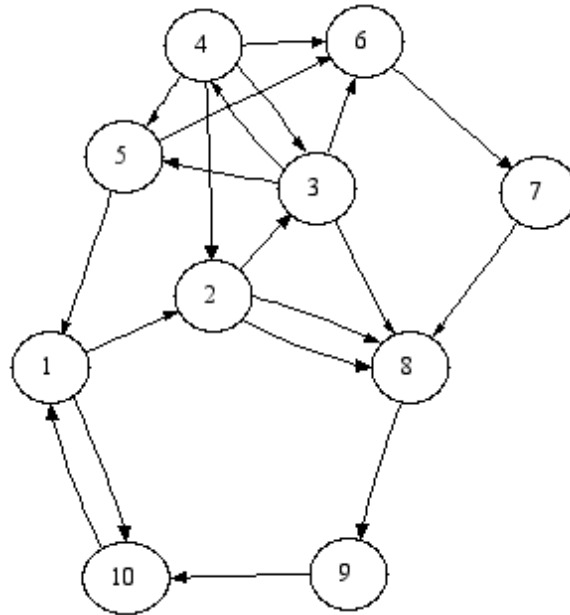


Figure -13 Graphe généré G (10, 20)

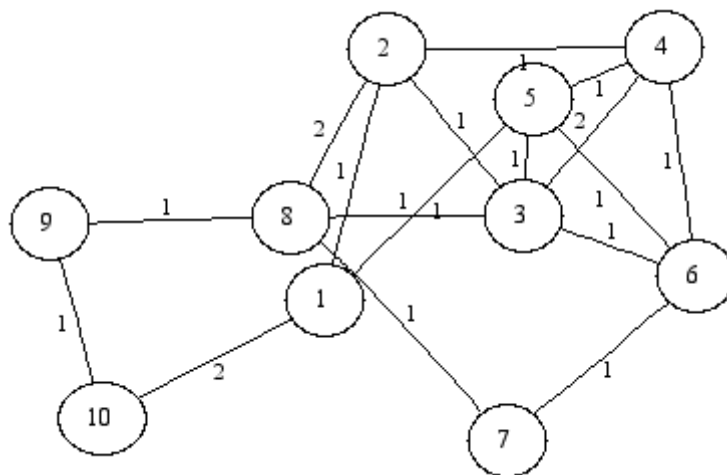


Figure -14 Graphe transformé à un format de bibliothèque METIS

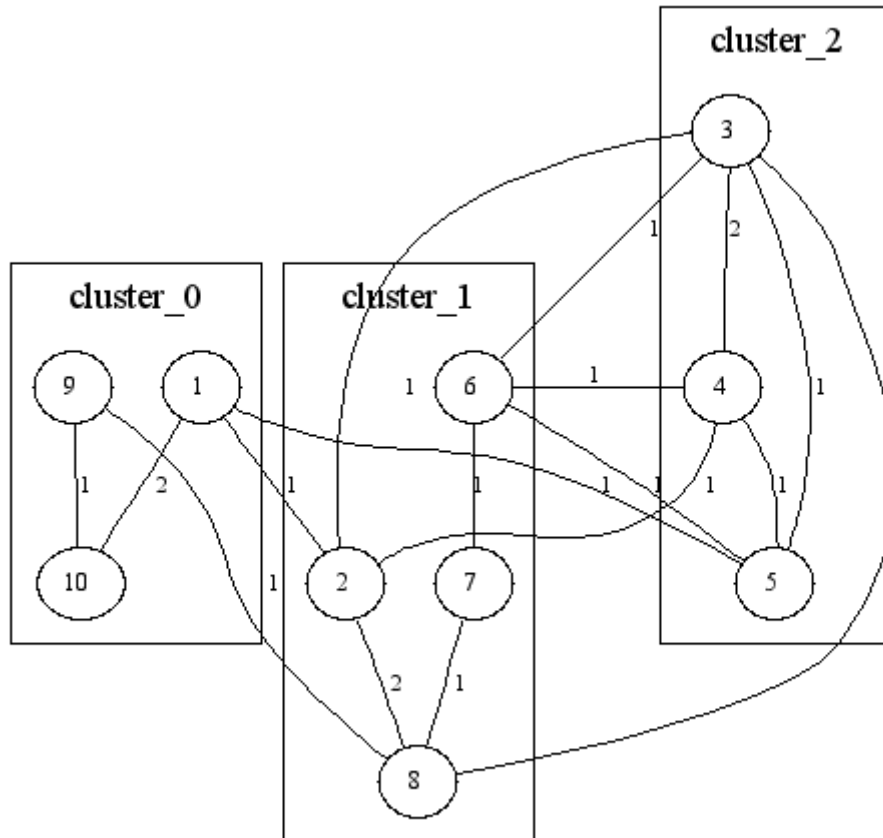


Figure -15 Graphe reçu, suite à la partition en trois parties par la bibliothèque METIS

### 3.3 Conclusion

Dans ce chapitre ont conclu que la représentation des données sur le web avec un modèle unique RDF ou OWL, permet de favoriser l'enrichissement et la réutilisation facile de l'information.

Il est maintenant possible de stocker un nombre grand de triplets dans une triple store, de plus la flexibilité et la simplicité de ce modèle facilite son interrogation avec le langage de requête SPRQL, QI et lui-même constitué de patron de graphe RDF même si SPRQL est un langage d'interrogation puissant sur les graphes mais il présente un énorme risque d'avoir un temps de réponse long pour cela on est dans l'obligation de diminuer le volume du graphe en sous-graphe, et là où se présente le METIS nous diminue l'espace de recherche pour les requêtes SPRQL.

Dans le prochain chapitre nous allons montrer les outils logiciels utilisés, pour l'implémentation du modèle.

# Chapitre 4

## implementation du système de traitement de donnees rdf/owl par graphe

### 4.1 Introduction

Dans ce chapitre nous allons présenter les différents outils de développement et les logiciels utilisés pour l'implémentation du modèle, ensuite, nous allons présenter notre application en indiquant en détail les étapes nécessaires pour son utilisation, et enfin on va terminer par une conclusion.

Pour l'implémentation de notre modèle, nous avons utilisé le langage java vue sa rapidité, sa simplicité, et sa portabilité sur un environnement Windows afin d'obtenir de bons résultats.

Notre application fournit un aperçu général sur les méthodes de partitionnement de graphe, elle permet d'illustrer ces avantages à l'aide de différents API (application programming interface) utilisées. Tel que L'API Jena permet la lecture et l'écriture RDF/XML, et l'API Jung permet la modélisation des arcs et les nœuds. Nous avons utilisé ainsi, l'API JenaJung pour récupérer les sujets, prédicat et l'objet d'un triplet RDF et l'API Jdom qui facilite la manipulation de document XML.

## 4.2 Environnement du travail

### 4.2.1 Ressources utilisées

Les ressources physiques exploitées :

- Processeur Intel® celeron ® 2957U@ 1.40GHz 1.40 GHz
- Mémoire vive d'une capacité de 2Go.
- système d'exploitation 32 bits.

Et comme ressource logicielle, nous avons utilisé :

- Système d'exploitation : Windows7
- Langage de programmation : java
- L'EDI : NetBeans de version 8.2

Le choix des systemes d'exploitation depend du type d'application que le nous voulons développer, par exemple puisque notre application est volumineuse et utilise des données de taille massive, il vaut mieux utiliser windows.

### 4.2.2 Windows :

Windows est un environnement d'exploitation créé par Microsoft qui fournit une interface, connue sous le nom Graphical User Interface (GUI), pour les ordinateurs. Windows élimine la nécessité de mémoriser les commandes de la ligne de commande (MS-DOS) en utilisant une souris pour naviguer dans les menus, les boîtes de dialogue, les boutons, les onglets et les icônes. Si vous utilisez un ordinateur PC (IBM), vous utilisez probablement une version de Windows. Si vous êtes sur un ordinateur Apple, vous utilisez MacOS.

Microsoft Windows a été introduit pour la première fois avec la version 1.0 le 10 novembre 1983. Depuis sa sortie, il y a eu plus d'une douzaine de versions de Windows. La version la plus récente de Windows pour les utilisateurs finaux est Windows 7 [36].

### 4.2.3 Langage de programmation:

Langage codé utilisé par les programmeurs pour écrire des instructions qu'un ordinateur peut comprendre pour faire ce que le programmeur (ou l'utilisateur de l'ordinateur) veut. Le langage informatique le plus bas (appelé bas niveau) est le langage machine qui utilise le code binaire ('1' et '0') qu'un ordinateur peut exécuter (exécuter) très rapidement sans utiliser de traducteur ou d'interprète, mais est fastidieux et complexe. Les langages de haut niveau (tels que Basic, C, Java) sont beaucoup plus simples (plus «English-like»), mais doivent utiliser un autre programme (un compilateur ou un interprète) pour convertir le code de haut niveau en machine Code, et sont donc plus lents.

Il existe des dizaines de langages de programmation et de nouveaux développements continus. Également appelé langage informatique. Pour l'implémentation de notre modèle nous avons utilisé le Java [37].

#### 4.2.3.1 Java

Nous avons utilisé comme langage de programmation le langage objet « java » Java est un langage de programmation de haut niveau développé par Sun Microsystems. Le langage de programmation Java a été développé par une petite équipe d'ingénieurs, connue sous le nom de Green Team, qui a lancé la langue en 1991. La langue a été appelée à l'origine OAK, et à l'époque conçue pour les appareils portables et décodeurs. Oak a échoué et en 1995, Sun a changé son nom en Java et a modifié la langue pour profiter du World Wide Web en plein essor.

Plus tard, en 2009, Oracle Corporation a acquis Sun Microsystems et a pris en charge deux actives clés du logiciel Sun: Java et Solaris.

Aujourd'hui, la plate-forme Java est une base couramment utilisée pour développer et diffuser du contenu sur le Web.

Le choix de java se justifie par les avantages suivants :

- ✓ C'est un langage bien connu et largement répandu. Il existe de nombreuses bibliothèques qui facilitent le développement des applications.

- ✓ Les applications java s'exécutent en utilisant une machine virtuelle, ce qui les rend indépendantes du système d'exploitation.
- ✓ Des machines virtuelles java ont été développées pour la plupart des systèmes actuels, ce qui facilite la portabilité des applications java.
- ✓ Les compilateurs java sont gratuits.
- ✓ Java permet de définir facilement des interfaces graphiques agréables à utiliser.

## 4.3 API utilisées

### 4.3.1 Jena

Jena est un programme API (interface de programmation d'application) pour les applications Java pour le web sémantiques [38]. Ainsi, Les graphiques sont représentés comme un «modèle» abstrait. Un modèle peut interagir avec les données des fichiers, bases de données, des URL ou une combinaison de ceux-ci. Un modèle peut également être interrogé par SPARQL. Nous traitons l'incohérence en utilisant le langage SPARQL d'où la nécessité de l'API Jena.

### 4.3.2 Jung

JUNG (Java Universal Network Graph) est une bibliothèque de logiciels open source qui fournit une langue commune et extensible pour la modélisation, l'analyse et la visualisation de données pouvant être représentées sous forme de graphique ou de réseau. Il est écrit en Java, qui permet aux applications basées sur JUNG d'utiliser les nombreuses fonctionnalités intégrées de l'API Java, ainsi que celles d'autres bibliothèques Java tierces existantes [39].

### 4.3.3 JenaJung

JenaJung est une API utilisée pour créer des graphes RDF avec l'API Jung en incorporant un modèle qui est la représentation d'un document RDF avec l'API Jena.

#### 4.3.4 JDOM

JDOM est une API open source Java dont le but est de représenter et manipuler un document XML de manière intuitive pour un développeur Java sans requérir une connaissance pointue de XML. Par exemple, JDOM utilise des classes plutôt que des interfaces. Ainsi pour créer un nouvel élément, il faut simplement instancier une classe [40].

#### 4.3.5 Metis

Metis est un ensemble de programme écrits en c utilisé pour pour partitionnement de graphe, le partitionnement des maillages [21]. Metis est utilisé ainsi pour la production de matrices de remplissage en réduisant l'ordonnancement pour les matrices creuses et la transformation de maillage en graphe, il a été développé dans le département d'informatique et d'ingénierie de l'université du minnesota.

Parmi les différents programmes implémentés pour le partitionnement de graphes dans Metis on utilisera « gpmetis ».

Gpmetis : est un programme qui partitionne un graphe en un ensemble nombre spécifique de parties [21] il reçoit en entrée un fichier avec une extension « graphe » et la sortie de ce programme est un fichier sous format « part.nparts » ou « nparts » avec le nombre obtenues de partition du graphe partitionné.

« gpmetis » offre deux manières pour partitionner un graphe soient avec la méthode récursif-bissection ou la méthode multi-niveaux k-way. Pour notre étude, nous avons opté pour la méthode multi-niveaux « K-way » qui est constituée de trois phases principales : la phase de contraction, la phase de partitionnement, et enfin la phase d'affinage.

A l'origine cette API fonctionne en ligne de commande à l'aide de cette commande :

```
Gpmetis [options] graphfile nparts
```

- **les paramètres**

« graphfile » : représente le fichier d'entrée, il contient le graphe qu'on veut partitionner.

« nparts » représente le nombre de parties qu'on désire avoir. Cela dépend généralement du nombre de machines de calcul.

- **Les options :**

Les options définissent les algorithmes que nous voulons utiliser durant les trois phases pour partitionner notre graphe.

Les options sont représentées dans le tableau suivant :

**Tableau -2 Explication des paramètres d'entrées de gpmets**

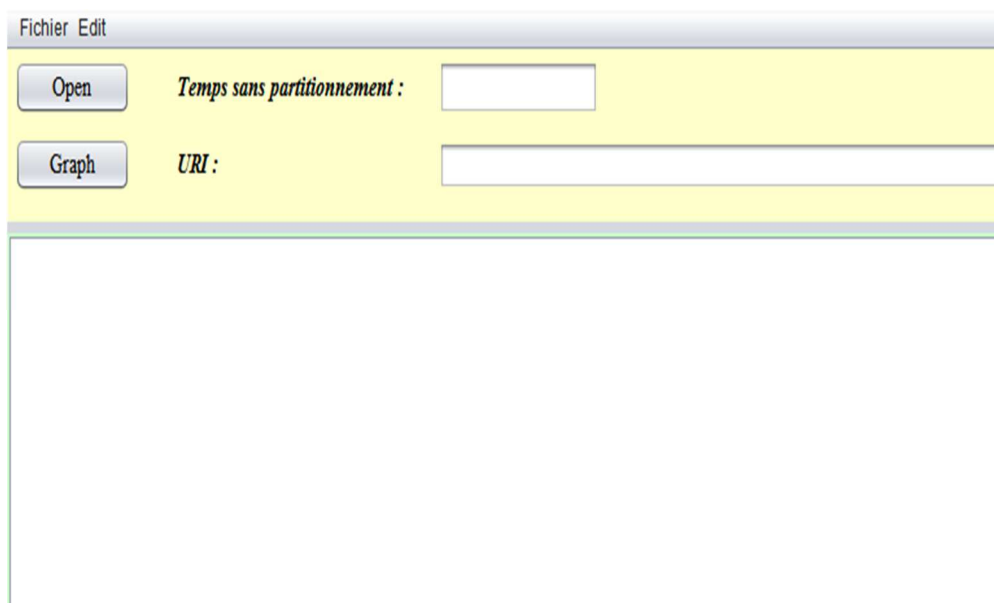
Options	Leurs rôles	Algorithmes
-ptype=string	Spécifie la méthode à utiliser pour le partitionnement du graphe	« rb » -> bisection récursive « Kway » -> partitionnement multi-niveaux K-way
Ctype=string	Spécifie le schema à utiliser pour regrouper les sommets dans la phases de contraction	« rm » -> random matching « shem »-> sorted heavy edge Matching
-niter=int	Indique le nombre d'itération pour les algorithmes de raffinement	
-ncuts=int	Spécifie le cout de coupe du graphe à partitionner dans la phase de partitionnement	

## 4.4 Développement de l'application :

Notre application définit un ensemble de fonctionnalités et d'opérations sur les documents rdf/owl afin de montrer les avantages d'une méthode de partitionnement de graphe. Pour se faire nous avons dû intégrer plusieurs API en respectant une certaine combinaison entre les versions de ces dernières, vu que ces API fonctionnent ensemble, le choix des versions est très important voici comment on a procédé :

Nous avons intégré la librairie D'apache jena 3.2.0, puis on a intégré jDom 2.0.6 jar, ensuite nous avons intégré la bibliothèque jung 2.0.1, et enfin on a extrait les deux classes de l'API JenaJung utilisées dans notre application. Ainsi, il fallait installer l'API Metis, pour le partitionnement du graphe.

La figure ci-dessous illustre la fenêtre principale de notre application.



**Figure –16** fenêtre principale de l'application

Comme nous pouvons le voir le menu principale de notre application contient 2 Menus (fichier,Edit) comme le présente la figure ci-dessous :



Ensuite, il est possible de charger le document voulu, et l'exécution de cette tâche s'effectue grâce à la classe « ConvertToXML »

```
public ConvertToXML(String path) throws JDOMException, IOException {
    this.texte = convertToXML(path);
}

protected static String convertToXML(String path) throws JDOMException, IOException {
    SAXBuilder builder = new SAXBuilder();
    Document document = null;
    try {
        //transformation d'un fichier xml en arbre
        document = builder.build(new File(path));
    } catch (IOException e) {
        e.printStackTrace();
    }
    XMLOutputter sortie = new XMLOutputter(Format.getPrettyFormat());
    String s = sortie.outputString(document);
    return s;
}
```

**Figure -19 la classe permettant d'afficher un document xml**

Dans notre cas, « saxBuilder builder » est une instance de la classe « ConvertToXML » et la variable texte et de type « string » contient le résultat de l'exécution qui est fourni dans le code ci-dessous :

```
protected static String convertToXML(String path) throws JDOMException, IOException {
    SAXBuilder builder = new SAXBuilder();
    File fichierXML = new File(path);
    Document document = null;
    try {
        //transformation d'un fichier xml en arbre
        document = builder.build(fichierXML);
    } catch (IOException e) {
        e.printStackTrace();
    }
    XMLOutputter sortie = new XMLOutputter(Format.getPrettyFormat());
    String s = sortie.outputString(document);
    return s;
}
```

**Figure -20 présentation de la méthode « saxBuilder builder »**

Dans le menu fichier nous allons trouver aussi le sous menu « graphe » qui peut représenter le graphe correspondant au document rdf/owl sélectionné.

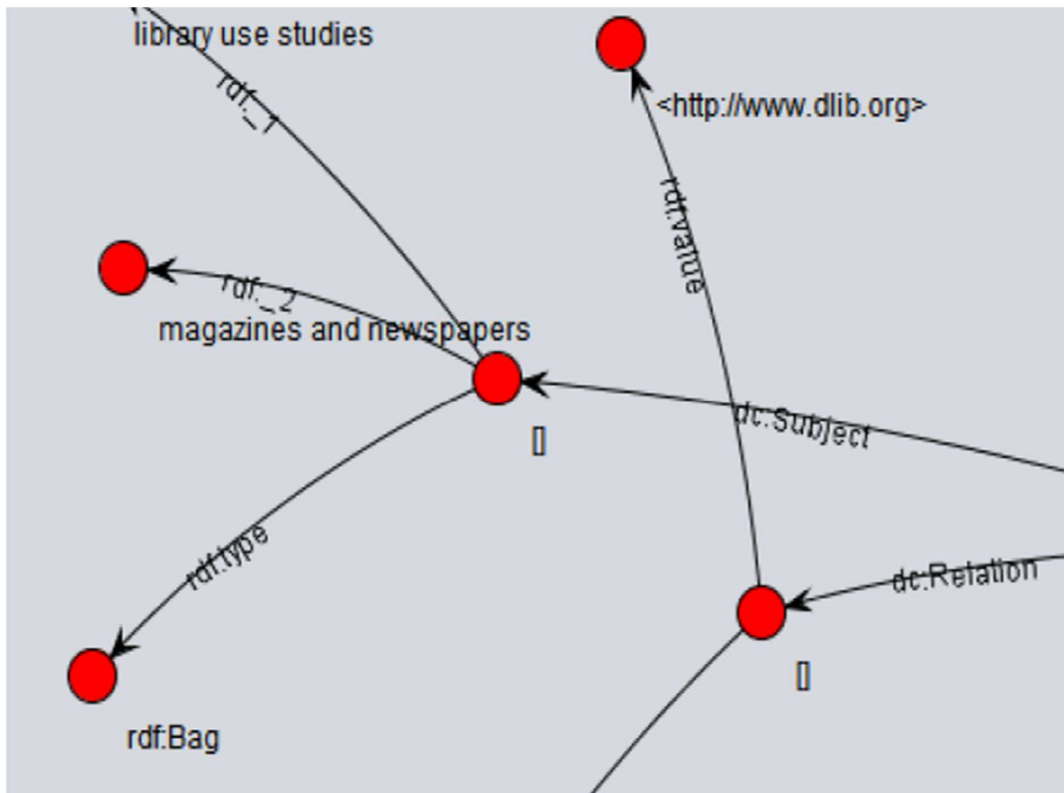
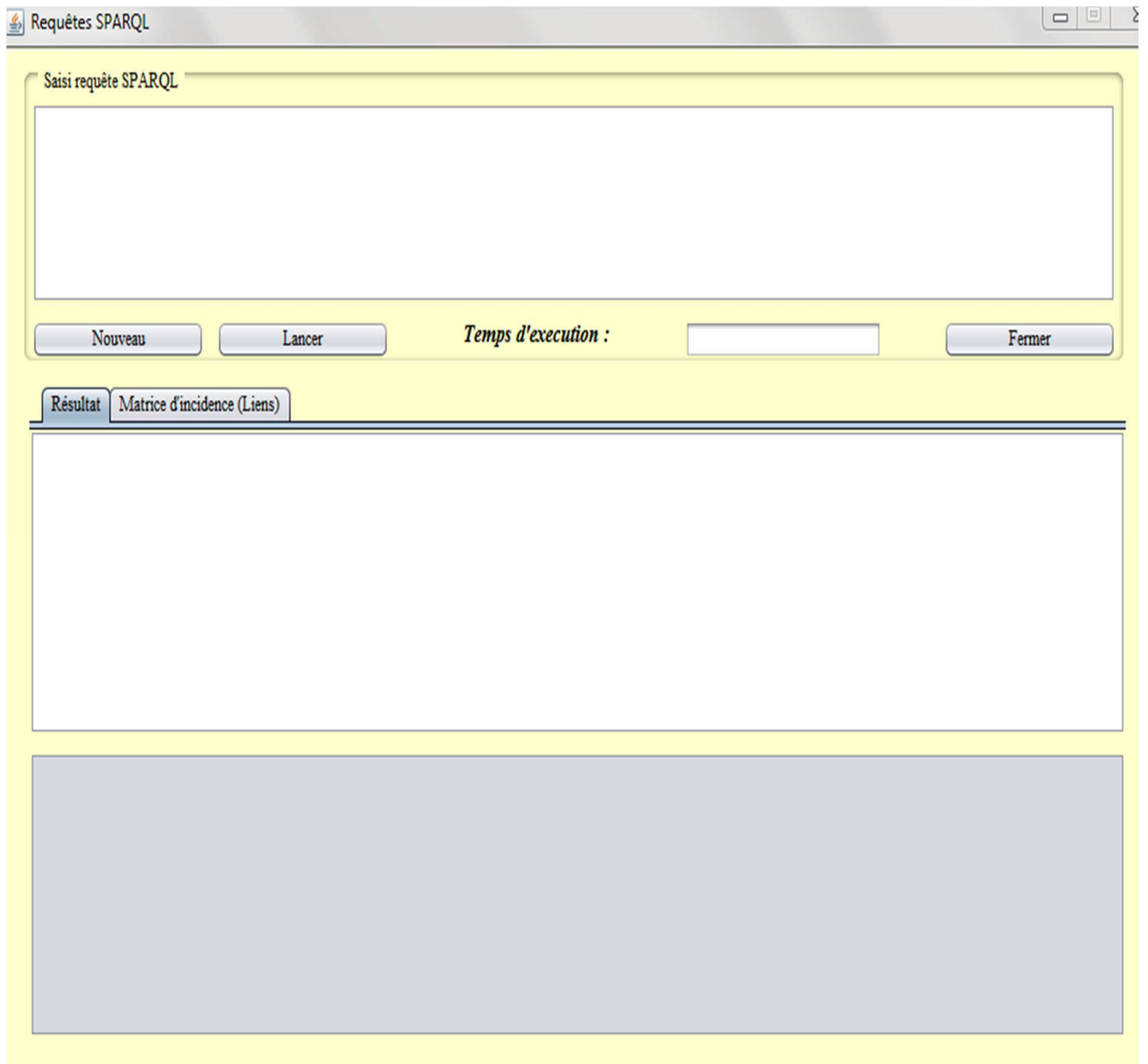


Figure -21 présentation graphique d'un document rdf/owl

#### 4.4.1.2 Menu Edit

Ce menu contient un seul sous menu « requête SPARQL » qui représente une fenêtre cette dernière contient toutes les fonctionnalités de notre application comme illustre la figure ci-dessous :



**Figure -22 la fenêtre de sous menu « requête sparql »**

Dans cette fenetre nous allons voir un emplacement pour saisir la requête sparql avec deux bouton, le premier « nouveau » qui nous aidons à afficher la requête par défaut, et le deuxième bouton « lancer » est pour exécuter notre requête sparql, après l'exécution de la requête nous allons voir une fenêtre affiche le graphe partitionné de la requête, comme nous pouvons voir le temps le temps de réponse moins important ; aussi un autre emplacement pour la matrice d'incidence et a la fin l'emplacement qui affiche les objets et les sujets de la requête comme illustre les figures suivantes :

```

Saisi requête SPARQL

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?subject ?object
WHERE { ?subject rdfs:subClassOf ?object }
    
```

**Figure -23** présentation de la requête sparql.

après l'exécution de la requête une matrice d'incidence qui provient comme resultat

Résultat	Matrice d'incidence (Liens)									
	ROLE	document	Infirmier	Medecin	Secrtaire	Fiche_de...	Certificat...	Dossier_...	Abdeloued	Ordonna...
ROLE	0	0	1	1	1	0	0	0	1	0
document	0	0	0	0	0	1	1	1	0	1
Infirmier	0	0	0	0	0	0	0	0	0	0
Medecin	0	0	0	0	0	0	0	0	0	0
Secrtaire	0	0	0	0	0	0	0	0	0	0
Fiche_de_r...	0	0	0	0	0	0	0	0	0	0
Certificat_...	0	0	0	0	0	0	0	0	0	0
Dossier m...	0	0	0	0	0	0	0	0	0	0

**Figure -24** présentation de la matrice d'incidence.

Et a la fin on reçoit un graphe finale qui explique le résultat de a matrice d'incidence :

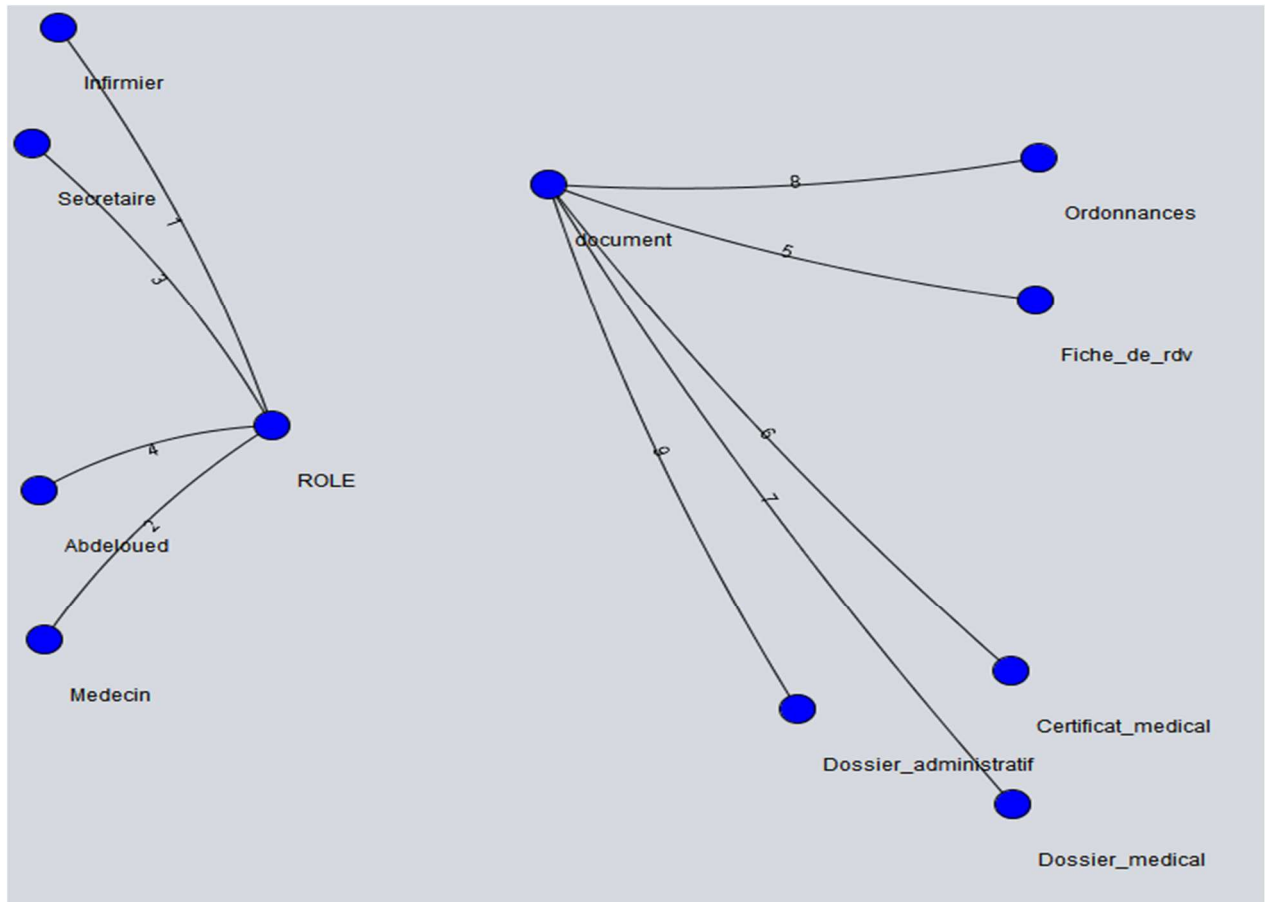


Figure -25 présentation de graphe final.

## 4.5 Conclusion

dans ce chapitre nous avons présenté l'implémentation de notre modèle, à l'aide des API spécialement dédiés à la représentation graphique des données sémantiques, comme l'API Jena qui permet de représenter un modèle rdf/owl de manière simple, l'API jung permet de représenter n'importe quel graphe et enfin l'API jenaJung offre la possibilité de représenter un modèle qui est jena sous forme de graphe jung, tous ces API permettant uniquement une visualisation de nos graphes mais pour le traitement de ces graphes metis est le meilleur propose des algorithmes dédiées à ce genre de tâche.



# Conclusion générale

Le web sémantique, propose une nouvelle plateforme permettant une gestion plus intelligente du contenu, à travers sa capacité de manipuler les ressources sur la base de leurs sémantiques. Le web sémantique favorise ainsi les coopérations Homme/Machine et permet de s'ouvrir à de nouvelles possibilités d'automatisation sur le Web.

Etant donné un grand graphe la recherche d'informations dessus n'est pas aussi simple, voire pénible, vu le nombre des sommets qui peut atteindre des milliards, de plus ces derniers, sont dispersées dans plusieurs serveur à travers le monde, né au moins si on utilise la méthode de partitionnement (la méthode de multi niveau) elle va en premier lieu contracter notre graphe sur une seule machine , puis le partitionner en plusieurs petits graphe de telle sorte que le nombre d'arêtes coupées est très réduit et le plus gros graphe obtenu contient un minimum de sommets, ensuite ces petits graphes obtenus seront agrandis en fonction du graphe initial, la recherche va prendre moins de temps et nous assure des resultats pertinents.

Le modele que nous avons proposé offre la possibilité de simuler, le déroulement de la méthode de partitionnement de graphe (la méthode multi-niveau) sur un graphe d'une taille massive.

Le but de notre travail est d'orienter l'évolution du Web pour permettre aux utilisateurs sans intermédiaires de trouver, partager et combiner l'information plus facilement. Les êtres humains sont capables d'utiliser le Web pour effectuer des, trouver un plan. Cependant, les machines ne peuvent pas accomplir toutes ces tâches sans direction humaine, parce que les pages web sont conçues pour être lues uniquement par des personnes et non par des machines. Aussi le Web sémantique est critiqué à cause de sa lourdeur : les langages utilisés pour le Web sémantique sont très verbeux, car dérivés du XML et donc souvent pénibles à utiliser. De ce fait, l'écriture d'ontologies est souvent très problématique, car elle exige une spécialisation dans un domaine particulier et lorsque l'on ne maîtrise pas ce domaine, elle devient très difficile à créer, mais notre motivation pour le domaine de traitement de données rdf ou owl se justifie par le fait que de nombreux applications et systèmes adoptent et utilisent la sémantique pour la représentation de leurs connaissances, ce qui engendre un nombre massif de données représentées suivant le modèle rdf ou owl .

# Bibliographie

- [1] BERNERS-LEE Tim, HENDLER James And LASILLA Ora, The Semantic Web, Scientific American, May 2001.
- [2] [Http://Dictionnaire.Reverso.Net/Francais-Definition/S%C3%A9mantique](http://Dictionnaire.Reverso.Net/Francais-Definition/S%C3%A9mantique). Consulté Le 05/12/2017
- [3] FENSEL Dieter, HENDLER James, LIEBERMAN Henry And WAHLSTER Wolfgang (Eds), Spinning The Semantic Web: Bringing The World Wide Web To Its Full Potential, The MIT Press, 2002.
- [4] HYVÖNEN Eero (Ed), Semantic Web Kick-Off In Finland, Vision, Technologies, Research And Applications, HIT Publications, 2002.
- [5] MAGKANARAKI Aimilia Et Al, Benchmarking RDF Schemas For The Semantic Web, In , P.132-146.
- [6] BROEKSTRA Jean, KAMPMAN Arjohn And VAN HARMELEN Frank, Sesame: A Generic Architecture For Storing And Querying RDF And RDF Schema, P.54-68.
- [7] R. Bendaoud. “Analyses formelle et relationnelle de concepts pour la construction d’ontologies de domaines à partir de ressources textuelles hétérogènes”. page 15, Juillet 2009.
- [8] W3C, Recommendation: Extensible Markup Language (XML) 1.0 (Fifth Edition), [Http://Www.W3.Org/TR/2008/REC-Xml-20081126/](http://Www.W3.Org/TR/2008/REC-Xml-20081126/), 26 Novembre 2008.
- [9] A. Bouramoul, Recherche D’information Contextuelle Et Sémantique Sur Le Web, Thèse Pour Obtenir Le Grade De Docteur En Sciences, Université MENTOURI De Constantine, 2011.
- [10] [Http://Fr.Slideshare.Net/Garlatti/Ontologies-2011-2012](http://Fr.Slideshare.Net/Garlatti/Ontologies-2011-2012) Consulté Le 10/02/2018
- [11] [Http://Optimiz.Me/Le-Fonctionnement-De-Google/](http://Optimiz.Me/Le-Fonctionnement-De-Google/) Consulté Le 10/02/2018
- [12] I. SAIDI, A, HAMSİ, Réalisation D’un Moteur De Recherche, Mémoire De Fin D’études

Pour L'obtention Du Diplôme De Licence En Informatique, Université Abou Bakr Belkaid–Tlemcen, 2013.

[13] Pnini O. Introduction A OWL, Un Language XML D'ontologies Web [Document]. Paris,2005.

[14] M. Gabbrielli Et S. Martini, Programming Languages: Principles And Paradigms, Springer, Consulté 2010.

[15] X. Zhang, L. Chen, Y. Tong, and M. Wang. EAGRE: Towards Scalable I/O Efficient SPARQL Query Evaluation on the Cloud. In ICDE, 2013.

[16] Etude Comparative Des Bases De Données Nosql : Mongodb, Couchbase, Cassandra, Hbase, Redis, Orient DB Réalisé Par : BENALLAL Zeyneb ET TAHRAOUI Hayet

[17] R. Bruchez, Les Bases De Données NOSQL Et Lebigdata Comprendre Et Mettre En Œuvre ,2ième Edition : ÉDITIONS EYROLLES , [Www.Editions-Eyrolles.Com](http://www.editions-eyrolles.com).

[18] Thès Présentée Pour Obtenir Le Titre De Docteur De L'institut National Polytechnique De Toulouse Prenté Par : Charles-Edmond Bicho.

[19] N. J. Belkin, P. Kantor, E. A. Fox, and J. A. Shaw. “Combining the evidence of multiple query representations for information retrieval”. In Information Processing and Management.,1995.

[20] Government Of Canada, Public Works And Government Services Canada, Translation Bureau, TERMIUM, [Http://Www.Btb.Termiumplus.Gc.Ca](http://www.btb.termiumplus.gc.ca), Consulté Le 8 Mars 2018.

[21] Contributions Au Partitionnement De Graphes Parallèles Multi-Niveaux Francois Pellegrin. Thèse Bordeaux : Laboratoire Bordelais De Recherche En Informatique ;2009.

[22] Graph Partitioning Algorithms For Distributing Workloads Of Parallel Computations Bradford L. Chamberlain October 13, 1998.

[23] Algorithms:[Http://Glaros.Edu/Gkhome/Vieus/Metis](http://Glaros.Edu/Gkhome/Vieus/Metis) METIS Family Of Multilevel Partitioning

[24] A Fast And High Quality Multilevel Scheme For Partitioning Irregular Graphs. S.L. : SIAM Journal Of Scientific Computing, 1998.

[25] G.Karypis, V.Kumar. Fast And High Quality Multilevel Scheme For Partitioning Irregular Graphs. S.L. : SIAM Journal Of Scientific Computing, 1998.

[26] C. WALSHAW, M. C ROSS : Mesh Partitioning : A Multilevel Balancing And Refinement Algorithm. SIAM Journal On Scientific Computing, 22:63–80, 2000.

- [27] C.Olivier, R, Dieng, C Faron-Zucker, F Gandon, A Giboin, le moteur de recherche sémantique Corese, (domaine), université de Nice.
- [28] F. Rastier Ontologies, Revue des sciences et technologies de l'information(2004), p.15.
- [29] B.mertcs, Specification of Coordination of Rule and Ontology Languages, Rapport technique, 2004.
- [30] H.Dorsaf, Le web sémantique et la recherche d'information, cours, IFT 6281.
- [31] M. Baziz. "Indexation conceptuelle guidée par ontologie pour la recherche d'information, Thèse de doctorat en informatique", Université Paul Sabatier de Toulouse, 2005.
- [32] J. Domingue, D. Fensel, A. Hendler, Handbook Of Semantic Web Technologie, Springer-Verlag Berlin Heidelberg, 2011.
- [33] Broekstra J., Kampman A. Et Harmelen F.V., Sesame: Ageneric Architecture For Storing And Querying Rdf And Rdf Schema, In Proceedings International Semantic Web Conference, Springer, 2002.
- [34] C. Walshaw : A Multilevel Algorithm For Force-Directed Graph Drawing. Journal Of Graph Algorithms And Applications, 2003.
- [35] B. W. KERNIGHAN , S. Lin : An Efficient Heuristic Procedure For Partitioning Graphs. Bell System Technical Journal, 49(2):291–307, 1970
- [36] <https://www.computerhope.com/jargon/w/windows.html>, consulté le 03 juin 2018.
- [37] [www.scoop.it/t/langages-de-programmation](http://www.scoop.it/t/langages-de-programmation) , consulté le 04 juin 2018.
- [38] <https://jena.apache.org>, consulté le 18 avril 2018.
- [39]. <http://jung.sourceforge.net/doc/manual.html#intro>, consulté le 18 avril 2018.
- [40]. <http://www.jdom.org/>, consulté le 19 avril 2018.