

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université de 8 Mai 1945-Guelma-  
Faculté des Mathématiques, d'Informatique et des Sciences de la matière  
Département d'Informatique



## Mémoire de fin d'études Master

Filière : Informatique

Option :

Science et technologie de l'information et de la communication

## Thème

---

Indexation des données de l'IoT sémantiques pour  
améliorer la prise de décision au profit des personnes  
handicapées.

---

Encadré Par :  
*Dr Halimi Khaled*

Présenté par :  
*Khebizi Hamed*

Membres du jury :  
*Dr Djakhdjakha Linda*  
*Dr Brahimi Said*

Juin 2022

# Remerciements

Avant tout, je remercie Dieu le tout puissant de nous avoir donné la force et la patience pour réaliser ce travail malgré les difficultés rencontrées.

Je tiens à remercier mon encadrant **Dr. HALIMI Khaled** ainsi que mon sous-encadrant **Mr. HADJADJI Halim** et le chef du département **Dr. KOUAHLA Zineddine** pour leurs disponibilités et leurs suivis qu'ils m'ont partagé durant la réalisation de ce travail.

Mes vifs remerciements vont également aux membres de jury **Dr. DJAKHDJAKHA Linda** et **Dr. BRAHIMI Said** d'avoir bien accepté d'examiner le contenu du présent travail et d'en être la présidente et l'examineur.

Je ne manquerai pas d'adresser mes remerciements au corps professoral et administratif du **Département d'informatique**.

Je tiens à remercier ma chère soeur **Allele Imen** qui m'a aidé durant la réalisation de ce travail.

Je tiens à remercier profondément mes chers amis **ahmed amin badji** et **me-hammedia saber abd errahmen** qui ont été toujours là pour moi pour m'aider et m'encourager tout au long de mes études.

Mille mercis à toute personne qui a contribué de près ou de loin à l'accomplissement de ce travail.

# Résumé

L'utilisation de la technologie dans la vie quotidienne augmente chaque jour, ce qui la rend indispensable en raison des facilités qu'elle apporte aux personnes dans leur vie quotidienne, notamment aux personnes handicapées. Parmi ces technologies, l'Internet des objets (IoT) a reçu beaucoup d'attention ces dernières années en raison de son interconnectivité. Le besoin de sémantique vient des différentes sources et capteurs dont nous dépendons pour la collecte d'informations. Nous utilisons également les technologies du Web sémantique (Ontologies, SWRL, SPARQL) pour les combiner avec les données des bases de données afin d'assurer une meilleure interopérabilité et ainsi que rendre l'Internet des objets sémantique. C'est ce que nous appelons le Web sémantique des objets. La nécessité des techniques d'indexation vient de la taille volumineuse des données de l'IoT. Nous utilisons donc les techniques d'indexation arborescente afin de gérer et stocker les informations dans le bon endroit pour les récupérer dans un temps limité. A partir de cette combinaison, un système basé sur un modèle ontologique enrichi contenant des connaissances sur la personne cible (telles que : signes vitaux, type de type de handicap, leur objectifs, les obstacles que l'on peut trouver, etc.) est réalisé afin d'offrir une meilleure vie aux personnes handicapées ainsi que de leur donner un peu plus d'autonomie dans leurs mouvements.

**Mots-clés : l'internet des Objets, Web Sémantique, Ontologies, SWRL, SPARQL, Interopérabilité, Web Sémantique des Objets, idO Sémantiquen, Personnes handicapées, Indexation**

# Abstract

The use of technology in daily life is increasing every year, which makes it indispensable because of the facilities it provides to people in their daily lives, especially to people with disabilities. Among these technologies, the Internet of Things (IoT) has received a lot of attention in recent years because of its interconnectivity. The need for semantics comes from the various sources and sensors we depend on for information collection. We also use Semantic Web technologies (Ontologies, SWRL, SPARQL) to combine them with database data to ensure better interoperability and thus make the Internet of Things semantic. This is what we call the Semantic Web of Things. The need for indexing techniques comes from the large size of the IoT data. We therefore use tree indexing techniques to manage and store the information in the right place to retrieve it in a limited time. From this combination, a system based on an enriched ontological model containing knowledge about the target person (such as : vital signs, type of type of disability, their objectives, the obstacles that can be found, etc.) is realized in order to offer a better life to disabled people as well as to give them a little more autonomy in their movements.

**Keywords : Internet of Things, Semantic Web, Ontologies, SWRL, SPARQL, Interoperability, Semantic Web of Things, idO Semantics, People with disabilities, Indexing**

# Table des matières

|   |           |
|---|-----------|
| Liste des tableaux  | iv        |
| Table des figures   | v         |
| <b>1 Personnes handicapées, Internet des Objets et Web Sémantique</b> | <b>2</b>  |
| 1.1 Introduction  | 2         |
| 1.2 Personnes handicapées   | 2         |
| 1.2.1 Types d'handicapé   | 2         |
| 1.2.2 Problèmes communs chez les personnes handicapées                | 3         |
| 1.2.3 Solutions proposées   | 3         |
| 1.3 Internet des objets   | 3         |
| 1.3.1 Architecture de l'IdO   | 4         |
| 1.3.2 Caractéristiques de l'IdO                                       | 4         |
| 1.3.3 Défi de l'internet des objets                                   | 4         |
| 1.4 Web Sémantique  | 5         |
| 1.4.1 Architecture du Web Sémantique                                  | 6         |
| 1.4.2 Les ontologies  | 7         |
| 1.4.3 Technologies utilisées dans le web sémantique                   | 8         |
| 1.4.4 Web Sémantique et domaine de la santé                           | 8         |
| 1.4.5 De l'internet des objets vers le Web sémantique des objets      | 9         |
| 1.4.6 De l'Internet des objets au Web des objets                      | 9         |
| 1.4.7 Web sémantique des objets                                       | 9         |
| 1.4.8 Caractéristiques de la combinaison des WS avec l'IoT            | 9         |
| 1.4.9 Les enjeux du SWoT  | 10        |
| 1.5 Conclusion  | 10        |
| <b>2 Techniques d'indexation</b>                                      | <b>11</b> |
| 2.1 Introduction  | 11        |
| 2.2 Exigences en matière d'indexation des données volumineuses        | 11        |
| 2.3 Les techniques d'indexation actuelles                             | 12        |
| 2.3.1 Techniques d'indexation multidimensionnelle                     | 13        |
| 2.3.2 Techniques basées sur les arbres                                | 15        |
| 2.3.3 Techniques basées sur le bitmap                                 | 16        |
| 2.3.4 Techniques d'indexation métrique                                | 17        |
| 2.4 Conclusion  | 22        |

|          |   |           |
|----------|---|-----------|
| <b>3</b> | <b>Conception</b>   | <b>23</b> |
| 3.1      | Introduction . . . . .  | 23        |
| 3.2      | Objectif de l'approche proposée : "Web Sémantique des Objets" . . . . . | 23        |
| 3.3      | L'architecture du système proposé . . . . .                             | 23        |
| 3.4      | Présentation des connaissances . . . . .                                | 24        |
| 3.4.1    | Concepts et classes . . . . .   | 25        |
| 3.4.2    | Relations entre les objets . . . . .                                    | 28        |
| 3.4.3    | Proportions littérales . . . . .  | 31        |
| 3.5      | La structure d'indexation proposée . . . . .                            | 32        |
| 3.6      | Scénario d'application . . . . .  | 34        |
| 3.7      | Conclusion . . . . .  | 36        |
| <b>4</b> | <b>Implémentation</b>   | <b>37</b> |
| 4.1      | Introduction . . . . .  | 37        |
| 4.2      | Environnement . . . . .   | 37        |
| 4.3      | Outils de développement . . . . .                                       | 37        |
| 4.3.1    | Protege . . . . .   | 37        |
| 4.3.2    | Visual Studio Code . . . . .  | 37        |
| 4.4      | Langage de Programmation . . . . .                                      | 38        |
| 4.4.1    | Python . . . . .  | 38        |
| 4.4.2    | Bibliothèques . . . . .   | 38        |
| 4.5      | Data set . . . . .  | 38        |
| 4.6      | Développement du système . . . . .                                      | 38        |
| 4.6.1    | Importation de bibliothèques . . . . .                                  | 38        |
| 4.7      | Le modèle ontologique . . . . .   | 39        |
| 4.7.1    | Importation de l'ontologie . . . . .                                    | 39        |
| 4.7.2    | Initiation de l'ontologie . . . . .                                     | 39        |
| 4.7.3    | Génération de fichiers RDF pour l'instance handicap . . . . .           | 39        |
| 4.8      | Présentation du système . . . . .                                       | 41        |
| 4.8.1    | Lecture des informations . . . . .                                      | 41        |
| 4.8.2    | Création des individus . . . . .  | 41        |
| 4.8.3    | Indexation des données . . . . .  | 43        |
| 4.9      | Conclusion . . . . .  | 44        |
|          | <b>Bibliographie</b>  | <b>49</b> |

# Liste des tableaux

|     |  |    |
|-----|--|----|
| 2.1 | Tableau comparatif des approches multidimensionnelles. . . . . | 20 |
| 2.2 | Tableau comparatif des approches métriques. . . . .            | 21 |
| 3.1 | Tableau descriptif des concepts . . . . .                      | 28 |
| 3.2 | Tableau des relations entre les objets . . . . .               | 31 |
| 3.3 | Tableau des proportions littérales . . . . .                   | 32 |

# Table des figures

|      |   |    |
|------|---|----|
| 1.1  | types des handicaps [Rey, 2022]   | 3  |
| 1.2  | Caractéristiques de l'IdO.[?]   | 5  |
| 1.3  | Architecture du WS.[?]  | 6  |
| 1.4  | graphe de l'ontologie   | 7  |
| 2.1  | Exigences d'indexation des données massives                             | 11 |
| 2.2  | Taxonomie globale des techniques d'indexation                           | 13 |
| 2.3  | Arbre-kD[Rey, 2022]   | 16 |
| 2.4  | Taxonomie des techniques d'indexation métrique[Joyce, 2020a]            | 19 |
| 2.5  | Taxonomie des techniques d'indexation multidimensionnelle[Joyce, 2020a] | 19 |
| 3.1  | Enchaînement de la proposition  | 24 |
| 3.2  | Concepts et classes   | 25 |
| 3.3  | Concepts et classes   | 26 |
| 3.4  | arbre SB  | 33 |
| 3.5  | Schéma descriptif du scénario de l'application                          | 35 |
| 4.1  | L'importation de bibliothèques  | 39 |
| 4.2  | Importation de l'ontologie  | 39 |
| 4.3  | initialisation des concepts   | 40 |
| 4.4  | initialisation des concepts   | 40 |
| 4.5  | initialisation des relations  | 40 |
| 4.6  | initialisation des propriétés des données                               | 40 |
| 4.7  | Exemple RDF généré  | 41 |
| 4.8  | Lecture des données collectées  | 42 |
| 4.9  | Ajouter les individus a l'ontologie                                     | 42 |
| 4.10 | Ajouter les individus a l'ontologie                                     | 42 |
| 4.11 | Ajouter les individus a l'ontologie                                     | 43 |
| 4.12 | Ajouter les individus a l'ontologie                                     | 43 |
| 4.13 | Ajout des relations   | 43 |
| 4.14 | Recherche sémantique  | 43 |
| 4.15 | Recherche par métriques   | 44 |
| 4.16 | Recherche knn   | 44 |



# Introduction Général

L'internet des objets est une nouvelle technologie apparue au cours des dernières années. La caractéristique fondamentale de l'internet des objets est qu'il transforme les objets en dispositifs intelligents. Il peut générer et envoyer des données sur le réseau et il se connecte les uns aux autres sans intervention humaine. Tout cela contribue à améliorer la vie et à encourager plusieurs domaines commencent à intégrer ces technologies. La demande pour l'internet des objets conduit à une augmentation du nombre d'objets connectés, selon les estimations de la recherche Statistique, les prévisions indiquent que d'ici 2025, plus de 150 milliards d'objets connectés à l'internet seront utilisés comme la principale fonction de ces objets sont de générer les données, la quantité de données a considérablement augmenté [22, n.d.]. Cette expansion a créé le besoin de stocker, gérer et sécuriser d'énormes volumes de données afin de tirer le meilleur partie de cette masse d'informations. La massivité des données ayant des défis techniques majeurs qui conduit un coût élevé, longtemps ainsi que n'arrive pas la recherche d'information efficace. Pour surmonter ces défis, le traitement et le stockage de cette quantité massive de données demande une solution efficace pour permettre une recherche et une découverte rapide et efficace de ces données et services. Cette solution s'appelle l'indexation. Elle est représentée par un processus d'organisation des données de manière qui permet un accès efficace à celles-ci. L'indexation permet de réduire l'espace de recherche, les coûts d'entrée/sortie et le nombre de calculs de distance entre les termes pour fournir un accès rapide aux informations contenues dans une base de données. Comme les données de l'internet des objets proviennent de différentes sources, les applications qui utilisent ces données souffrent d'un manque d'interopérabilité. Le web sémantique est une solution adéquate pour le traitement de ce problème. Mais L'indexation simple ne permet pas de trouver les liens sémantiques entre les termes. Dans ce projet, nous avons comme objectif de proposer une nouvelle structure d'indexation des données basée sur l'intégration de la technologie du web sémantique afin d'enrichir ces données, de faire ressortir de nouvelles connaissances et de fournir une recherche efficace. La solution proposée est appliquée dans le domaine de la santé des personnes handicapées afin d'éliminer les difficultés qu'elles rencontrent dans leur vie.

# Chapitre 1

## Personnes handicapées, Internet des Objets et Web Sémantique

### 1.1 Introduction

Actuellement, La technologie est devenue l'un des éléments les plus importants qui facilitent la vie quotidienne de l'homme, y compris celle des personnes handicapées. Dans ce contexte, nous parlerons d'eux, de leurs types et des problèmes auxquels ils sont confrontés au vie quotidien. En outre, dans ce chapitre, nous allons utiliser les technologies du web sémantique des objets dans le même contexte. Ce concept combine l'Internet des objets et le Web sémantique. Nous allons parler de leurs architectures et de leurs défis.

### 1.2 Personnes handicapées

Selon la définition de l'Organisation mondiale de la santé (OMS), une personne handicapée est toute personne dont l'intégrité physique ou mentale est altérée de façon temporaire ou permanente, soit en raison de l'âge ou d'un accident, de telle sorte que son autonomie et sa capacité à accomplir les tâches quotidiennes des personnes normales sont compromises[unk, n.d.].

#### 1.2.1 Types d'handicapé

Différents types de handicap existent comme montre la figure 1.1 suivante :

Les différents types de personnes handicapées sont énumérés ci-dessous :

1. Handicap moteur.
2. Handicap visuel.
3. Handicap auditif.
4. Handicap psychique.
5. Déficience intellectuelle.
6. Maladies invalidantes.

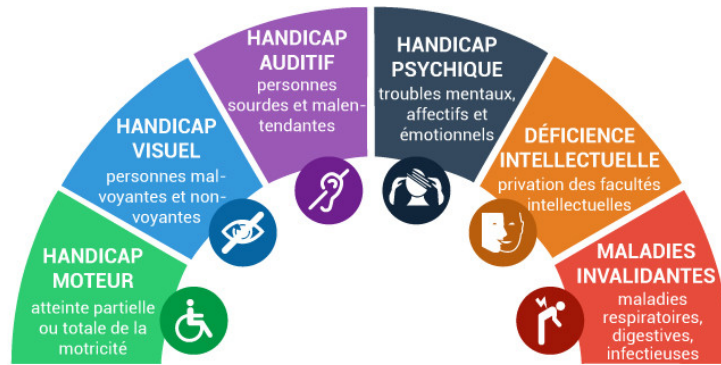


FIGURE 1.1 – types des handicaps [Rey, 2022]

### 1.2.2 Problèmes communs chez les personnes handicapées

- La difficulté de réalisation des tâches de la vie quotidienne telles que l’habillement, l’hygiène, l’alimentation, les activités à la maison.
- La vie sociale, comme la communication avec les autres.
- La relocalisation et le déménagement.
- Les études à l’école ou à l’université ainsi que la recherche d’un emploi.
- Le manque de respect et de considération, la sous-estimation et l’agressivité dont sont victimes les personnes handicapées de la part d’autres personnes non handicapées.
- La difficulté de surveiller son état de santé

### 1.2.3 Solutions proposées

Avec l’évolution de la science et de la technologie, les problèmes des personnes handicapées peuvent être résolus, du moins partiellement. Dans la prochaine section, nous allons discuter de deux disciplines que nous allons utiliser pour résoudre ces problèmes et rendre la vie de ces personnes plus facile. Ces deux disciplines sont le Web sémantique et l’Internet des objets.

## 1.3 Internet des objets

L’Internet des objets est fondamentalement un nouveau domaine, ce qui rend difficile l’utilisation d’une définition standardisée. On a donc choisi la définition qui nous conviendrait le mieux. L’action de coordination et de soutien pour les activités et la normalisation liées à la RFID au niveau mondial : "Une infrastructure de réseau mondiale reliant des objets physiques et virtuels grâce à l’exploitation de capacités de saisie et de communication de données. Cette infrastructure comprend les développements existants et en cours de l’Internet et des réseaux. Elle offrira des capacités spécifiques d’identification d’objet, de capteur et de connexion comme base pour le développement de services et d’applications coopératifs indépendants. Ceux-ci seront caractérisés par un degré élevé de saisie autonome de données, de transfert d’événements, de connectivité de réseau et d’interopérabilité. [Rose *et al.*, 2015]

### 1.3.1 Architecture de l'IdO

- **Couche des dispositifs** : la couche des dispositifs IdO comprend tous les dispositifs intelligents qui sont connectés au système. Ces dispositifs sont des matériaux qui sont dotés de capteurs, des processeurs, des actionneurs et la capacité de transmettre des données sur les réseaux. [Joyce, 2020a]
- **Couche passerelle** : la couche passerelle est l'intergiciel entre la couche dispositif IoT et la couche de la plateforme IoT. Cette couche contient un dispositif physique ou un logiciel qui recueillent les données des dispositifs intelligents et les transfèrent vers le cloud. [Joyce, 2020a]
- **Couche plate-forme** : après avoir été transmises au nuage, les données peuvent être traitées par des outils et des applications de la couche plate-forme IoT. Celle-ci comprend également des logiciels qui visualisent les données traitées des capteurs sur les dispositifs destinés aux utilisateurs.

[Joyce, 2020a]

### 1.3.2 Caractéristiques de l'IdO

- **Connectivité** : Grâce à l'infrastructure de l'IoT, tout peut être connecté et intercommuniquer.
- **Objets** : Tout ce qui peut être connecté comme les téléphones, les ordinateurs, les caméras, et les drones. Les appareils peuvent contenir des capteurs ou des matériaux de détection qui peuvent être attachés aux appareils et objets.
- **Données** : Les données sont le principal composant de l'Internet des objets, le premier pas vers l'action et l'intelligence.
- **Communication** : les objets sont connectés, ce qui leur permet de communiquer et de partager des données. Ces données peuvent être analysées. La communication peut être établie sur différents types de réseaux : LAN, MAN et WAN.
- **Intelligence** : l'aspect de l'intelligence comme dans les capacités de détection des dispositifs IoT et l'intelligence recueillie par l'analyse et le traitement des données volumineuses.
- **Action** : la conséquence de l'intelligence. Elle peut être effectuée manuellement ou automatisée par un logiciel implémenté dans les appareils.
- **Écosystème** : La place de l'internet des objets dans la perspective d'autres technologies, communautés et objectifs et du tableau dans lequel l'internet des objets s'inscrit.

La dimension de l'internet des objets, la dimension de la plateforme et le besoin de partenariats solides.

### 1.3.3 Défi de l'internet des objets

Aujourd'hui, l'Internet des objets est confronté à de nombreux défis, mais le défi le plus important qui nous préoccupe est l'interopérabilité. Elle peut être définie comme la capacité de faire communiquer deux ou plusieurs systèmes/composants en échangeant des données entre eux et en utilisant des outils intégrés ou non. Ainsi qu'en utilisant des informations. C'est pourquoi nous faisons appel au web sémantique dans ce chapitre.[Vermesan *et al.*, 2014]

## DEFINING IoT: 7 CHARACTERISTICS

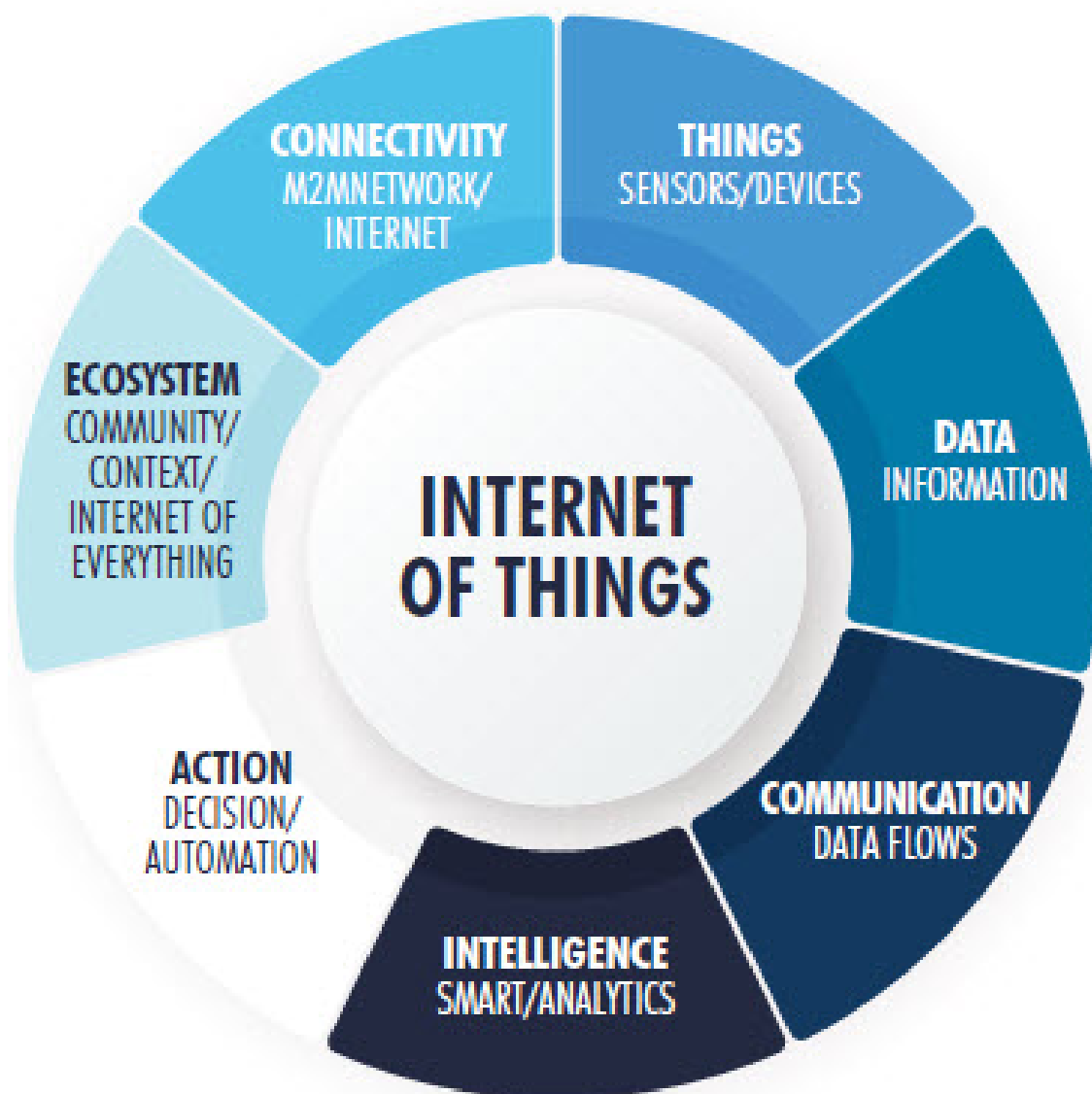


FIGURE 1.2 – Caractéristiques de l’IdO.[?]

### 1.4 Web Sémantique

La vision du Web sémantique est celle d’un Web dans lequel les ressources sont accessibles non seulement aux humains, mais également aux processus automatisés comme celui d’un agent automatisé errant le Web, accomplissant des tâches utiles telles que la recherche améliorée (en termes de précision), la découverte de ressource le filtrage d’information, . . .etc., ce qui transformera le web en un vaste espace d’échanges de ressources entre machines, permettant l’exploitation de grands volumes d’informations et de services variés. L’automatisation des tâches nécessite d’élever le statut du Web lisible par machine à quelque chose que nous pouvons appeler compréhensible par machine. L’idée principale est d’avoir des données sur le Web définies et liées de telle manière que leur signification soit interprétable par des humains et des machines.[Berners-Lee *et al.*, 2001]

### 1.4.1 Architecture du Web Sémantique

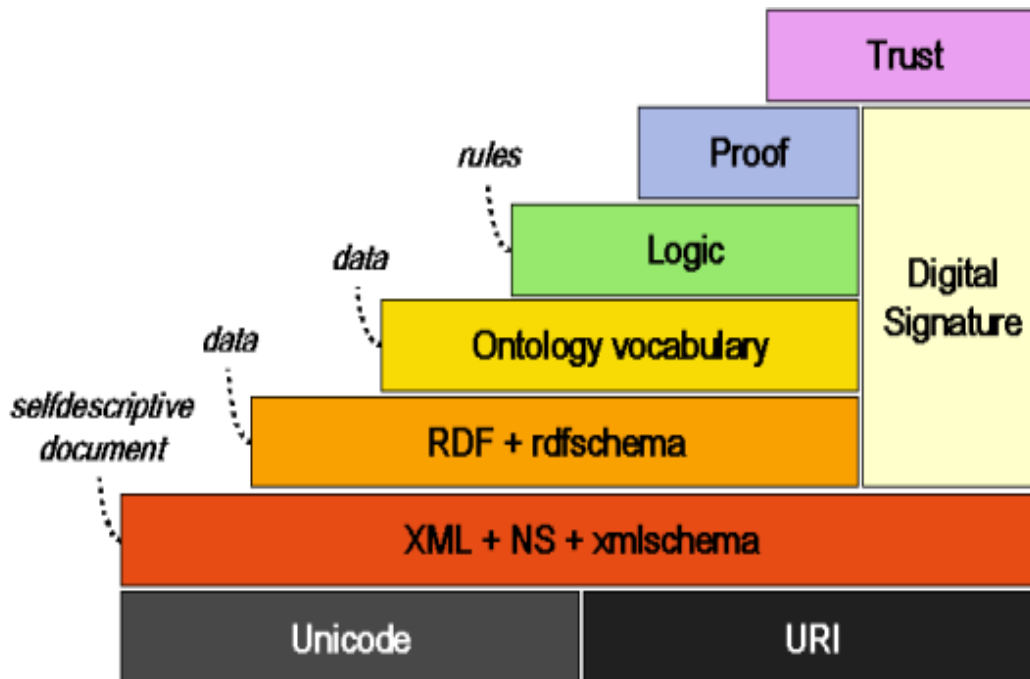


FIGURE 1.3 – Architecture du WS.[?]

1. **Unicode et URI** : Les couches Unicode et URI (également appelés localisateurs de ressources uniformes) sont les premières couches de l'architecture du Web sémantique ; elles imposent l'utilisation de caractères internationaux, elles identifient les ressources sur le Web : documents, images, fichiers téléchargeables et services.
2. **XML, XML NS, and XML Schema** : La couche constituée de XML, de l'espace de noms XML et du schéma XML permet une représentation uniforme de la structure des documents. XML est un langage de balisage créé pour nous fournir des généralités sur Internet grâce à son format simple, lisible par l'homme et la machine. En outre, l'espace de noms XML est utilisé pour nous fournir une méthode permettant d'éviter les conflits de noms d'éléments et permet la combinaison de documents ayant un vocabulaire hétérogène. Enfin, le schéma XML décrit la structure d'un document XML.
3. **RDF et RDFS** : La couche suivante est RDF + RDFS, en utilisant ces technologies, il est possible de lier les ressources web avec un vocabulaire prédéfini. La ressource Description Framework (RDF) est un graphe orienté contenant un modèle de données simple en trois parties (sujet, prédicat, objet) pour faire référence à des objets ou à des ressources et à la manière dont ils sont liés les uns aux autres.
4. **Langage d'ontologie Web (OWL)** : OWL est utilisé dans la couche ontologique pour représenter de manière explicite la signification de termes dans les vocabulaires et les relations entre ces termes. Cette représentation des termes et de leurs interrelations est appelée une ontologie. OWL offre plus de possibilités pour exprimer la signification que XML, RDF et RDF-S, et il va au-delà de ces langages dans sa capacité à représenter un contenu interprétable par une machine.
5. Couches de logique et de preuve : Enfin, les couches de logique et de preuve permettent la définition de règles formelles et le raisonnement basé sur ces règles.[?]

## 1.4.2 Les ontologies

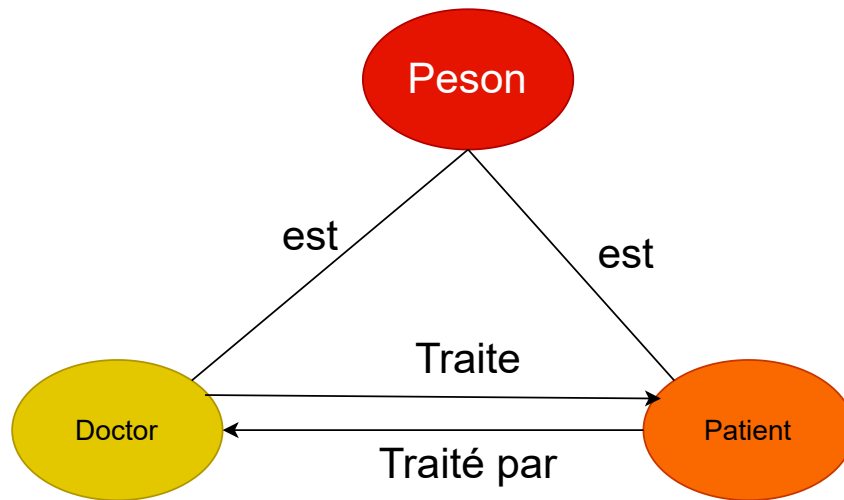


FIGURE 1.4 – graphe de l'ontologie

Le concept d'ontologie est issu de la philosophie et des œuvres de Platon. En informatique, une ontologie est "une description formelle de toutes les entités d'un domaine et des relations existant entre ces entités".

Cette définition peut paraître compliquée ! Il s'agit en fait de décrire des connaissances de telle manière qu'elle puisse être exploitée par une machine, et avec un souci de complétude et d'"universalité". Les ontologies font partie de l'intelligence artificielle dite "symbolique", qui consiste à structurer les connaissances pour les exploiter et les rendre accessibles à un ordinateur, par opposition aux machines, par opposition à l'apprentissage automatique (comme les réseaux de neurones, l'apprentissage profond, etc.).[Jean-Baptiste, 2021]

### Les composants d'une ontologies :

Les aspects que nous pouvons utiliser pour représenter quelque chose en utilisant l'ontologie

1. **Concept** : Les concepts peuvent être abstraits ou concrets, élémentaires ou composites, réels ou fictifs, en bref, un concept peut être n'importe quoi. classes (XOL, RDF(S), OIL, DAML+OIL), objets (OML) ou catégories (SHOE). Par exemple : Personne, Ville, Capteur ... etc.
2. **Instances** : Les instances représentent les éléments liés à un concept spécifique. Ahmed, par exemple, est une instance du concept de personne, Guelma est une instance du concept de ville, etc. Ville ... etc.
3. **Relations** : Les relations sont une interaction entre les concepts du domaine et les attributs. En fait, les attributs peuvent être utilisés pour définir des relations binaires dans XOL, RDF(S) et DAML+OIL. Les relations sont appelées relations dans SHOE et OML, et rôles dans OIL.
4. **Attributs** : une caractéristique inhérente à un concept qui permet de mieux définir ce concept. Chaque attribut a une valeur qui diffère d'une instance à l'autre. Par exemple : le nom du concept Personne a pour valeur KHEBIZI.
5. **Axiomes** : Les axiomes modélisent les connaissances considérées comme vrais dans le domaine traité.

[Lord, 2010]

### 1.4.3 Technologies utilisées dans le web sémantique

1. **RDF** : Le Web sémantique utilise RDF comme format de stockage de l'information. Il est par nature ouvert, ce qui signifie que l'on peut toujours ajouter de nouvelles données ou relations. Un autre avantage de l'utilisation de RDF est qu'il supporte les données non structurées, les documents, sites Web, articles et autres.[Prasad *et al.*, 2021]
2. **OWL** :  
Est un langage conçu pour représenter des connaissances complexes sur les choses et les relations entre elles. Ce langage offre la caractéristique descriptive du web sémantique, ce qui signifie que les ontologies sont indépendantes des données qu'elles décrivent, alors que dans les bases de données classiques, les données décrites sont déterminées directement.[Prasad *et al.*, 2021]
3. **SPARQL** :  
Alors que RDF et OWL permettent de stocker et de définir des connaissances et des relations, il est nécessaire de disposer d'une base de données et des relations, il est nécessaire de disposer d'un langage d'interrogation pour en tirer parti. SPARQL est un protocole et un langage d'interrogation RDF [Prudhommeaux et Seaborne, 2008], on peut le voir comme un SQL pour le Web sémantique. Il utilise le principe d'identification des chemins dans un graphe [West, 2000] pour retrouver les résultats d'une requête donnée. L'un des avantages de SPARQL est que les requêtes peuvent être distribuées à plusieurs points de terminaison SPARQL (services qui acceptent les requêtes SPARQL et renvoient les résultats). [Domingue *et al.*, 2011]
4. **SWRL** : Est un langage de la couche logique de l'architecture SW qui améliore la sémantique d'une ontologie. la couche logique dans l'architecture SW qui améliore la sémantique d'une ontologie définie en OWL. Il permet de manipuler des instances par des variables ( ?x, ?y, ?z). Cependant, SWRL ne permet pas de créer des concepts ni des relations, il permet simplement d'ajouter des relations en fonction des valeurs des variables et de la satisfaction de la règle. Ses règles sont construites selon le schéma suivant : antécédent -> conséquent.[Mehla & Jain, 2019]

### 1.4.4 Web Sémantique et domaine de la santé

Plusieurs domaines sont touchés par les SW nous concentrons par la santé et ses domaines.

De nos jours, où que vous alliez, les services de santé sont indispensables. Pour fournir ces services de manière optimale, l'échange d'informations est une nécessité particulièrement dans les sciences médicales, car la plupart des informations médicales disponibles doivent être partagées entre des systèmes informatiques disparates. disponibles ont besoin d'un moyen de partager des systèmes informatiques disparates.L'interopérabilité basée sur l'ontologie a été conçue pour le partage des connaissances et l'échange d'informations à la fois entre les personnes et entre les services/applications, à l'aide de plusieurs dispositifs connectés de différents fabricants qui utilisent des normes différentes qui seront traités sémantiquement à l'aide du web sémantique. [Karami & Rahimi, 2019]



### 1.4.5 De l'internet des objets vers le Web sémantique des objets

Les recherches récentes sur le web sémantique ont montré la puissance de la sémantique pour gérer les données structurées, semi-structurées et non structurées. Ce qui nous permet de bénéficier des Big Data collectées à partir de diverses sources de données (dispositifs IoT) en essayant d'obtenir de meilleurs résultats. Cette combinaison en est encore aux premiers stades de la recherche, ce qui constitue un défi qui contribuera à la science dans différents domaines tels que le domaine des soins de santé, le domaine commercial et financier. Dans les prochaines sous-sections, nous parlerons de la façon dont nous avons résolu le problème de l'interopérabilité lié à l'IdO et de la façon dont nous pouvons en tirer parti pour aider les utilisateurs, et comment nous pouvons en tirer profit pour aider les personnes handicapées en passant de l'Internet des objets au Web des objets et enfin au Web sémantique des objets.[Gacitua *et al.*, 2019]

### 1.4.6 De l'Internet des objets au Web des objets

Comme de plus en plus d'appareils intelligents sont connectés à l'Internet, la meilleure étape à prendre est d'utiliser le World Wide Web et ses technologies et protocoles pour rendre ces appareils interopérables, comme nous l'avons dit dans la sous-section précédente. Il y a quelques années, Kindberg et al ont proposé de relier des objets physiques à des pages Web en utilisant des interfaces infrarouges ou des codes-barres sur les objets. Les utilisateurs pouvaient trouver l'URI de la page correspondante simplement en interagissant avec l'objet.[Kindberg *et al.*, 2002]

### 1.4.7 Web sémantique des objets

Tandis que les paries précédentes visent les bénéfices de l'intégration du web des objets, cette section approfondit l'intégration de la sémantique au WoT. Elle explique la manière d'interpréter sémantiquement les données produites par les dispositifs IoT en utilisant la sémantique du W3C. Qu'est-ce qu'un web sémantique des objets ?

#### Définition

Le Web sémantique des objets (SWoT) est une sorte d'extension du W3 (World Wide Web) qui tente de résoudre les problèmes causés par les systèmes et les informations hétérogènes et les problèmes de sécurité. l'hétérogénéité des systèmes et des informations et de fournir une meilleure compréhension des différents domaines de l'IdO. Elle fournit une meilleure compréhension des différents domaines de l'IdO. Elle peut également être décrite comme une combinaison du Web sémantique (WS) et de l'internet des objets (IoT). Cette approche permet l'existence des abstractions qui sont appropriées pour faire correspondre les capteurs avec leurs données brutes à des entités du monde réel avec une sémantique réelle.[Chatzimichail *et al.*, 2021]

### 1.4.8 Caractéristiques de la combinaison des WS avec l'IoT

- l'utilisation généralisée des URI et du protocole HTTP.
- Permettre l'interopérabilité entre des données provenant de différentes sources grâce à l'annotation du contenu.
- Utilisation de langages standard communs.

- Combinaison des connaissances du domaine et des informations de base avec les données des capteurs, ajoutant plus de significations aux machines.[Chatzimichail *et al.*, 2021]

### 1.4.9 Les enjeux du SWoT

Défis communs du Web sémantique des objets rencontrés par les chercheurs [Chatzimichail *et al.*, 2021] :

- L'évolutivité et la croissance des systèmes IoT avec de multiples dispositifs.
- Le manque de développeurs intéressés par ce domaine.
- Le manque d'ensembles de données sémantiques.
- La précision du raisonnement peut ne pas être optimale.

## 1.5 Conclusion

A la fin de ce chapitre, nous avons présenté les concepts les plus liés aux handicaps, aux besoins de santé et aux technologies fournies pour aider les personnes à devenir plus autonomes. Nous avons besoins de santé et des technologies mises à disposition pour les aider à devenir plus autonomes et à bénéficier d'une vie quotidienne plus facile. Les technologies basées sur l'IdO qui sont utilisées pour recueillir les données médicales des patients, les traiter et les échanger avec d'autres. Le chapitre explique également les défis de l'IdO, notamment l'hétérogénéité des données et la sémantique comme une solution pour y faire face. Dans le chapitre suivant, nous présenterons d'autres concepts qui seront appliqués dans ce travail pour optimiser en fonction du temps d'accès aux informations.

# Chapitre 2

## Techniques d'indexation

### 2.1 Introduction

Selon les études de nos jours, le nombre de nœuds de capteurs et d'actionneurs dans les réseaux de l'Internet des Things (IoT) a augmenté, générant une grande quantité de données. La plupart des techniques de recherche sont basées sur la division des données cibles en sous-ensembles.[Benrazek *et al.*, 2020] L'utilisation des objets de l'IdO sera augmentée jour après jour et dépassera 1 trillion d'objets connectés en 2030. Cette croissance exponentielle des objets provoque notamment l'émergence de données massives. Malheureusement, la quantité massive de données générées par ces objets et leurs caractéristiques distinctives rendent l'utilisation des systèmes traditionnels de gestion de bases de données inadaptée à la gestion en temps réel de ces volumes de données, qui sont générés en continuité et variables dans le temps. [Ferrag *et al.*, 2019] Dans notre projet on va utiliser les techniques d'indexation afin d'ignorer les informations inutiles ou bien redondantes dans le but de réduire le temps d'extraction d'informations pertinentes.

### 2.2 Exigences en matière d'indexation des données volumineuses

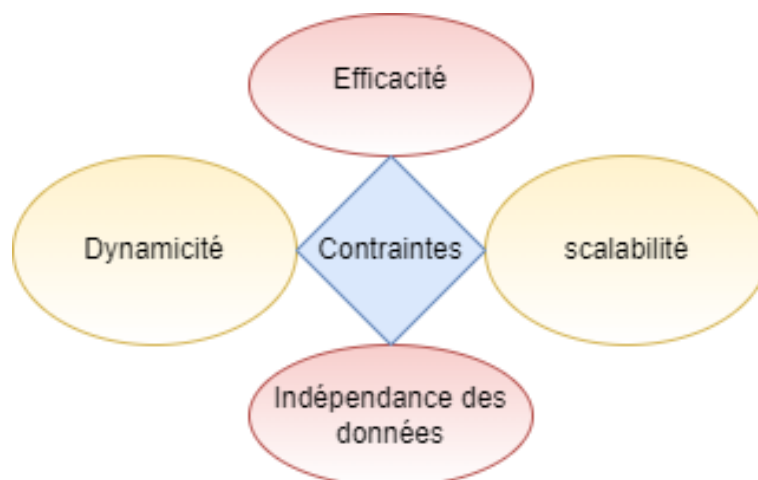


FIGURE 2.1 – Exigences d'indexation des données massives

Pour indexer des grandes données dont la taille est assez importante dans le domaine

de l'IoT, des études récentes sur les techniques d'indexation ont montré comment optimiser les performances de la recherche dans de grands ensembles de données avec une plus grande efficacité. Cette section décrit certains des besoins d'indexation qui sont plus importants que les données traditionnelles. Sur la base de ces contraintes, chaque technique d'indexation sera analysée pour déterminer son applicabilité à grande échelle. Les méthodes d'indexation existantes sont décrites et les ensembles de données utilisés pour évaluer leur efficacité sont également discutés dans cette section. En plus des difficultés déjà soulevées par la généralisation des espaces multidimensionnels aux espaces métriques, des contraintes, parfois techniques, doivent être imposées au préalable aux solutions possibles. Il s'agit donc de proposer une technique d'indexation, sachant que toute solution doit respecter certaines contraintes. Comme le montre la Fig. 2.1 :

- **Evolutivité** : La structure de l'index doit être capable de indexer des volumes de données toujours plus importants sans perte de performance (efficacité dans la réponse aux requêtes).
- **Efficacité** : Contrairement aux structures classiques d'accès physique aux E/S physiques classiques, qui essaient "seulement" de réduire le nombre d'entrées/sorties sur le disque, un index de recherche de similarité doit aussi tenir compte des coûts de calcul, car les calculs de similarité entre deux objets peuvent être très coûteux, voire plus coûteux que les temps de lecture du disque.
- **Dynamisme** : La structure de l'index doit être dynamique par rapport aux modifications de la base de données, c'est-à-dire ne pas nécessiter de réorganisations périodiques coûteuses mais assurer une indexation continue.
- **Indépendance des données** : Les structures d'accès doivent fournir de bonnes performances pour toute distribution possible de données, c'est-à-dire qu'elles traitent efficacement les problèmes liés à la "malédiction des dimensions" ou inefficacité dans les grandes dimensions. Le problème de la malédiction de la dimension, principal élément qui affecte l'efficacité de l'algorithme d'indexation, lorsque la dimension est augmentée. Les approches existantes se sont avérées peu fiables, il devient difficile d'indexer de grands volumes de données, de les gérer et de les analyser. Ce problème est causé par les déficiences inhérentes au partitionnement de l'espace et aussi par le facteur de chevauchement entre les régions. Cette question reste donc ouverte à la discussion.[Ferrag *et al.*, 2019]

## 2.3 Les techniques d'indexation actuelles

Plusieurs techniques d'indexation ont été introduites pour répondre aux problèmes d'indexation de grandes données. Cet article fournit une étude comparative des performances des techniques d'indexation récentes et de leur capacité à résoudre les problèmes d'indexation de grandes données. leur capacité à résoudre les problèmes d'indexation de données volumineuses. En outre, ces techniques sont examinées selon une taxonomie proposée.[BENRAZEK, 2021]

Cette exigence découle de la nécessité d'indexer et de rechercher des données de plus en plus nombreuses et complexes. Les petites collections de données, qui contiennent également des objets simples, peuvent être facilement, mais la gestion de grandes bases de données, comme la plupart des bases utilisées aujourd'hui, nécessite des techniques plus sophistiquées, surtout lorsqu'elles contiennent des types de données complexes. En effet, les objets à indexer sont souvent plus complexes que de simples vecteurs (homogènes -

ex : espaces vectoriels - ou hétérogènes - ex : tuplets dans une base de données relationnelle) . Par exemple, dans une base de données multimédia, il peut y avoir un mélange de descriptions "sémantiques" (telles que de simples mots-clés mais jusqu'aux graphiques de description conceptuelle associés à une ontologie) et de contenus médiatiques (tels que des histogrammes de couleurs). Tous ces facteurs soulèvent des questions sur les méthodes d'indexation et de recherche. Par conséquent, le développement de l'indexation a été partiellement transféré aux espaces métriques, afin d'exploiter non pas la distribution des données en tant que telle, devenue trop riche et complexe, mais de travailler "uniquement" sur les similarités qui peuvent être calculées entre les objets. Intrinsèquement, les problèmes d'indexation dans espaces multidimensionnels se poursuivent dans une version généralisée, tandis que de nouveaux défis apparaissent en raison de l'absence d'informations sur les objets. [Ferrag *et al.*, 2019]

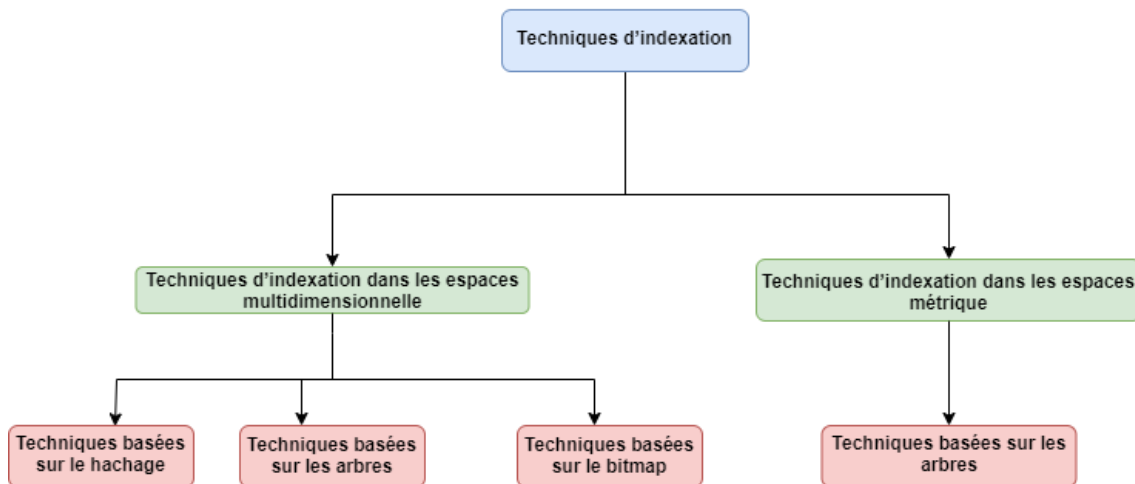


FIGURE 2.2 – Taxonomie globale des techniques d'indexation

### 2.3.1 Techniques d'indexation multidimensionnelle

Un espace multidimensionnel est défini lorsque les éléments de l'ensemble étudié sont des vecteurs, homogènes ou hétérogènes, et que leurs composantes sont totalement ordonnées. [Kouahla, 2013]

Il est un espace multidimensionnel est défini lorsque les éléments de l'ensemble sont considérés comme des vecteurs (c'est-à-dire que les données a un nombre donné de dimensions), homogènes ou hétérogènes, dont les composantes sont totalement ordonnées. Ainsi, les techniques d'indexation dans les espaces multidimensionnels peuvent être classées en trois types principaux en fonction du type de structure utilisé :

- 1) la technique basée sur le hachage
- 2) la technique basée sur les arbres
- 3) la technique basée sur les bitmaps

[BENRAZEK, 2021] Dans ce qui suit, nous passons en revue les trois types de techniques d'indexation multidimensionnelle.

## Technique basée sur le hachage

Il s'agit d'une technique plus populaire dans le domaine de l'indexation des données multidimensionnelles en raison de sa capacité de transformer un élément de données en une représentation à faible dimension (code court composé de quelques bits). Les structures d'indexation basées sur le hachage sont plus efficaces en termes de temps et d'espace de stockage et ils peuvent détecter les données dupliquées dans un grand ensemble de données. Il existe de nombreuses méthodes basées sur la technique du hachage appliquées à plusieurs applications réelles, telles que la vision par ordinateur, la recherche d'informations et l'analyse (par exemple, images, vidéos, documents). Les structures d'indexation basées sur le hachage peuvent être classées en deux courants principaux : le hachage indépendant des données et le hachage dépendant des données (ou hachage basé sur l'apprentissage).[Kouahla *et al.*, 2021]

### 1. Hachage indépendant des données

Parmi les méthodes de hachage indépendantes des données, la méthode Locality-Sensitive Hashing (LSH) développée par Gionis et al est la plus populaire dans la littérature. Elle permet d'extraire l'ensemble suffisant de voisins les plus proches approximatifs (ANNs) dans un espace de grande dimension. L'un des principaux critères de la famille de techniques LSH est la fonction de hachage qui renvoie avec de fortes probabilités, le même bit pour des points de données proches dans l'espace original. Depuis la proposition de LSH, plusieurs variantes ont été proposées pour améliorer la méthode SLH comme : MultiProbe LSH , BayesLSH , Boosted LSH , Superbit LSH , Non-metric LSH , Kernelized LSH (KLSH) et Asymmetric LSH (ALSH) . Cependant, les techniques basées sur LSH souffrent de problèmes de coûts de stockage et de temps de recherche en raison des longs codes binaires et des fonctions de hachage élevées requises lorsque la précision de la récupération est améliorée . En général, les méthodes de hachage indépendantes des données sont bien adaptées aux petites données, mais elles ne sont pas suffisantes pour traiter les grandes données. Le tableau 2 présente un résumé des avantages et des inconvénients des méthodes ci-dessus, ainsi que de leurs difficultés.[Kouahla *et al.*, 2021]

### 2. Hachage dépendant des données

Plusieurs méthodes ont été proposées dans le flux de hachage pour surmonter les problèmes et les limites des méthodes de hachage indépendantes des données, en fonction des données. Ces méthodes sont classées en trois catégories selon le degré de supervision, à savoir :

- le hachage non supervisé
- le hachage supervisé
- le hachage semi-supervisé
- le hachage profond[BENRAZEK, 2021]

L'objectif de ces méthodes est d'ajuster finement la distribution des données dans l'espace de caractéristiques pour obtenir une meilleure précision tout en préservant autant que possible la localité[Chi, 2015]. Cependant, le principal inconvénient des techniques de cette famille réside dans la nécessité d'utiliser un grand nombre de tables de hachage et de longs mots de code, ce qui rend difficile l'obtention de performances satisfaisantes. Il est donc difficile d'obtenir des performances satisfaisantes.

Par conséquent, de nouvelles solutions sont nécessaires pour résoudre le problème de l'optimisation de l'apprentissage des fonctions de hachage et des codes de hachage. [ALLELE, 2021]

### 2.3.2 Techniques basées sur les arbres

Certains auteurs traitent les techniques d'indexation multidimensionnelle comme des méthodes de classification non supervisées. Il est important de noter qu'en classification, la cardinalité des classes est différente, et en classification hiérarchique, la profondeur de toutes les classes n'est pas la même. Les techniques d'indexation peuvent être divisées en deux approches principales :

#### Partitionnement des données (Pas de partition de l'espace)

Le concept principal du deuxième type d'approche d'indexation multidimensionnelle est créé des ensembles de données ou des clusters appelés "formes inclusives". La technique la plus connue dans cette catégorie est l'arbre B [Srinivasan & Carey, 1991] et ses variantes (e.g. arbre B+ [Srinivasan & Carey, 1993]), arbre R [Guttman, 1984], arbre X [Berchtold *et al.*, 1996] et arbre SR [Berchtold *et al.*, 1996].

L'arbre-R [Guttman, 1984] est une hiérarchie basée sur l'arbre-R. L'idée principale de cette structure est de générer un minimum limitant rectangle (MBR) autour de l'objet. Le principal avantage de l'arbre-R est la structure dynamique et le dépouillement hiérarchique de l'arbre, mais il souffre de la malédiction de la dimensionnalité et de l'inefficacité pour les hautes dimensions [Kouahla, 2013]. De plus, le chevauchement des boîtes de délimitation dans le catalogue augmente avec la dimension [Berchtold *et al.*, 1996] ce qui rend les recherches k-nn particulièrement difficiles.

L'arbre-SR (pour sphères et rectangles) est une structure d'indexation proposée en 1997 pour éviter les problèmes de chevauchement entre les cellules [Katayama & Satoh, 1997]. Basé sur le principe de l'arbre-SR d'utiliser la forme limite résultant de l'intersection d'un hyper-rectangle et d'une hypersphère [Katayama & Satoh, 1997]. Bien que cet arbre apporte une solution, il présente également les problèmes suivants La complexité de la table fermée augmente le coût de l'opération d'insertion recherchée. [ALLELE, 2021]

L'arbre-X [Chakrabarti & Mehrotra, 1999] est une structure basée sur l'arbre-R développée pour empêcher le chevauchement entre les MBR grâce au nouveau type de nœud proposé. Ces nœuds sont des nœuds étendus de taille variable appelés Super-nœuds . Grâce à ce type de nœud, l'arbre-X prend en charge l'indexation de grandes données avec moins de chevauchement et moins de réduction des performances par rapport à la structure l'arbre-R. L'arbre-X est un index hybride qui se compose d'une partie hiérarchique (arbre) et d'une partie linéaire (liste). L'arbre X est une structure variable dont la taille et la complexité sont difficiles à calculer car de leur sensibilité à la taille et à la distribution des données [Böhm *et al.*, 2001]. De plus, l'arbre en X consomme beaucoup d'espace mémoire pour le stockage, et ses performances sont limitées à la dimension des données.

#### Partitionnement de l'espace

Ce type de méthode d'indexation multidimensionnelle basée sur les partitions espace en sous-espaces. Toutes les intersections entre ces sous-espaces sont nulles. Plusieurs tech-

niques d'indexation reposent sur ce principe, par exemple : arbre-KD [Bentley, 1979], arbre-PH [Zäschke *et al.*, 2014], et l'arbre-BB plus récent [Sprenger *et al.*, 2019].

L'arbre-KD [Kouahla, 2013] est un index basé sur le partitionnement spatial dimension  $k$  suivant chacun de ses axes. Contrairement aux grilles "simples", toutes les axes ne sont pas considérés en même temps. Les données sont structurées sous forme d'arbre binaire. A chaque niveau de l'indice, le sous-espace courant est divisé en deux sous-espaces selon des dimensions différentes. Le choix (i) de l'axe et (ii) de la valeur sur laquelle les données doivent être doublement partitionnées sur cet axe sont des paramètres importants.

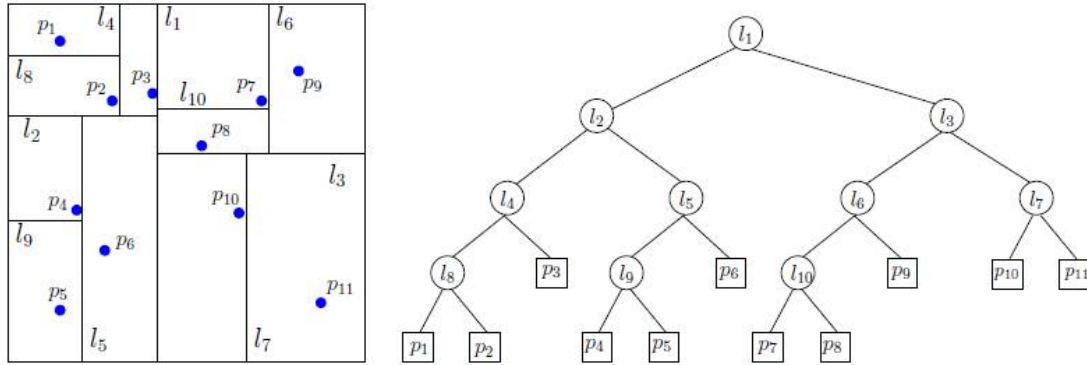


FIGURE 2.3 – Arbre-kD [Rey, 2022]

Récemment, Zäschke *et al.* [Zäschke *et al.*, 2014] ont proposé la structure PATRICIA-Hypercube-tree (PHtree), qui est basée sur la représentation binaire des objets de données sous forme de chaînes de bits [Germann *et al.*, 2009] et une structure quadtree [Finkel & Bentley, 1974], qui utilise des hypercubes pour le partitionnement spatial dans toutes les dimensions de chaque nœud de l'arbre [Mokbel *et al.*, 2003]. Ce partitionnement facilite la navigation vers les nœuds enfants et les entrées stockées. Le stockage des entrées est plus efficace par rapport aux arbres binaires [Zäschke *et al.*, 2014]. D'autres améliorations ont été proposées pour améliorer l'efficacité des structures arborescentes PH [Vancea, 2015] [Adrien, 2014].

Plus récemment, Stefan Sprenger *et al.* ont proposé une nouvelle structure d'index en 2019. Le multidimensionnel est appelé BB-tree [Sprenger *et al.*, 2019]. Cette structure combine KD-tree et X-tree [Berchtold *et al.*, 1996]. Le BB-tree est composé de deux éléments : un arbre de recherche  $k$ -aire et un ensemble de baril à bulles. Les nœuds internes de l'arbre de recherche divisent récursivement l'espace en  $k$  sous-ensembles disjoints, comme un arbre KD, et conservent toutes les données. Dans le seau à bulles à nœud feuille à capacité limitée, inspiré de X-tree. BB-tree montre une meilleure efficacité pour les requêtes de plage, mais la requête Knn indépendamment de la similarité.

### 2.3.3 Techniques basées sur le bitmap

L'index bitmap (également connu sous le nom de BitArray ou index vectoriel) est une structure d'indexation efficace pour la recherche et l'extraction de grandes bases de données et d'entrepôts de données (DW) avec moins de complexité et très efficace lorsque les attributs ont un faible nombre de valeurs distinctes. Cette technique est utilisée par plusieurs systèmes commerciaux populaires tels que Oracle [Antoshenkov, 1995] [Antoshenkov & Ziauddin, 1996] et



SybaseIQ[MacNicol & French, 2004][O'Neil & Quass, 1997]. Bitmap La technique de l'index bitmap est basée sur la représentation de l'existence ou de l'absence d'une propriété spécifique par une séquence de bits où chaque bit (0/1) représente la valeur d'un attribut pour un tuple donné de sorte que la séquence de bits a un 1 en position  $i$  si le  $i$  élément de données satisfait à la propriété, et 0 sinon[Chan & Ioannidis, 1998][Wu *et al.*, 2016]. L'index bitmap utilise des opérations logiques, telles que AND, OR, NOT et XOR, pour répondre et accélérer les réponses aux requêtes complexes[Wu *et al.*, 2002].

La méthode de l'index bitmap est une technique très efficace pour répondre à des requêtes complexes en lecture seule. et pour les données qui ne sont pas fréquemment mises à jour comme un entrepôt de données, mais elle est moins efficace dans d'autres cas (c'est-à-dire pour les données fréquemment mises à jour). Ce problème est causé par la compression où cette dernière est utilisée pour réduire l'espace de stockage (comme nous l'avons dit précédemment), mais à chaque opération de mise à jour, il est nécessaire de décompresser les données ainsi que décoder et d'encoder le bitmap, et cette opération est coûteuse[Athanassoulis *et al.*, 2016]. Manos *et al.*[Athanassoulis *et al.*, 2016] ont proposé un nouvel index bitmap nommé UpBit (Updatable Bitmap) pour surmonter ce problème. Cet index offre des mises à jour efficaces sans affecter les performances de lecture. L'index UpBit ajoute un vecteur de mise à jour supplémentaire pour chaque vecteur bitmap dans lequel les processus de mise à jour seront effectués sur le vecteur de mise à jour, où ce dernier stocke les mises à jour correspondant à sa valeur bitmap uniquement. Bitmap minimise le coût du décodage et améliore la navigation grâce à l'utilisation de pointeurs de fermeture sur les vecteurs de bit.

### 2.3.4 Techniques d'indexation métrique

La structure de l'index dans l'espace multidimensionnel est plus stricte car Ils dépendent fortement du type ou, plus précisément, des propriétés géométriques. ces dépendances rendent très difficile la mise en œuvre de structures flexibles pour des systèmes fiables. Par conséquent, l'objectif de l'indexation s'est partiellement déplacé de l'espace multidimensionnel à l'espace métrique. Les espaces métriques ont été proposés comme abstractions des données de masse génériques[Mao *et al.*, 2015]. Formellement, un espace métrique  $(O, d)$  est un ensemble de points  $O$  (où,  $O \neq \emptyset$ ) auquel on associe une notion de distance  $d(O \times O) \rightarrow \mathbb{R}^+$  entre les éléments qui répond aux propriétés suivantes :

- Non-négativité :  $(x, y) \in O^2, d(x, y) \geq 0$ ;
- Identité :  $x \in O, d(x, x) = 0$ ;
- Symétrie :  $(x, y) \in O^2, d(x, y) = d(y, x)$ ;
- Inégalité triangulaire :  $(x, y, z) \in O^3, d(x, y) + d(y, z) \geq d(x, z)$ .

Les techniques d'indexation dans les espaces métrique disparaissent la notion de l'axe utilisé dans les techniques basé sur les arbres dans les espaces multidimensionnels[Kouahla, 2013]. Deux différents types de ces techniques ont été développées. La première basée sur le partitionnement de l'espace et la deuxième basée sur le non-partitionnement de l'espace.

#### Techniques basées sur le partitionnement de l'espace

Dans la littérature, deux techniques de partitionnement ont été développées : la première technique est basée sur le partitionnement par hyper-sphère (ou boule) comme dans : VP-tree, mVP-tree et MM-tree, etc. tandis que la deuxième technique est basée sur

le partitionnement par hyperplan comme dans : GH-tree , GNAT-tree et EGNAT-tree , etc.[Kouahla *et al.*, 2021]

VP-tree (Vantage Point) est basé sur une structure d'index hiérarchique qui divise l'espace par des sphères. VP-tree sélectionne aléatoirement un point Une observation appelée pivot(p), calcule alors le pivot sélectionné avec l'étude des points de l'espace pour définir un cycle B(p, dm) qui divise l'espace en deux Espaces disjoints.[ALLELE, 2021]

Le mVP-tree est proposé pour répondre au problème de performances réduites dans la recherche de similarité de l'arbre VP dans les espaces métriques à haute dimension. L'arbre mVP (multiple Vantage Points tree) est une extension de l'idée de l'arbre VP qui utilise plusieurs points d'observation au lieu d'un seul. Le principal avantage de l'arbre mVP par rapport à l'arbre VP est qu'il utilise également des distances pré-calculées (à l'étape de construction) pour améliorer la vitesse de recherche et réduire le nombre de calculs de distance et d'erreurs. et de diminuer le temps nécessaire à l'exécution des requêtes. Les expériences présentées dans montrent que mVP-tree améliore légèrement le VP-tree, mais ce n'est pas dans tous les cas, tandis qu'une amélioration plus remarquable est obtenue lorsque plusieurs pivots par nœud sont utilisés.[BENRAZEK, 2021]

Le GH-tree (Generalised Hyper-plane) est basé sur le principe du partitionnement de l'espace métrique à l'aide d'hyperplans. Le principe de cette technique est la partition récursive des données en deux groupes selon les deux pivots qui sont choisis dans chaque partitionnement en fonction de la distance la plus éloignée.[Benrazek *et al.*, 2020]

L'arbre-GNAT est une standardisation utilise k pivot dans chaque noeud dit interne a la place de deux.L'idée de l'arbre-GNAT repose sur le choix de k points de séparation(les pivots) à partir de l'ensemble que nous désirons indexer et le reste des données sont partitionnées selon la fonction de la distance la plus proche par rapport les pivots choisis[Kouahla, 2013][Brin, 1995].

### Techniques basées sur le non-partitionnement de l'espace

Cette catégorie ne nécessite pas de partitionnement de l'espace. Nous retrouvons essentiellement la famille M-tree parmi les familles qui utilisent ce type de partitionnement (partitionnement des données).

M-tree[Ciaccia *et al.*, 1997] est une arborescence métrique structure équilibrée en hauteur, permettant des mises à jour incrémentales basées sur le regroupement de données dynamiques en boules (ou hyper-sphères). Le M-tree stocke certaines données dans les nœuds internes pour des raisons de routage.

Le M-tree souffre du problème du chevauchement des sous-espaces, ce qui augmente le nombre de calculs de distance pour répondre à une requête [Kouahla *et al.*, 2019]. Plusieurs structures récentes qui partagent les mêmes principes que le M-tree et le Slim-tree[Traina *et al.*, 2000]on font partie. Slim-tree améliore la structure M-tree grâce à une nouvelle technique de fractionnement basée sur l'arbre couvrant minimum (MST).

Slim-tree réduit également le coût de construction.En outre, il introduit une méthode de post-traitement qui réduit le chevauchement, par conséquent le coût de la recherche. L'inconvénient majeur de cet algorithme est la possibilité de générer des nœuds avec peu d'objets et/ou des nœuds vides, ce qui réduit considérablement les performances de l'index, surtout dans les grands espaces[Traina *et al.*, 2002].

Les travaux de Murgante et al.[BENRAZEK, 2021] visent à éviter le partitionnement insatisfaisant des nœuds et à réduire le le chevauchement régional dans la structure M-tree[Ciaccia *et al.*, 1997]. Les auteurs ont proposé une nouvelle structure d'indexation

métrique appelée MX-tree basée sur la structure M-tree originale. MX-tree met en œuvre le concept de super-nœuds inspiré de la structure de l'arbre X[Berchtold *et al.*, 1996].

L'arbre BCCF est une technique d'indexation de l'espace métrique qui divise de manière récursive l'ensemble des données en deux régions disjointes en sélectionnant deux pivots à chaque itération. Il s'agit d'une structure similaire à l'arbre GH en ce sens que les deux subdivisent l'ensemble de données de manière récursive via le principe de l'hyperplan généralisé. La différence est que l'arbre BCCF utilise différentes techniques de partitionnement des données, différentes stratégies de sélection des pivots, et différents types de nœuds.[BENRAZEK, 2021]

a partir de cette étude on peut conclure que la technique d'indexation universelle capable de gérer des ensembles de données arbitraires n'a pas encore été trouvée et sa réalisation, qui répond aux exigences de l'indexation des données massives mentionnées dans la section 2.2,est presque irréalisable.

Notre analyse montre que les techniques d'indexation basées sur les arbres ont abordé en célébrité ces derniers années, puisque les structures sont des structures dynamiques permet des changement de données.Ces structures ne nécessitent pas de ré-organisations périodiques de la structure qui exige des ressources plus chers. Mais elles garantissent une indexation ininterrompu avec une complexité logarithmique de l'insertion et de la recherche.vu que ces avantages nous pensons que l'indexation arborescente sera le bon choix pour notre index. [ALLELE, 2021]

Pour mieux comprendre structures d'indexation arborescence, nous allons faire une analyse comparative entre ces techniques. Les approches multidimensionnelles sont résumées dans le tableau 2.1 et les approches métriques sont résumées dans le tableau 2.2.

Sur la base des études précédentes, nous pouvons introduire une taxonomie simple des techniques d'indexation dans un espace métrique comme illustré à la figure 2.4, et une taxonomie qui représente les techniques d'indexation dans un espace multidimensionnel, comme illustré à la figure 2.5.

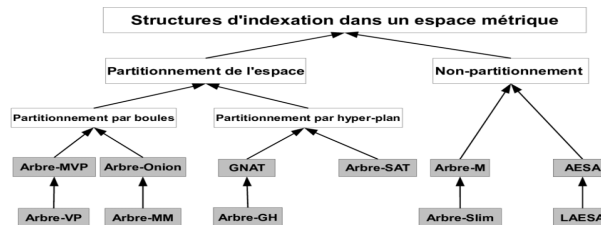


FIGURE 2.4 – Taxonomie des techniques d'indexation métrique[Joyce, 2020a]

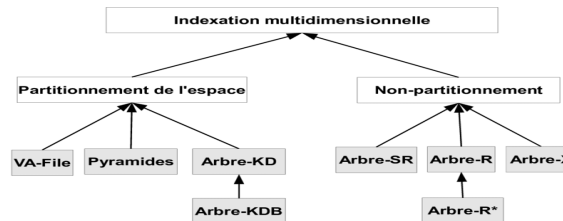


FIGURE 2.5 – Taxonomie des techniques d'indexation multidimensionnelle[Joyce, 2020a]

| <b>Proposition</b> | <b>Avantages</b>   | <b>Inconvénients</b>   |
|--------------------|--|--|
| Arbre-R            | <ul style="list-style-type: none"> <li>-Dynamisme de la structure</li> <li>-Création de cellules filtrantes REM</li> <li>-Contrainte de couverture minimale</li> </ul>                         | <ul style="list-style-type: none"> <li>-Pas efficace pour les requêtes ponctuelles</li> <li>-Nécessitent beaucoup d'espace, de temps et de complexité de calcul</li> </ul>                             |
| Arbre-PH           | <ul style="list-style-type: none"> <li>-Rapide et efficace en terme de rendement de l'espace.</li> <li>-Rapide en terme de performance des requêtes par rapport aux arbres binaires</li> </ul> | <ul style="list-style-type: none"> <li>-Accroître la dégradation</li> </ul>  |
| Arbre-BB           | <ul style="list-style-type: none"> <li>-Meilleures performances par rapport à R*-tree, Kdtree, PH-tree, et VA-file</li> </ul>  | <ul style="list-style-type: none"> <li>-Ne pas supporter la recherche k-nn</li> </ul>  |
| Arbre-X            | <ul style="list-style-type: none"> <li>-Contrôle du chevauchement</li> </ul>   | <ul style="list-style-type: none"> <li>-Complexité de la limite maximale</li> <li>-Consomme beaucoup d'espace mémoire</li> <li>-Les performances sont limitées par la dimension des données</li> </ul> |
| Arbre-SR           | <ul style="list-style-type: none"> <li>-Simple construction</li> <li>-Raffinement</li> <li>-Taux de chevauchement réduit</li> </ul>  | <ul style="list-style-type: none"> <li>-Complexité des formes</li> <li>-Algorithme d'insertion et de recherche coûteux</li> </ul>  |

TABLE 2.1 – Tableau comparatif des approches multidimensionnelles.

| Proposition  | Avantages  | Inconvénients  |
|--------------|--|--|
| Arbre-VP     | -Mise en œuvre simple  | -Distance et temps les plus élevés<br>-Les coûts de la recherche augmentent dans les grandes dimensions            |
| Arbre-M      | -Structure dynamique<br>-Réduction des calculs de distances                            | -Non adapté aux données fortement groupées<br>-Problème des chevauchements<br>- Dégradation au passage à l'échelle |
| Arbre-slim   | -Performante par rapport arbre-M<br>-Réduction du taux de chevauchement                | -Réduction du taux de chevauchement  |
| Arbre-GNAT   | -Affiner la recherche<br>-Pas de chevauchement   | -Forme compliquée à manipuler<br>- Structure statique  |
| Arbre-GH     | -Partitionnement simple<br>-Structure dynamique<br>-Réduction du taux de chevauchement | -Forme compliquée à manipuler<br>-Dégradation au passage à l'échelle   |
| Arbre-mVP    | -Peu affecté à grande échelle  | -Structure statique  |
| Arbre-MM     | -Meilleur partitionnement de l'espace  | -Dégradation au passage à l'échelle<br>- Dégénérative de l'index   |
| Arbre-oignon | -Meilleur partitionnement de l'espace  | -« Réinsertion » des objets (semi-équilibrage)<br>-Dégradation au passage à l'échelle                              |
| Arbre-UD     | -Regroupement des données les plus similaires  | /  |

TABLE 2.2 – Tableau comparatif des approches métriques.

## 2.4 Conclusion

Dans ce chapitre, nous avons présenté un état de l'art sur les techniques d'indexation dans les espaces métriques et multidimensionnels. Nous avons commencé par présenter les besoins en matière d'indexation de données massives, puis nous avons résumé les différentes techniques d'indexation métriques et multidimensionnelles. Nous concluons cette étude par une analyse comparative et une taxonomie des techniques d'indexation à base d'arbres mentionnées dans le chapitre.

# Chapitre 3

## Conception

### 3.1 Introduction

Avec l'évolution technologique que le monde connaît aujourd'hui, de nombreuses personnes profitent à l'aide du technologie qui peut leur apporter dans de nombreuses tâches quotidiennes et c'est le cas pour les personnes Handicapées. Les disciplines technologiques permettent à ce type de personnes de vivre une vie meilleure et plus indépendante et bien sûr dans de meilleures conditions.

Alors dans ce chapitre, nous allons présenter notre proposition de l'approche d'indexation sur les donnée de l'IoT au profit des personnes en situation Handicap qui leur donne la possibilité de communiquer efficacement et en temps réel dans n'importe quelle activité de la vie quotidienne.

### 3.2 Objectif de l'approche proposée : "Web Sémantique des Objets"

L'approche web sémantique des objets a plusieurs objectifs principaux qui sont :

- Fournir aux handicapées des opportunités d'accroître l'indépendance en réduisant le besoin d'aide personnelle.
- Notre objectif principal se résume dans la manière d'indexer les données massives issues des différents objets connectés associés aux activités des personnes qui le besoins puis envoyer ces données au module Web sémantique pour faire l'inférence et le raisonnement sur des connaissances propres.
- Garantir l'intéropérabilité entre les différents systèmes d'IdO hétérogènes.
- Une inférence qui conduit à de nouveaux faits et donc à de nouvelles connaissances et à une base de connaissances beaucoup plus riche.
- Faire appel aux techniques d'indexation de données afin de fournir au module sémantique le minimum de données nécessaires et efficaces permettant d'optimiser et améliorer la prise de décision au profit des personnes handicapées.

### 3.3 L'architecture du système proposé

Les éléments qui composent le système sont présentés dans la figure ci-dessous : Notre système commence la collecte des information utiles sur l'handicap à l'aide des capteurs

tels que des capteurs de signes vitaux. Chaque capteur peut avoir son propre fabricant et sa propre unité. En utilisant l'ontologie, l'interopérabilité est garantie car elle peut comprendre les différentes représentations d'une connaissance sur un tel capteur. Sur la base d'un moteur de raisonnement, nous avons utilisé des règles SWRL pour déduire de nouvelles connaissances. Tout ces informations ainsi que les règles qui sont stockés dans un fichier RDF on peut les indexer en utilisant une techniques d'indexation afin de réduire le temps de recherche et rendre la communication en temps réel.

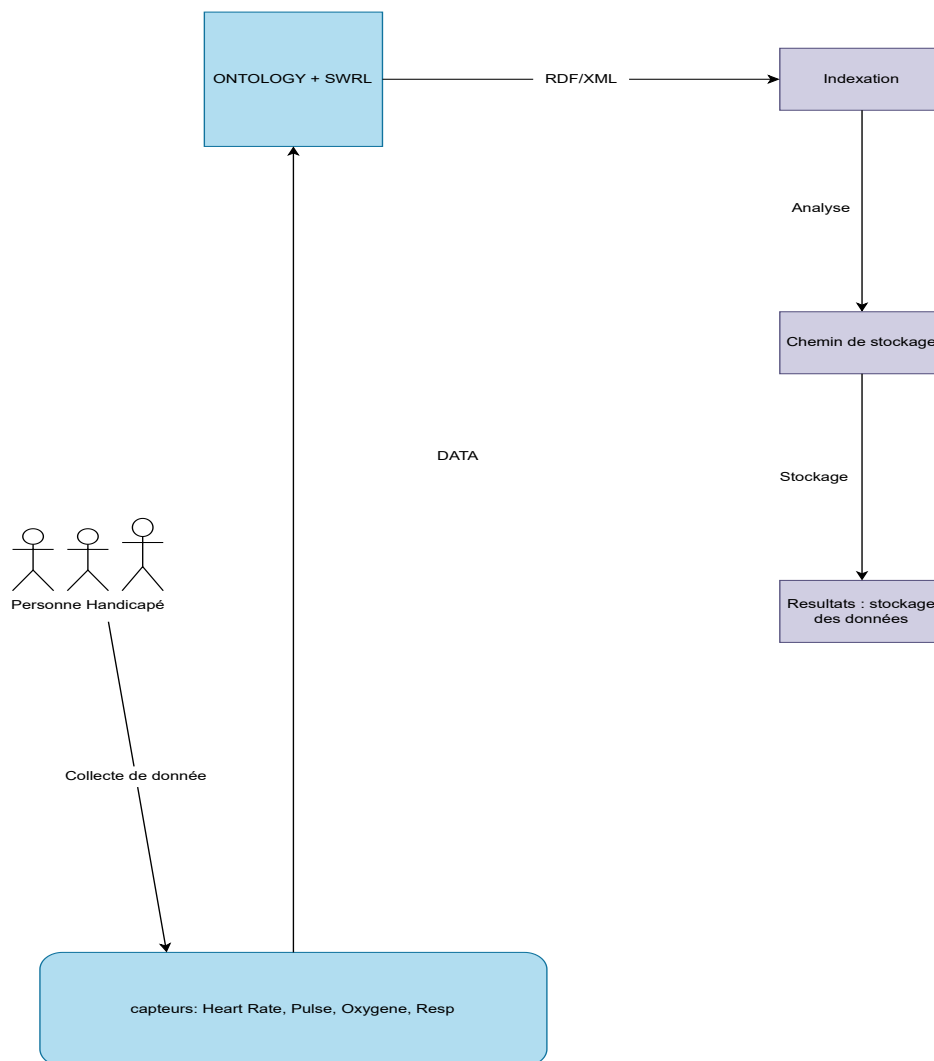


FIGURE 3.1 – Enchaînement de la proposition

### 3.4 Présentation des connaissances

Dans notre contexte de travail, nous avons créé une ontologie qui décrit mieux le domaine de la santé lié au patient handicapé. Nous avons essayé de nous concentrer sur



le suivi des signes vitaux du handicap pour l'aider à faire face à sa situation actuelle. Le manque d'informations et d'ensembles de données dans le domaine sémantique a été un grand défi pour nous, ce qui a limité notre ontologie à dépendre uniquement des signes vitaux pour prendre la décision.

### 3.4.1 Concepts et classes

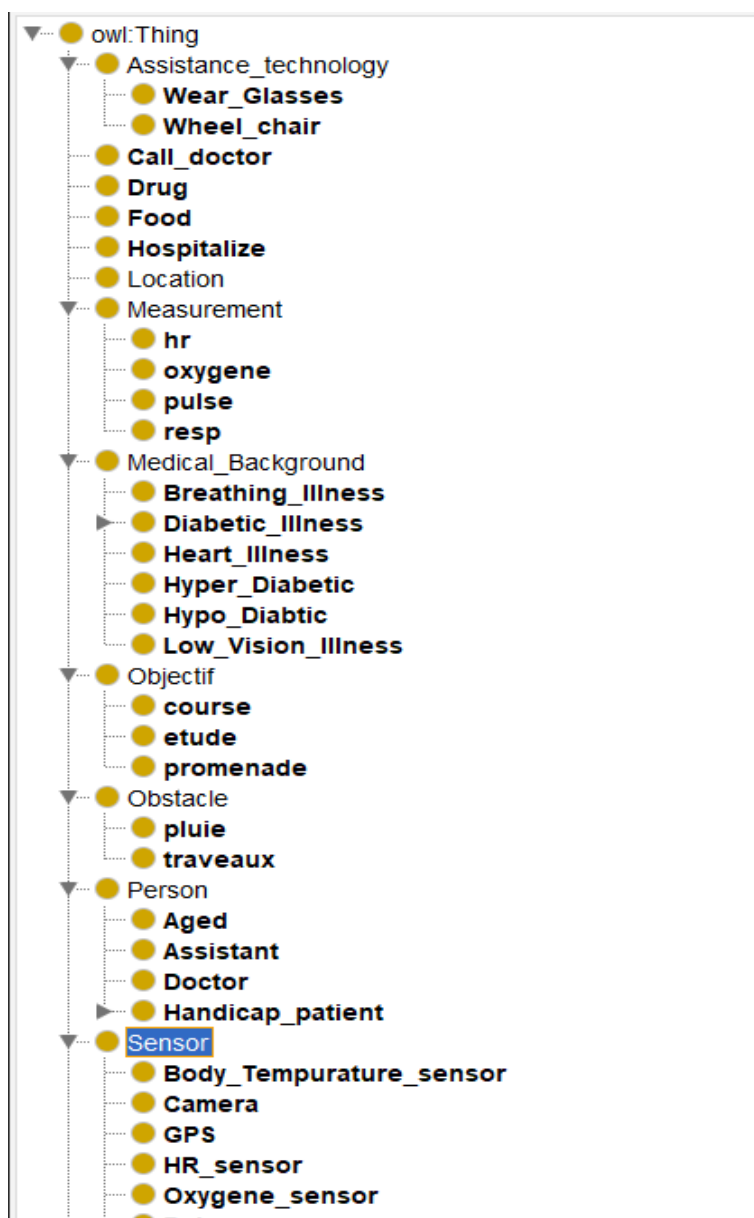


FIGURE 3.2 – Concepts et classes



FIGURE 3.3 – Concepts et classes

| Concepts        | Description   |
|-----------------|---|
| Person          | C'est la classe qui représente un être humain.                                    |
| Handicap_person | C'est la classe qui représente une personne handicapé.                            |
| Doctor          | Le médecin est la classe qui représente personne qui peut traiter les handicapés. |

|                    |   |
|--------------------|---|
| Medical_Background | Medical_Background représente les antécédents médicaux d'un patient handicapé.                              |
| Diabetic_Illness   | Diabetic_Illness est une sous-classe qui représente des antécédents médicaux.                               |
| Breathing_Illness  | Breathing_Illness est une sous-classe qui représente des antécédents médicaux.                              |
| Heart_Illness      | Heart_Illness indique si le patient handicapé a des problèmes cardiaques dans ses antécédents médicaux.     |
| Low_Vision_Illness | Low_Vision_Illness indique si le patient handicapé a des problèmes de vision dans ses antécédents médicaux. |
| Location           | C'est la classe qui représente un endroit sur la carte défini par des coordonnées.                          |
| Body_Temp_sensor   | Body_Temperature_sensor c'est le capteur qui mesure la température corporelle du handicapé.                 |
| Oxygene_sensor     | Oxygen_sensor est un capteur qui mesure la quantité d'oxygène dans le sang.                                 |
| Pulse_sensor       | pulse_sensor représente un capteur qui mesure le débit sanguin.   |
| HR_sensor          | HR_sensor représente un capteur qui mesure la fréquence cardiaque d'une personne handicapée.                |
| RESP_sensor        | Le capteur respiratoire est un appareil qui mesure la fréquence respiratoire de la personne handicapée.     |
| Objectif           | Objectif est le concept qui représente une tâche que le handicapé veut accomplir.                           |

|                      |   |
|----------------------|---|
| Obstacle             | Obstacle est le concept qui représente quelque chose qui interrompt le handicap pendant qu'il effectue une tâche objective. |
| Type_of_disability   | Le cocept Type_of_disability représente la limitation de la personne.   |
| Assistant_Technology | Représente l'ensemble d'outils qui peuvent aider une personne handicapée.   |
| Measurement          | C'est le concept qui représente un nombre récupéré à partir de capteurs.  |
| hr                   | Hr représente la valeur récupérée par HR_sensor.  |
| oxygene              | oxygène représente la valeur récupérée à partir du Oxyegene_sensor.   |
| pulse                | pulse représente la valeur récupérée par Pulse_sensor.  |
| resp                 | RESP représente la valeur récupérée par le RESP_sensor.   |

TABLE 3.1 – Tableau descriptif des concepts

### 3.4.2 Relations entre les objets

Le tableau suivant contient quelques propriétés qui décrivent les relations entre les objets.

| Object Propriety          | Domain           | Range       | Function | Description   |
|---------------------------|------------------|-------------|----------|---|
| Benefit_from_doc<br>_call | Handicap_patient | Call_doctor | /        | Cet attribut représente un intérêt du médecin de garde à mettre en relation des patients handicapés avec des médecins de garde. |
| Benefit_from_drug         | Handicap_patient | Drug        | /        | L'attribut représente la relation bénéfique du médicament qui relie le patient handicapé au service du médicament.              |
| Benefit_from_food         | Handicap_patient | Food        | /        | La propriété représente un intérêt pour la nourriture, reliant les patients handicapés au service de la nourriture.             |

|                                  |                            |                          |   |   |
|----------------------------------|----------------------------|--------------------------|---|---|
| Benefit_from<br>_hospitalization | Handicap_patient           | Hospital                 | / | L'attribut représente la relation bénéficié de l'hospitalisation relie le patient handicapé au service hospitalier. |
| has_oxygene_sensor               | Person<br>Handicap_patient | Sensor<br>Oxygene_sensor | / | Cette propriété représente la relation Has_oxygene_sensor qui relie l'handicapés au capteur d'oxygène.              |
| has_hr_sensor                    | Person<br>Handicap_patient | Sensor<br>HR_sensor      | / | Cette propriété représente la relation Has_hr_sensor qui relie le handicap au capteur de fréquence cardiaque.       |
| Has_resp_sensor                  | Person<br>Handicap_patient | Sensor<br>RESP_Sensor    | / | Cet attribut représente la relation Has_resp_sensor qui relie le handicap au capteur respiratoire.                  |

|                  |                            |                        |        |  |
|------------------|----------------------------|------------------------|--------|--|
| has_pulse_sensor | Person<br>Handicap_patient | Sensor<br>Pulse_sensor | /      | L'attribut représente la relation Has_pulse_sensor qui relie le handicap au capteur de pulse.                          |
| treatedBy        | Person<br>Handicap_patient | Person<br>Doctor       | treats | La propriété représente la relation treatedBy qui relie le patient handicapé au médecin qui décrit l'action treatedBy. |

TABLE 3.2 – Tableau des relations entre les objets

### 3.4.3 Proportions littérales

| Data Propriety | Domain  | Range   | Description  |
|----------------|---|---------|--|
| Age            | Person<br>Doctor<br>Handicap_patient<br>Assistant | Integer | Représente un nombre qui décrit l'âge de la personne.      |
| Condition      | Weather   | String  | Représente une description des conditions météorologiques. |

|             |   |         |  |
|-------------|---|---------|--|
| First_Name  | Person<br>Doctor<br>Handicap_patient<br>Assistant | String  | Représente le nom donné à une personne après sa naissance.             |
| Name        | Person<br>Doctor<br>Handicap_patient<br>Assistant | String  | Représente le nom de famille que tous les proches frères et sœurs ont. |
| Sexe        | Person<br>Doctor<br>Handicap_patient<br>Assistant | String  | Représenter la description sexuelle de la personne.                    |
| Temperature | Weather   | Float   | Un nombre qui représente la mesure du temps froid ou chaud.            |
| time        | Measurement                                       | Integer | Représenter un nombre qui décrit un moment du temps.                   |
| value       | Measurement                                       | Integer | Représente un nombre qui décrit une mesure.                            |

TABLE 3.3 – Tableau des proportions littérales

### 3.5 La structure d’indexation proposée

Dans cette section nous allons parler sur la structure d’indexation proposée et la raison qui nous a poussé a choisir cette solution. L’indexation c’est organiser et enregistrer



notre data dans le bon endroit pour accéder et récupérer la plus tard facilement avec un temps de recherche limité, le choix d'indexation arborescente c'est à cause de temps de recherche logarithmique et parfois même l'insertion, c'est quelque chose que nous pouvons pas trouver mieux que ça.

La structure d'indexation utilisée est un arbre n-aire, notre but est de regrouper les fichiers RDF en fonction des contraintes, ainsi le niveau 1 de l'arbre est pour les types d'handicap, chaque handicap sera indexé en fonction de son type, puis au niveau 2 nous avons l'objectif comme contrainte, dans ce niveau chaque groupe d'handicaps qui ont le même type et le même objectif sera indexé, en descendant encore pour arriver au niveau 3, c'est le niveau dont la contrainte est la l'obstacle rencontré par l'handicap. Dans cette étape chaque fils de notre arbre contient un groupe de patient qui sont similaire en fonction de type, objectif et obstacle et ça c'est notre bénéfice pour gagner le temps, dans la recherche par exemple à la place de parcourir la totalité de l'arbre en descend juste dans la branche de similarité.

L'étape suivante a également un second arbre, dans ce cas binaire (ou arbre de décision) la sortie du premier arbre sera insérée dans le second sur la base du tableau de chiffres (température, hr, oxygène ...), nous allons calculer la moyenne du tableau, si la moyenne est inférieure à la racine nous l'insérons à gauche et l'inverse sinon. Il réduit la complexité de la recherche de  $O(n)$  à  $O(\log n)$ . Comme sa nous avons gagné en simulant une communication plus rapide et en temps réel. La figure 3.2 présente bien ce qu'on a parlé toute a l'heure.

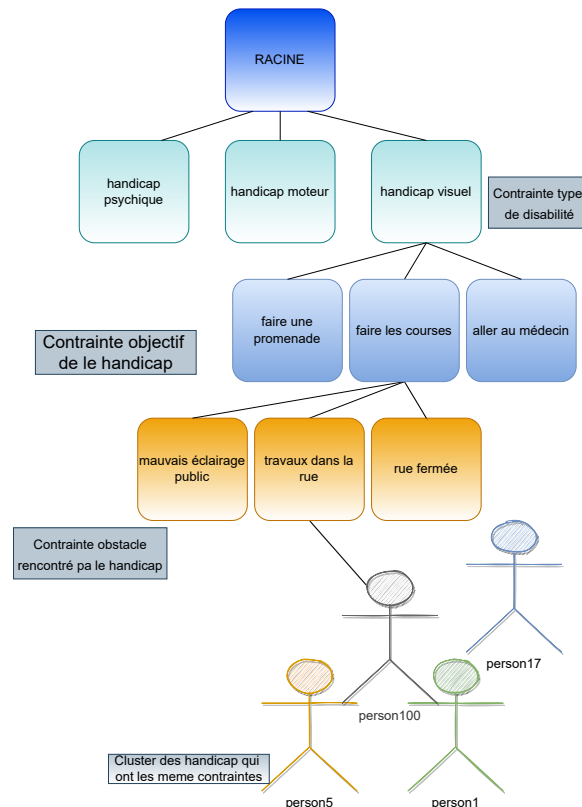


FIGURE 3.4 – arbre SB

L'approche proposée est appelée Arbre-SB(SEMANTIQUE ET BINAIRE).L'arbre-SB est une structure arborescente proposée pour indexer les données massives de l'IdO ins-

pirer de la méthode CAH(classification hiérarchique ascendance) et l'arbre BCCF .Cette dernière est basée sur le regroupement des données les plus similaires d'une manière descendante. A chaque étape de création de processus de l'arbre-SB, les personnes handicapés qui ont la même contraintes descendent dans le même chemin afin d'avoir un groupe (cluster) des personnes qui ont les mêmes contraintes, comme le montre la figure3.2

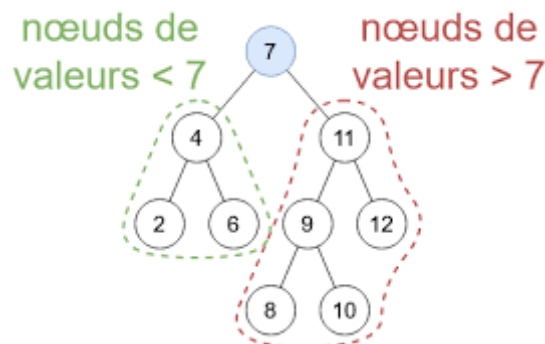
### Trois niveaux de noeuds pour la construction :

- **Niveau des noeuds de la contraintes : type handicap :** Les noeuds de ce niveau contiennent le type de l'handicap et un pointeur vers les sous-arbres.
- **Niveau des noeuds de la contraintes : objectif :** Les noeuds de ce niveau contiennent l'objectif de l'handicap et un pointeur vers les sous-arbres.
- **Niveau des noeuds de la contraintes : obstacle :** Les noeuds de ce niveau contiennent l'obstacle rencontré et un pointeur vers les feuilles.
- **Niveau des noeuds feuilles :** Les Noeuds de ce niveau contient le cluster des handicap qui sont similaire par rapport au trois contraintes précédentes.

Maintenant la partie qui vient c'est la deuxième étape de l'indexation quand nous avons utiliser un arbre binaire de recherche pour spécifier en plus notre chemin d'index. l'insertion dans cet arbre est comme suit :

- L'extraction de vecteur des caractéristique numérique [l'age,hr,pulse,resp,o2] à partir du fichier rdf de patient.
- A la base de ces caractéristique on fait la construction et l'insertion dans l'arbre de tel sorte si la moyenne de vecteur de l'objet qui vient être inférieure à la racine on l'insère à gauche et on l'insère à droite dans l'autre cas.

**Exemple d'un arbre binaire de recherche :** la figure suivante montre un exemple d'un arbre binaire.



## 3.6 Scénario d'application

Prenant un exemple réel d'une personne en situation de handicap qui va faire ses courses ; En s'appuyant sur le système proposé qui combine entre l'Internet des Objets et le Web Sémantique et l'indexation des données massives les étapes à suivre sont :

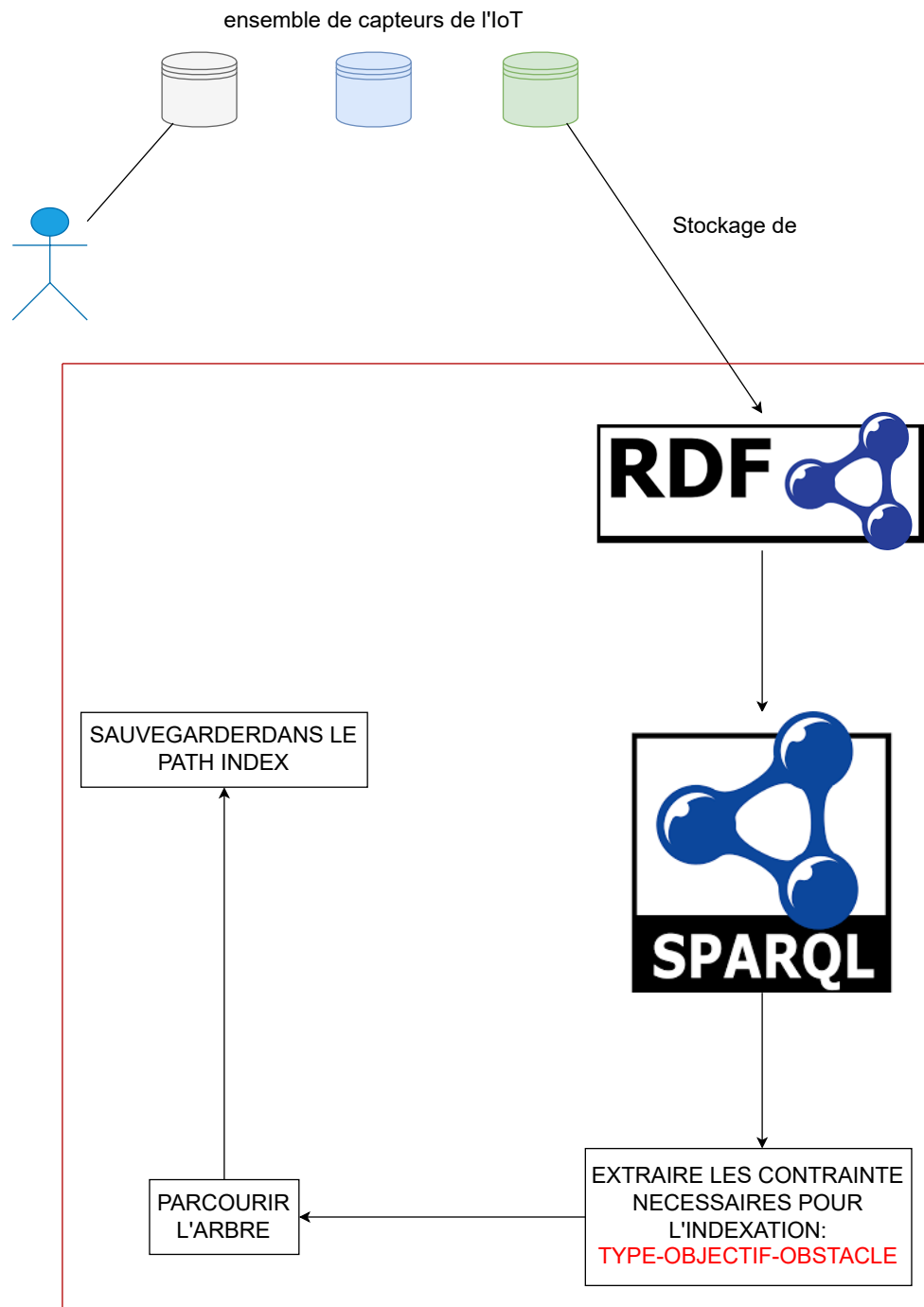


FIGURE 3.5 – Schéma descriptif du scénario de l'application

1. Introduction des informations de la personne handicapée (son nom, son âge, etc.) ainsi que l'objectif qu'elle va atteindre. Dans cet exemple c'est bien la supérette.
2. Dans la deuxième étape, sur la base du modèle ontologique, les règles basées sur le moteur d'inférence sont appliquées en fonction des connaissances déjà présentes dans l'ontologie pour obtenir le résultat attendu. Les règles basées sur le moteur

d'inférences seront appliquées en plus des connaissances déjà présentes dans l'ontologie pour acquérir de nouvelles connaissances ; par exemple, si l'individu dit qu'il a une déficience auditive et par conséquent il n'entend rien quand il marche dans la rue. En prenant en compte cette information, le système le rendra plus prudent, grâce à la vibration pour protéger son capteur en cas de danger. Autre cas, si la personne est aveugle, grâce aux capteurs de distance (comme la RFID) plusieurs tâches seront facilitées telles que : dès qu'elle s'approche de la porte, elle sera ou lorsqu'il y a un obstacle (escalier, pierre...) il sera possible de l'avertir par des capteurs installés sur les dispositifs portés par cette personne en envoyant des signaux de vibration.

3. Dans une dernière étape et après avoir exécuté les règles, des requêtes SPARQL seront effectuées pour recommander des solutions aux personnes handicapées afin de les aider à éviter les problèmes.
4. Pour le processus d'indexation nous allons supposer une personne avec les valeurs : name=Ahmed, age=33, type\_of\_dysability=handicap\_moteur, objectif=faire une promenade, obstacle=travaux dans la rue, la première des chose c'est l'extraction du fichier RDF correspond au malade, puis on extraire le type d'invalidité et l'objectif ainsi que l'obstacle, avec ses information nous pouvons l'insérer dans notre arbre selon le chemin :  
**niveau 1** : handicap\_moteur, **niveau 2** : faire une promenade, **niveau 1** : travaux dans la rue. Maintenant si une deuxième personne avec les mêmes contraintes (handicap\_moteur, faire une promenade, travaux dans la rue) le système l'insère avec Ahmed dans le même cluster, Mais si la personne qui vienne a des contraintes différentes le système l'insère dans un chemin différent dans un cluster différent et ainsi de suite.

## 3.7 Conclusion

Dans ce chapitre, nous avons présenté l'architecture du système, en mettant en évidence les technologies du Web sémantique qui garantissent l'interopérabilité entre les différents capteurs. Ensuite, nous avons également présenté le modèle ontologique (ses concepts, les propriétés reliant les objets et celles liées aux données) suivi des règles SWRL. Nous avons également expliqué la méthode d'indexation utilisé pour réduire le temps de recherche.

# Chapitre 4

## Implémentation

### 4.1 Introduction

Après avoir expliqué les détails de la conception de notre système dans le chapitre précédent, nous allons présenter l’environnement de développement et les bibliothèques utilisées dans notre système. Explication des principales parties du code et des résultats.

### 4.2 Environnement

#### Matériel informatique

- CPU : I7 7ème génération
- GPU : Intel(R) HD Graphics 620
- RAM : 8 GB
- Drive : 180 GB SSD

### 4.3 Outils de développement

#### 4.3.1 Protege

Protege est un éditeur d’ontologie libre et gratuit et un cadre pour la construction de systèmes intelligents. C’est un logiciel soutenu par une large communauté académique, des gouvernements et d’entreprises qui l’utilisent pour construire des solutions basées sur la connaissance dans divers domaines tels que les soins de santé, le commerce électronique et la modélisation organisationnelle. [Joyce, 2020a]

#### 4.3.2 Visual Studio Code

Visual Studio Code est un éditeur de code source puissant qui fonctionne sur les ordinateurs et qui est disponible en téléchargement gratuit sous Windows, macOS et Linux. Il est livré avec un support intégré JavaScript, TypeScript et Node.js et dispose d’un riche écosystème d’extensions pour d’autres langages (tels que C++, C sharp, Java, Python, PHP, Go) et les moteurs d’exécution (tels que .NET et Unity).[Joyce, 2020b]

## 4.4 Langage de Programmation

### 4.4.1 Python

Python est un langage de programmation interprété, orienté objet et de haut niveau. Il est un langage facile à apprendre en raison de la simplicité de sa syntaxe. Python supporte les modules et les paquets qui encouragent la modularité des programmes et la réutilisation du code. Il contient de nombreuses bibliothèques utilisées dans l'apprentissage automatique et l'IA. Les développeurs tombent amoureux de ce langage de programmation en raison de la productivité croissante qu'il offre.[Joyce, 2020c]

### 4.4.2 Bibliothèques

- **Owlready2** : Il s'agit d'un packaging permettant de manipuler des ontologies OWL 2.0 en Python. Il peut charger, modifier, sauvegarder des ontologies, et il supporte le raisonnement via HermiT et Pellet. Owlready permet un accès transparent aux ontologies OWL.
- **CSV** : Le module csv implémente des classes pour lire et écrire des données tabulaires au format CSV.
- **Names** : Générer des noms aléatoires en fonction du sexe.
- **Math** : Ce module permet d'accéder aux fonctions mathématiques définies par le standard C.
- **PyQt5** : Il s'agit d'un ensemble complet d'interfaces Python pour Qt v5. comme plus de 35 modules d'extension et nous permet de développer facilement des interfaces graphiques.

## 4.5 Data set

**BIDMC** : Cette base de données contient des signaux vitaux extraits de la base de données de formes d'onde appariées MIMIC II, qui est plus importante, ainsi que des annotations manuelles de la respiration effectuées par deux annotateurs, à l'aide du signal d'impédance respiratoire.[BID, n.d.]

## 4.6 Développement du système

### 4.6.1 Importation de bibliothèques

Avant de faire que ce soit, nous devons importer les bibliothèques que nous allons utiliser sur le premier script qui est extractor.py qui charge l'ontologie et les données recueillies à partir d'un fichier csv. Puis génère des individus dans l'ontologie en fonction de ces données.

```
C: > Users > YOGA > Desktop > final > extractor.py > ...
1  #LIBRARIES
2  from faulthandler import disable
3  import names, csv, random
4  import owlready2 as owl
5  import math
6
7  print("Libraries imported. \n")
```

FIGURE 4.1 – L’importation de bibliothèques

## 4.7 Le modèle ontologique

### 4.7.1 Importation de l’ontologie

La figure suivante montre comment importer une ontologie en Python.

```
8  #Load Ontology
9  onto = owl.get_ontology("finale-kh.owl").load()
0  print("Ontology Loaded.")
1
2
```

FIGURE 4.2 – Importation de l’ontologie

### 4.7.2 Initiation de l’ontologie

Comme nous l’avons mentionné précédemment, nous avons utilisé le logiciel Protege pour créer notre première ontologie. Dans cette section, nous discuterons de la création de concepts, les Propriétés.

1. **Concepts** : Les figures 4.3 et 4.4 montre les concepts que nous avons créés en utilisant protege.
2. **Propriétés des objets (relations)** : La figure 4.5 montre les relations que nous avons créées en utilisant protege.
3. **Propriétés des données** : La figure 4.6 montre les propriétés des données que nous avons créées en utilisant protege.

### 4.7.3 Génération de fichiers RDF pour l’instance handicap

Nous avons généré un fichier RDF pour chaque instance de handicap en utilisant python. Ces fichiers peuvent être utilisés dans d’autres systèmes ou pour visualiser le graphique lié à un handicap. La figure 4.7 montre la structure d’un fichier généré

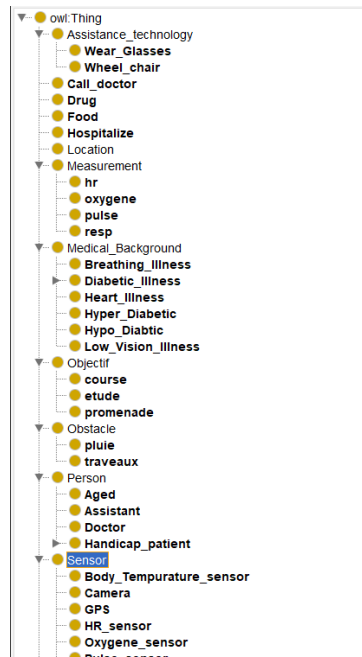


FIGURE 4.3 – initialisation des concepts

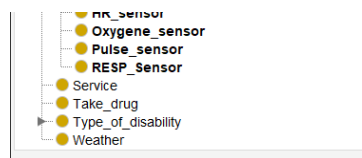


FIGURE 4.4 – initialisation des concepts

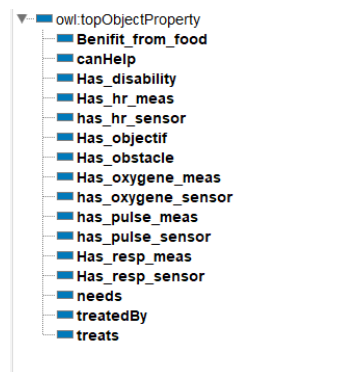


FIGURE 4.5 – initialisation des relations

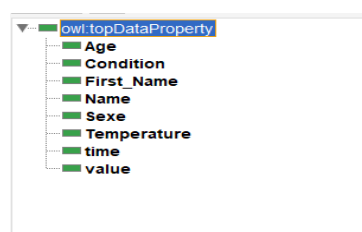


FIGURE 4.6 – initialisation des propriétés des données



```

<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:PFE="http://www.semanticweb.org/ahmedaminbadji/ontologies/2022/0/untitled-ontology-7#" >
  <rdf:Description rdf:about="http://www.semanticweb.org/ahmedaminbadji/ontologies/2022/0/untitled-ontology-7#Anthony_Jenkins">
  <rdf:type rdf:resource="http://www.semanticweb.org/ahmedaminbadji/ontologies/2022/0/untitled-ontology-7#Handicap_patient"/>
  <PFE:Age rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">78</PFE:Age>
  <PFE:First_Name rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Hugh</PFE:First_Name>
  <PFE:Name rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Lopez</PFE:Name>
  <PFE:Sexe rdf:datatype="http://www.w3.org/2001/XMLSchema#string">male</PFE:Sexe>
  <PFE:Has_disability>
  <rdf:Description rdf:about="http://www.semanticweb.org/ahmedaminbadji/ontologies/2022/0/untitled-ontology-7#vision">
  </rdf:Description>
  </PFE:Has_disability>
  <PFE:Has_objectif>
  <rdf:Description rdf:about="http://www.semanticweb.org/ahmedaminbadji/ontologies/2022/0/untitled-ontology-7#etude">
  </rdf:Description>
  </PFE:Has_objectif>
  <PFE:Has_obstacle>
  <rdf:Description rdf:about="http://www.semanticweb.org/ahmedaminbadji/ontologies/2022/0/untitled-ontology-7#travaux">
  </rdf:Description>
  </PFE:Has_obstacle>
  <PFE:has_hr_sensor>
  <rdf:Description rdf:about="http://www.semanticweb.org/ahmedaminbadji/ontologies/2022/0/untitled-ontology-7#HR_sensor">
  <PFE:Has_hr_meas>
  <rdf:Description rdf:about="http://www.semanticweb.org/ahmedaminbadji/ontologies/2022/0/untitled-ontology-7#Measurement0">
  <PFE:value rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">94</PFE:value>
  <PFE:time rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">0</PFE:time>
  </rdf:Description>
  </PFE:Has_hr_meas>
  </rdf:Description>
  </PFE:has_hr_sensor>
  <PFE:has_pulse_sensor>
  <rdf:Description rdf:about="http://www.semanticweb.org/ahmedaminbadji/ontologies/2022/0/untitled-ontology-7#Pulse_sensor">
  <PFE:has_pulse_meas>
  <rdf:Description rdf:about="http://www.semanticweb.org/ahmedaminbadji/ontologies/2022/0/untitled-ontology-7#Measurement1">
  <PFE:value rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">93</PFE:value>

```

FIGURE 4.7 – Exemple RDF généré

## 4.8 Présentation du système

Dans cette section, nous allons présenter les principales fonctions de notre système. Tout d’abord, nous devons enrichir l’ontologie avec des données telles que les patients, les médecins, les assistants, etc.

### 4.8.1 Lecture des informations

En raison du manque de jeux de données liés à notre travail, nous avons décidé de n’utiliser que les signes vitaux des patients handicapés pour faire fonctionner notre système. Dans cette section, nous allons vous montrer comment nous avons extrait les informations de l’ensemble de données BIDMC. L’ensemble de données contient plusieurs fichiers :

- • Fichier texte contenant l’âge et le sexe de la personne.
- • Fichier CSV contenant les signes vitaux que nous utiliserons pour enrichir notre système. Le fichier csv contient environ 400 secondes de mesures continues, mais nous n’utiliserons que la première. Comme le montre la figure, nous stockons les données dans un tableau pour les utiliser lorsque nous créons des individus.

### 4.8.2 Création des individus

Après avoir lu et stocké les informations, nous allons commencer à créer des individus. Il y a une certaine différence dans les modules de programmation qui utilisent les connaissances. Par exemple : en Java, nous devons travailler sur des fichiers RDF pour

```

260     filename = "bidmc_csv/bidmc_"+n+"_Numerics.csv"
261     with open(filename, 'r') as csvfile:
262         so = csv.reader(csvfile, delimiter=',', quotechar='')
263         so_data = list(so)
264         del so_data[0]

```

FIGURE 4.8 – Lecture des données collectées

chaque instance, mais en Python, le fichier OWL est suffisant pour faire fonctionner notre système. Nous avons donc décidé de générer les deux, les individus dans le fichier OWL et les fichiers RDF pour chaque instance.

- • Génération des noms des patients handicapés en utilisant le sexe que nous avons sauvegardé précédemment et création d'un individu en utilisant l'objet ontologique.
- • Générer les noms des docteurs des patients en utilisant le même genre que nous avons sauvegardé plus tôt et en créant l'individu en utilisant l'objet ontologique.
- • Générer les noms des assistants en utilisant également le même genre que nous avons enregistré précédemment et créer un individu en utilisant l'objet ontologique.
- • Ajout des relations entre le handicapé et son médecin et assistant
- • Sauvegarder l'ontologie

```

219     #GENERATE NAME FROM GENDER
220     name = names.get_full_name(gender=g)
221     fname_pat = name.split()[0]
222     name_pat = name.split()[1]

```

FIGURE 4.9 – Ajouter les individus a l'ontologie

```

223     #Creating Handicap instance
224     person = onto.Handicap_patient()
225     person.Name = [name_pat]
226     person.First_Name = [fname_pat]
227     person.Age = [int(age)]
228     person.Sexe = [g]

```

FIGURE 4.10 – Ajouter les individus a l'ontologie

```

230     #Creating Doctor of the handicap
231     name = names.get_full_name(gender=g)
232     fname = name.split()[0]
233     name = name.split()[1]
234     doctor = onto.Doctor()
235     doctor.Name = [name]
236     doctor.First_Name = [fname]
237     doctor.Sexe = [g]

```

FIGURE 4.11 – Ajouter les individus a l'ontologie

```

239     #Creating Assistant of the handicap
240     name = names.get_full_name(gender=g)
241     fname = name.split()[0]
242     name = name.split()[1]
243     assistant = onto.Assistant()
244     assistant.Name = [name]
245     assistant.First_Name = [fname]
246     assistant.Sexe = [g]

```

FIGURE 4.12 – Ajouter les individus a l'ontologie

```

248     #ADD RELATIONSHIPS
249     owl.AllDifferent([person, doctor,assistant])
250     person.needs = [assistant]
251     person.treatedBy = [doctor]
252

```

FIGURE 4.13 – Ajout des relations

### 4.8.3 Indexation des données

Cette section présente la méthode d'indexation proposée sur l'ensemble de données extraite du data set BIDMC ainsi que le temps d'exécution de la recherche.

#### La recherche dans l'arbre partie sémantique

voici une itération de recherche dans la première partie de notre structure arborescente qui est la partie de recherche sémantique ainsi que le temps d'exécution :

```

la recherche avec les caractéristique :
nom : Lopez type de disability : vision objectif : etude obstacle : travaux
La personne existe

Le temps de recherche dans la premiere partie
0.005357980728149414

```

FIGURE 4.14 – Recherche sémantique

### La recherche dans la partie de l'arbre binaire

La figure suivante montre une itération de recherche dans la deuxième partie de notre arbre-SB qui est basée sur le calculs des métriques :

```

existe
[88, 80, 41, 15, 98]
le chemin parcouru est :
['G', 'G', 'G', 'G', 'G', 'D']
Le temps de recherche dans la deuxieme partie est :
0.0041656494140625

```

FIGURE 4.15 – Recherche par métriques

### La recherche knn

Supposons que nous avons comme objectif la recherche d'une personne handicapé à travers le vecteurs des caractéristiques [age,hr,pulse,resp,o2] et les personnes handicapées qui sont proches a cette personne, donc nous avons utilisé comme solution la recherche knn, et la figure ci dessous montre le résultat de knn ainsi que les distances calculées :

```

La recherche knn
Enter K :
4
Les caractéristique similaires de cette personnes sont :
[[0.0, [88, 80, 41, 15, 98], 'Glover'], [0.399999999999999915, [57, 75, 74, 20, 98], 'Sanderson'], [1.199999999999999886, [57, 78, 76, 17, 100], 'Nelson'], [1.399999999999999915, [43, 90, 89, 11, 96], 'Palmer']]
Nombre de distances calculées
4

```

FIGURE 4.16 – Recherche knn

## 4.9 Conclusion

Dans ce chapitre, nous avons présenté les outils que nous avons utilisé pour implémenter le concept de IoT et Web Sémantique. Nous avons présenté par la suite les résultats de notre structure d'indexation. Nous avons évalué la recherche dans notre arbre-SB selon deux modes de recherche, et chaque mode contient deux parties. commençant tout d'abord par le premier mode qui est la recherche sémantique et binaire simple dont la première partie est la recherche par le vecteur des caractéristiques sémantiques [type d'handicap, objectif, obstacle, nom] et la deuxième partie par lecteurs des métrique [age,hr,pulse,resp,o2]. Ensuite la première étape du deuxième mode de recherche c'est la même, la différence donc est dans la deuxième partie au lieu d'utiliser une recherche simple on introduit la recherche knn. Par conséquent, l'arbre-SB a un impact très positif pour faire une optimisation en fonction du temps d'exécution.

# Conclusion générale et perspectives

Le travail que nous avons réalisé dans ce projet final consiste à combiner trois approches : "Internet des objets", "Web sémantique" et "Techniques d'indexation" au profit et à l'usage des personnes handicapées, plus précisément les chercheurs font des études pour aider ce type de personnes à travers les capteurs IoT qui posent le problème de l'hétérogénéité des données et leur grande taille qui rend la gestion et la mise à jour plus lente et plus difficile, Ils trouvent comme une solution l'introduction du Web sémantique pour résoudre le problème de l'hétérogénéité et les techniques d'indexation pour résoudre le problème du temps d'exécution. Notre système commence par la conception du modèle ontologique qui comporte les concepts, les relations, les règles SWRL et les requêtes sparql (en utilisant le protege) et par la suite la construction des structures d'indexation en utilisant le langage de programmation python. Suite au travail réalisé, quelques perspectives pourront être données, citons :

- Tester notre système sur une plate-forme réel avec des capteurs et on contact des médecins et les autres acteurs de système.
- Tester notre système on utilisant un data set plus riche et plus proche à la réalité.
- Évaluation de l'ontologie créée par les experts du domaine, ou par les personnes ciblées (personnes handicapées).
- En raison de manque de base de données de signaux vitaux à grande taille, dans le futur notre travaille peut tester avec une base de données générée ou une base de données concerne autre domaine.
- Dans le futur aussi nous pouvons utiliser autre algorithme différent de knn pour improuver les résultats.

# Bibliographie

- [BID, n.d.] *BIDMC PPG and Respiration Dataset v1.0.0*. [Online ; accessed 2022-06-19].
- [22, n.d.] *Internet des objets wikipedia* [https://fr.wikipedia.org/wiki/Internet\\_des\\_objets](https://fr.wikipedia.org/wiki/Internet_des_objets). [Online; accessed 2022-06-18].
- [unk, n.d.] *Les-differents-types-de-handicap - CCAH* <https://www.ccah.fr/CCAHA/Articles/Les-differents-types-de-handicap>. [Online ; accessed 2022-06-18].
- [Adrien, 2014] Adrien, FB. 2014. *Data Preprocessing and Other Improvements for the Multi-Dimensional PH-Index*. Ph.D. thesis, ETH Zurich Zürich, Switzerland.
- [ALLELE, 2021] ALLELE, IMANE. 2021. Une nouvelle structure d’indexation pour les données du processus de suivi des objets dans l’Internet des objets vidéo.
- [Antoshenkov, 1995] Antoshenkov, Gennady. 1995. Byte-aligned bitmap compression. *Page 476 of : Proceedings DCC’95 data compression conference*. IEEE.
- [Antoshenkov & Ziauddin, 1996] Antoshenkov, Gennady, & Ziauddin, Mohamed. 1996. Query processing and optimization in Oracle Rdb. *The VLDB Journal*, **5**(4), 229–237.
- [Athanasoulis *et al.*, 2016] Athanasoulis, Manos, Yan, Zheng, & Idreos, Stratos. 2016. Upbit : Scalable in-memory updatable bitmap indexing. *Pages 1319–1332 of : Proceedings of the 2016 International Conference on Management of Data*.
- [BENRAZEK, 2021] BENRAZEK, Ala-Eddine. 2021. *Internet of Things : Analysis of suspicious behaviour in a surveillance camera network*. Ph.D. thesis.
- [Benrazek *et al.*, 2020] Benrazek, Ala-Eddine, Kouahla, Zineddine, Farou, Brahim, Ferag, Mohamed Amine, Seridi, Hamid, & Kurulay, Muhammet. 2020. An efficient indexing for Internet of Things massive data based on cloud-fog computing. *Transactions on emerging telecommunications technologies*, **31**(3), e3868.
- [Bentley, 1979] Bentley, Jon Louis. 1979. Multidimensional binary search trees in database applications. *IEEE Transactions on Software Engineering*, 333–340.
- [Berchtold *et al.*, 1996] Berchtold, Stefan, Keim, Daniel A, & Kriegel, Hans-Peter. 1996. The X-tree : An index structure for high-dimensional data. *Pages 28–39 of : Very Large Data-Bases*.
- [Berners-Lee *et al.*, 2001] Berners-Lee, Tim, Hendler, James, & Lassila, Ora. 2001. The semantic web. *Scientific american*, **284**(5), 34–43.
- [Böhm *et al.*, 2001] Böhm, Christian, Berchtold, Stefan, & Keim, Daniel A. 2001. Searching in high-dimensional spaces : Index structures for improving the performance of multimedia databases. *ACM Computing Surveys (CSUR)*, **33**(3), 322–373.
- [Brin, 1995] Brin, Sergey. 1995. Near neighbor search in large metric spaces.
- [Chakrabarti & Mehrotra, 1999] Chakrabarti, Kaushik, & Mehrotra, Sharad. 1999. The hybrid tree : An index structure for high dimensional feature spaces. *Pages 440–447 of :*

*Proceedings 15th International Conference on Data Engineering (Cat. No. 99CB36337)*. IEEE.

- [Chan & Ioannidis, 1998] Chan, Chee-Yong, & Ioannidis, Yannis E. 1998. Bitmap index design and evaluation. *Pages 355–366 of : Proceedings of the 1998 ACM SIGMOD international conference on Management of data*.
- [Chatzimichail *et al.*, 2021] Chatzimichail, Angelos, Stathopoulos, Evangelos, Ntioudis, Dimos, Tsanoua, Athina, Rousi, Maria, Mavropoulos, Athanasios, Meditskos, George, Vrochidis, Stefanos, & Kompatsiaris, Ioannis. 2021. Semantic Web and IoT. *Pages 3–33 of : Semantic IoT : Theory and Applications*. Springer.
- [Chi, 2015] Chi, Lianhua. 2015. *Hashing for large-scale structured data classification*. Ph.D. thesis.
- [Ciaccia *et al.*, 1997] Ciaccia, Paolo, Patella, Marco, & Zezula, Pavel. 1997. M-tree : An efficient access method for similarity search in metric spaces. *Pages 426–435 of : Vldb*, vol. 97.
- [Domingue *et al.*, 2011] Domingue, John, Fensel, Dieter, & Hendler, James A. 2011. *Handbook of semantic web technologies*. Springer Science & Business Media.
- [Ferrag *et al.*, 2019] Ferrag, Mohamed Amine, Kouahla, Zineddine, Seridi, Hamid, & Kuralay, Muhammet. 2019. Big IoT Data Indexing : Architecture, Techniques and Open Research Challenges. *Pages 1–6 of : 2019 International Conference on Networking and Advanced Systems (ICNAS)*. IEEE.
- [Finkel & Bentley, 1974] Finkel, Raphael A, & Bentley, Jon Louis. 1974. Quad trees a data structure for retrieval on composite keys. *Acta informatica*, **4**(1), 1–9.
- [Gacitua *et al.*, 2019] Gacitua, Ricardo, Mazon, Jose N, & Cravero, Ania. 2019. Using Semantic Web technologies in the development of data warehouses : A systematic mapping. *Wiley Interdisciplinary Reviews : Data Mining and Knowledge Discovery*, **9**(3), e1293.
- [Germann *et al.*, 2009] Germann, Ulrich, Joanis, Eric, & Larkin, Samuel. 2009. Tightly packed tries : How to fit large models into memory, and make them load fast, too. *Pages 31–39 of : Proceedings of the Workshop on Software Engineering, Testing, and Quality Assurance for Natural Language Processing (SETQA-NLP 2009)*.
- [Guttman, 1984] Guttman, Antonin. 1984. R-trees : A dynamic index structure for spatial searching. *Pages 47–57 of : Proceedings of the 1984 ACM SIGMOD international conference on Management of data*.
- [Jean-Baptiste, 2021] Jean-Baptiste, Lamy. 2021. *Ontologies with Python : Programming OWL 2.0 Ontologies with Python and Owlready2*. Springer.
- [Joyce, 2020a] Joyce, PhD, John. 2020a (01). *AI IoT : A Powerful Combination*.
- [Joyce, 2020b] Joyce, PhD, John. 2020b. *AI IoT : A Powerful Combination* <https://code.visualstudio.com/docs>.
- [Joyce, 2020c] Joyce, PhD, John. 2020c. *AI IoT : A Powerful Combination* <https://www.python.org/doc/essays/blurb/>.
- [Karami & Rahimi, 2019] Karami, Mahtab, & Rahimi, Azin. 2019. Semantic web technologies for sharing clinical information in health care systems. *Acta Informatica Medica*, **27**(1), 4.

- [Katayama & Satoh, 1997] Katayama, Norio, & Satoh, Shin'ichi. 1997. The SR-tree : An index structure for high-dimensional nearest neighbor queries. *ACM Sigmod Record*, **26**(2), 369–380.
- [Kindberg *et al.*, 2002] Kindberg, Tim, Barton, John, Morgan, Jeff, Becker, Gene, Caswell, Debbie, Debaty, Philippe, Gopal, Gita, Frid, Marcos, Krishnan, Venky, Morris, Howard, *et al.* 2002. People, places, things : Web presence for the real world. *Mobile Networks and Applications*, **7**(5), 365–376.
- [Kouahla, 2013] Kouahla, Zineddine. 2013. *Indexation dans les espaces métriques Index arborescent et parallélisation*. Ph.D. thesis, Université de Nantes.
- [Kouahla *et al.*, 2019] Kouahla, Zineddine, Anjum, Adeel, Akram, Sheeraz, Saba, Tanzila, & Martinez, José. 2019. XM-tree : data driven computational model by using metric extended nodes with non-overlapping in high-dimensional metric spaces. *Computational and Mathematical Organization Theory*, **25**(2), 196–223.
- [Kouahla *et al.*, 2021] Kouahla, Zineddine, Benrazek, Ala-Eddine, Ferrag, Mohamed Amine, Farou, Brahim, Seridi, Hamid, Kurulay, Muhammet, Anjum, Adeel, & Asheralieva, Alia. 2021. A Survey on Big IoT Data Indexing : Potential Solutions, Recent Advancements, and Open Issues. *Future Internet*, **14**(1), 19.
- [Lord, 2010] Lord, Phillip. 2010. Components of an Ontology. *Ontogenesis*.
- [MacNicol & French, 2004] MacNicol, Roger, & French, Blaine. 2004. Sybase IQ multiplex-designed for analytics. *Pages 1227–1230 of : Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*.
- [Mao *et al.*, 2015] Mao, Rui, Xu, Honglong, Wu, Wenbo, Li, Jianqiang, Li, Yan, & Lu, Minhua. 2015. Overcoming the challenge of variety : big data abstraction, the next evolution of data management for AAL communication systems. *IEEE Communications Magazine*, **53**(1), 42–47.
- [Mehla & Jain, 2019] Mehla, Sonia, & Jain, Sarika. 2019. Rule languages for the semantic web. *Pages 825–834 of : Emerging Technologies in Data Mining and Information Security*. Springer.
- [Mokbel *et al.*, 2003] Mokbel, Mohamed F., Ghanem, Thanaa M., & Aref, Walid G. 2003. Spatio-temporal access methods. *IEEE Data Eng. Bull.*, **26**(2), 40–49.
- [O'Neil & Quass, 1997] O'Neil, Patrick, & Quass, Dallan. 1997. Improved query performance with variant indexes. *Pages 38–49 of : Proceedings of the 1997 ACM SIGMOD international conference on Management of data*.
- [Prasad *et al.*, 2021] Prasad, Jayashree R, Shelke, Priya M, & Prasad, Rajesh S. 2021. Semantic Web Technologies. *Pages 35–57 of : Semantic IoT : Theory and Applications*. Springer.
- [Rey, 2022] Rey. 2022. *De l'histoire aux différentes typologies de handicap - 2ème partie*.
- [Rose *et al.*, 2015] Rose, Karen, Eldridge, Scott, & Chapin, Lyman. 2015. The internet of things : An overview. *The internet society (ISOC)*, **80**, 1–50.
- [Sprenger *et al.*, 2019] Sprenger, Stefan, Schäfer, Patrick, & Leser, Ulf. 2019. BB-Tree : A practical and efficient main-memory index structure for multidimensional workloads. *Pages 169–180 of : EDBT*.
- [Srinivasan & Carey, 1991] Srinivasan, Venkathachary, & Carey, Michael J. 1991. Performance of B-tree concurrency control algorithms. *Pages 416–425 of : Proceedings of the 1991 ACM SIGMOD international conference on Management of data*.



- [Srinivasan & Carey, 1993] Srinivasan, Venkathachary, & Carey, Michael J. 1993. Performance of B+ tree concurrency control algorithms. *The VLDB Journal*, **2**(4), 361–406.
- [Traina *et al.*, 2000] Traina, Caetano, Traina, Agma, Seeger, Bernhard, & Faloutsos, Christos. 2000. Slim-trees : High performance metric trees minimizing overlap between nodes. *Pages 51–65 of : International Conference on Extending Database Technology*. Springer.
- [Traina *et al.*, 2002] Traina, Caetano, Traina, Agma, Faloutsos, Christos, & Seeger, Bernhard. 2002. Fast indexing and visualization of metric data sets using slim-trees. *IEEE Transactions on Knowledge and Data Engineering*, **14**(2), 244–260.
- [Vancea, 2015] Vancea, Bogdan Aurel. 2015. *Cluster-computing and parallelization for the multi-dimensional ph-index*. M.Phil. thesis, ETH Zurich.
- [Vermesan *et al.*, 2014] Vermesan, Ovidiu, Friess, Peter, *et al.* 2014. *Internet of things-from research and innovation to market deployment*. Vol. 29. River publishers Aalborg.
- [Wu *et al.*, 2002] Wu, Kesheng, Otoo, Ekow J, & Shoshani, Arie. 2002. Compressing bitmap indexes for faster search operations. *Pages 99–108 of : Proceedings 14th international conference on scientific and statistical database management*. IEEE.
- [Wu *et al.*, 2016] Wu, Yinjun, Chen, Zhen, Wen, Yuhao, Zheng, Wenxun, & Cao, Junwei. 2016. Combat : A new bitmap index coding algorithm for big data. *Tsinghua Science and Technology*, **21**(2), 136–145.
- [Zäschke *et al.*, 2014] Zäschke, Tilmann, Zimmerli, Christoph, & Norrie, Moira C. 2014. The PH-tree : a space-efficient storage structure and multi-dimensional index. *Pages 397–408 of : Proceedings of the 2014 ACM SIGMOD international conference on Management of data*.