

## **MÉMOIRE DE FIN D'ETUDES**

Pour l'obtention du diplôme

## **MASTER EN INFORMATIQUE**

Spécialité : Système d'information

### *Thème*

---

*Catégorisation des services web sémantique avec  
K-means*

---

*Réalisé par :*  
**Khelifi Abdelkarim**

*Encadré par :*  
**Mr Kaouan Moussa**

*Année universitaire : 2018 ~ 2019*

# Résumé

---

Les services Web sont des technologies émergentes et prometteuses pour le développement, le déploiement et l'intégration d'applications Internet. Ils sont basés sur trois briques principales qui sont SOAP (Simple Object Access Protocol), WSDL (Web Service Description Language) et UDDI (Universal Description, Discovery and Integration). Le langage utilisé qui sous-tend ces protocoles est XML (eXtensible Markup Language), ce qui rend les Web Services indépendants des plates-formes et des langages de programmation. Ils sont devenus un moyen très efficace dans l'interopérabilité des systèmes. Le besoin d'introduire la sémantique dans les services Web se fait sentir, afin d'automatiser les différentes phases de leur cycle de vie, en l'occurrence la phase de découverte.

Le concept des services Web sémantiques, est le fruit de la convergence du domaine des services web avec le Web sémantique, En effet, son ultime objectif est de rendre les services web plus accessibles à la machine en automatisant les différentes tâches qui facilitent leur utilisation. Dans ce travail, on étudie la problématique de classification des services web sémantique avec l'algorithme k-means.

**Mots-Clés** : Service Web sémantique, classification des services web sémantique.

# Abstract

---

Web services are emerging and promising technologies for the development, deployment and integration of Internet applications. They are based on three main bricks: Simple Object Access Protocol (SOAP), Web Service Description Language (WSDL), and Universal Description, Discovery and Integration (UDDI). The language used behind these protocols is XML (eXtensible Markup Language), which makes Web Services independent of platforms and programming languages. They have become a very effective means of interoperability of systems. The need to introduce semantics into web services is felt in order to automate the different phases of their life cycle, in this case the discovery phase.

The concept of semantic Web services is the result of the convergence of the domain of web services with the semantic Web, In fact, its ultimate objective is to make web services more accessible to the machine by automating the different tasks that facilitate their use. In this work, we study the problem of classification of semantic web services with the k-means algorithm.

**Keywords:** Semantic Web Service, classification of semantic web services.

---

# *Remerciement*

*Je tiens à remercier :*

*Allah le tout puissant*

*J'exprime toute ma reconnaissance à l'égard de mon encadreur Mr Kaouan moussa pour ses conseils, ses encouragements et pour la confiance qu'il m'a témoigné au cours de ce travail. C'est grâce à lui que j'ai pu mener à bien ce travail.*

*Les jurys pour leurs efforts et leur soin apporté à notre travail. Aux enseignants de notre université et département de mathématiques et informatique*

*Je tiens aussi à remercier également tous mes collègues et amis pour leur soutien.*

*Khelifi Abdelkarim*

---

# *Dédicace*

*Je dédie ce mémoire :*

*A Ceux qui ont fait de moi l'homme que je suis  
aujourd'hui : mes très chers parents, que dieu les  
récompense et les garde.*

*Mes chers frères et mes sœurs et toute la famille.*

*A mes collègues et mes amis.*

*Khelifi Abdelkarim*

---

# Table des matières

<b>Introduction générale.....</b>	<b>1</b>
<b>Chapitre I : Architecture Orientée Services .....</b>	<b>3</b>
<b>1. Introduction.....</b>	<b>4</b>
<b>2. Architecture Orientée Services : .....</b>	<b>4</b>
<b>2.1 Définition (SOA) .....</b>	<b>4</b>
<b>2.2 Concepts de base d'une architecture orientée services .....</b>	<b>4</b>
<b>3. Conclusion .....</b>	<b>5</b>
<b>Chapitre II : Les services web .....</b>	<b>6</b>
<b>1. Introduction.....</b>	<b>7</b>
<b>2. Principe de fonctionnement de Web-Services.....</b>	<b>7</b>
<b>2.1 Le fournisseur de service (serveur) .....</b>	<b>7</b>
<b>2.2 L'annuaire de service (Annuaire UDDI) .....</b>	<b>7</b>
<b>2.3 Le consommateur de service (Client) .....</b>	<b>7</b>
<b>3. Architecture de Web-Services .....</b>	<b>8</b>
<b>4. Les technologies utilisées .....</b>	<b>9</b>
<b>4.1 SOAP (Simple Object Access Protocol).....</b>	<b>9</b>
<b>4.2 WSDL (Web Services Description Langage).....</b>	<b>11</b>
<b>4.2.1 Structure d'un document WSDL Un document WSDL .....</b>	<b>12</b>
<b>4.2.2 Eléments de WSDL .....</b>	<b>12</b>
<b>4.3 UDDI (Universal Description Discovery and Integration).....</b>	<b>13</b>
<b>4.3.1 Présentation.....</b>	<b>13</b>
<b>4.3.2 L'annuaire UDDI est consultable de différentes manières : .....</b>	<b>13</b>
<b>5. Conclusion .....</b>	<b>14</b>
<b>Chapitre III : Les services web sémantique .....</b>	<b>15</b>
<b>1. Introduction.....</b>	<b>16</b>
<b>2. Langages de modélisation des services web .....</b>	<b>17</b>
<b>2.1 Approche de description à base d'annotations.....</b>	<b>17</b>
<b>2.2 Approches basées sur des langages sémantiques.....</b>	<b>18</b>
<b>3. Conclusion .....</b>	<b>21</b>
<b>Chapitre IV :classification .....</b>	<b>22</b>
<b>1. Introduction.....</b>	<b>23</b>
<b>2. Domaines d'application de classification : .....</b>	<b>23</b>
<b>3. Critères pour une bonne classification.....</b>	<b>24</b>

---

3.1 Validité .....	24
3.2 Interopérabilité.....	24
3.3 Stabilité.....	24
3.4 D'autre critère .....	24
4. Classification.....	25
4.1 Classification supervisée .....	25
4.2 Classification non supervisée .....	26
4.2.1 Définition .....	26
4.2.2 Méthodes non hiérarchiques .....	27
4.2.3 Méthodes hiérarchiques .....	31
5. Conclusion .....	35
<b>Chapitre V : Conception.....</b>	<b>36</b>
1. Introduction.....	37
2. Présentation d'UML .....	37
3. Diagrammes UML utilisé pour la conception de notre application .....	38
<b>Chapitre VI : IMPLEMENTATION .....</b>	<b>41</b>
1. Introduction.....	42
2. Outils et environnement de développement .....	42
3. Matching et mesure de similarité .....	42
4. Corpus utilisés pour la découverte.....	43
5. Présentation de L'application réalisée :.....	44
6. Exemple illustratif : .....	49
7. Conclusion .....	50
<b>Conclusion Générale .....</b>	<b>51</b>

---

## Liste De Figure :

Figure 1 : Modèle d'interaction des Web-Services .....	8
Figure 2 : Pile des Web-Services .....	9
Figure 3 : Structure d'un message SOAP .....	11
Figure 4 : L'ontologie supérieure de OWL-S.....	19
Figure 5 : Ontologie de Service Profile.....	21
Figure 6 : Les deux types de clustering hiérarchique/non hiérarchique.....	27
Figure 7 : La partition hiérarchique. ....	31
Figure 8: Diagramme de cas d'utilisation. ....	38
Figure 9: Diagramme d'activité.....	39
Figure 10 : Interface principale de la classification des SWS avec k_means .....	44
Figure 11 : Charger fichier owls.....	45
Figure 12 : Afficher les services.....	46
Figure 13 : Saisir le nombre de classe .....	46
Figure 14 : Saisir les centres des classes. ....	47
Figure 15 : Choisir le seuil. ....	47
Figure 16 : Choisir la métrique. ....	48
Figure 17: Les résultats de la classification en 3 classes. ....	48

# Introduction générale

---

## Introduction générale

- **Contexte**

Les services Web sont une technologie émergente. Ce sont des composants logiciels qui fournissent des fonctionnalités accessibles via des protocoles standardisés du Web (XML, UDDI, SOAP,...) les services Web ont marqué les méthodes actuelles d'ingénierie web, et sont universellement pris en charge par les fournisseurs informatiques et les utilisateurs.

Découverte des services Web est d'un grand intérêt et est un domaine fondamental de la recherche en informatique. De nombreux chercheurs ont mis l'accent sur la découverte de services Web via un système centralisé. Bien que les registres centralisés puissent fournir des méthodes plus au moins efficaces pour la découverte de services Web, mais ils souffrent d'un nombre de problèmes classiques connus dans les systèmes centralisés comme le seul point centralisé (point d'échec), le coût élevé de la maintenance et la dégradation des fonctionnalités. Actuellement, et en conséquence de la demande croissante d'utilisation des technologies services web, une série de questions concernant les méthodes, les architectures et les procédures de découverte de services web se pose.

- **Problématique**

Comme les services Web ont commencé à se développer à travers l'Internet, les utilisateurs doivent être en mesure d'accéder et de partager efficacement les services Web. La production et l'interopérabilité d'un plus grand nombre de services web ont conduit à l'apparition de nouvelles besoins sur la façon dont les services peuvent être publiés, découverts, ou utilisés. Par conséquent, des mécanismes sont nécessaires pour une sélection efficace de l'instance du service Web approprié en termes de facteurs, de qualité et de performance.

- **Objectif**

L'objectif de notre travail est de regrouper les services Web. Ce regroupement organisationnel va réduire l'espace de travail. En conséquence, nous allons améliorer la découverte en terme de temps de réponse, et améliorer le stockage en terme d'espace, et ce afin de surmonter le problème de centralisation des UDDIs. Dans ce contexte, les performances de la qualité de ressemblance (similarité) restent toujours essentielles.

# Introduction générale

---

- **Organisation de mémoire**

Le mémoire est organisé en six chapitres, dans le premier chapitre nous présentons l'architectures orientées services. Ensuite, nous présentons, dans le deuxième chapitre, la technologie des services Web avec leurs principaux standards. Le chapitre trois résume les objectifs visés par les services web sémantique ainsi que les différents travaux et langages qui sont utilisés pour formuler la sémantique pour les services web. Le chapitre quatre est consacré à l'état de l'art sur la classification, parmi lesquels nous citons les principaux algorithmes qui étudient ce problème, comme par exemple : k-means. Le chapitre cinq présente le coté conceptuel qui constitue une étape fondamentale avant l'implémentation. Le chapitre six présente la mise en œuvre de notre implémentation en décrivant notre environnement. Nous montrons également un exemple illustratif de classification de services web sémantique par algorithme k-means. Et enfin nous terminons par une conclusion.

# **CHAPITRE I : ARCHITECTURE ORIENTÉE SERVICES**

## 1. Introduction

Le système d'information de l'entreprise est généralement constitué d'applications et de données constituant son héritage. Avec les fusions de groupe, l'évolution des technologies, cet héritage a tendance à devenir hétérogène et à se spécialiser par métier (entité, service, etc.), ce qui provoque un fonctionnement en silo, c'est-à-dire un cloisonnement des différents métiers empêchant certaines formes de transversalité et masquant au décideur une vision globale du système d'information de son entreprise.

L'intégration des applications de l'entreprise (EAI) est une solution à ce problème. Elle consiste à développer des connecteurs spécifiques pour faire communiquer entre eux les différents silos de l'entreprise. [1]

L'objectif d'une architecture orientée services est de décomposer une fonctionnalité en un ensemble de fonctions basiques, appelées services, fournies par des composants et de décrire finement le schéma d'interaction entre ces services.

## 2. Architecture Orientée Services :

### 2.1 Définition (SOA)

Une architecture orientée services est essentiellement un ensemble de services. Ces services communiquent entre eux. La communication peut impliquer soit un simple transfert de données, soit deux ou plusieurs services coordonnant une activité. Quelques moyens de connecter les services entre eux sont nécessaires.

Les architectures orientées services ne sont pas nouvelles. Par le passé, la première architecture orientée services pour de nombreuses personnes consistait à utiliser DCOM ou de ORB (Object Request Brokers) basés sur la spécification CORBA. [2]

### 2.2 Concepts de base d'une architecture orientée services

Une architecture orientée service n'as pas une spécification officielle mais on peut donner les concepts de base pour son fonctionnement :

#### Les services :

C'est le concept de base, ça constitue des applications auto-contenues et indépendantes de la plateforme, qui supportent une composition rapide et facile d'applications logicielles distribuées et faiblement couplées avec un moindre coût. [3]

Les fournisseurs de services et leurs consommateurs possèdent des questions sur lesquelles une SOA doit répondre Ces questions peuvent être regroupées en trois catégories :

**La description du service :**

Constituant la réponse aux questions : «Comment le représenter ?», «Que fait-il ? ». Le principal format de description de services est WSDL (Web Services Description Language), normalisé par le W3C.

**La publication et la découverte des services :**

Constituant la réponse aux questions: «Comment le publier ? », «Où l'héberger ?» et « Où puis-je le trouver ? ». L'architecture doit répondre à la première question en publiant dans un annuaire les services disponibles, et à la dernière question en offrant la possibilité de rechercher un service parmi ceux qui ont été publiés. Le principal standard utilisé est UDDI (Universal Description Discovery and Integration).

**L'invocation des services :**

Constituant la réponse aux questions : « Comment pourrai-je l'utiliser ? » et « Comment le réaliser ? » représentant la connexion et l'interaction du client avec le service. Le principal protocole utilisé pour l'invocation de services est SOAP (Simple Object Access Protocol). Dans un système d'information il existe trois grandes catégories de services [4].

**3. Conclusion**

Dans ce chapitre, nous avons présenté l'architecture SOA de manière abstraite : définition et Concepts de base. Dans le chapitre qui se suit, nous présentons l'aspect service web l'architecture concrète du SOA. Dans lequel, nous exposons sa définition, ses standards recommandés et son architecture.

## **CHAPITRE II : LES SERVICES WEB**

## 1. Introduction

Un Web-Service est un système logiciel destiné à supporter l'interaction entre ordinateurs sur un réseau. Il a une interface décrite en un format pouvant être traité par ordinateur (e.g. WSDL). D'autres systèmes interagissent avec le Web-Service d'une manière prescrite par sa description en utilisant des messages SOAP, typiquement transmis avec le protocole http et une sérialisation XML, en conjonction avec d'autres standards relatifs au web. [5]

IBM définit les Web-Services antan qu'une nouvelle vague des applications Web. Ce sont des applications modulaires, auto-contenues et auto descriptives. Les Web services effectuent des actions allant de simples requêtes à des processus métiers complexes. Une fois qu'un Web service est déployé, d'autres applications (y compris des Web services) peuvent le découvrir et l'invoquer.

## 2. Principe de fonctionnement de Web-Services

Le modèle de fonctionnement des Web-Services est principalement composé de trois intervenants en interaction (voir figure1).

### 2.1 Le fournisseur de service (serveur)

Un fournisseur de services est la source de la fonctionnalité des services. Un fournisseur crée le Web-Service, puis publie un contrat d'interface ainsi que les informations d'accès au service, dans un annuaire de Web-Services.

### 2.2 L'annuaire de service (Annuaire UDDI)

Un annuaire de services est un registre des services disponibles. Chaque fournisseur publie dessus son contrat d'interface avec les informations à utiliser pour localiser le service.

Le client recherche les services appropriés dans l'annuaire et extrait le contrat d'interface correspondant (où et comment trouver un service).

### 2.3 Le consommateur de service (Client)

Le client accède à l'annuaire de service pour effectuer une recherche afin de trouver le service désiré. Lorsque il le trouve, il extrait le contrat d'interface correspondant et il l'utilise pour comprendre comment (méthodes disponibles) et où (adresse) accéder au service. Ensuite, il se lie au fournisseur pour invoquer le service.

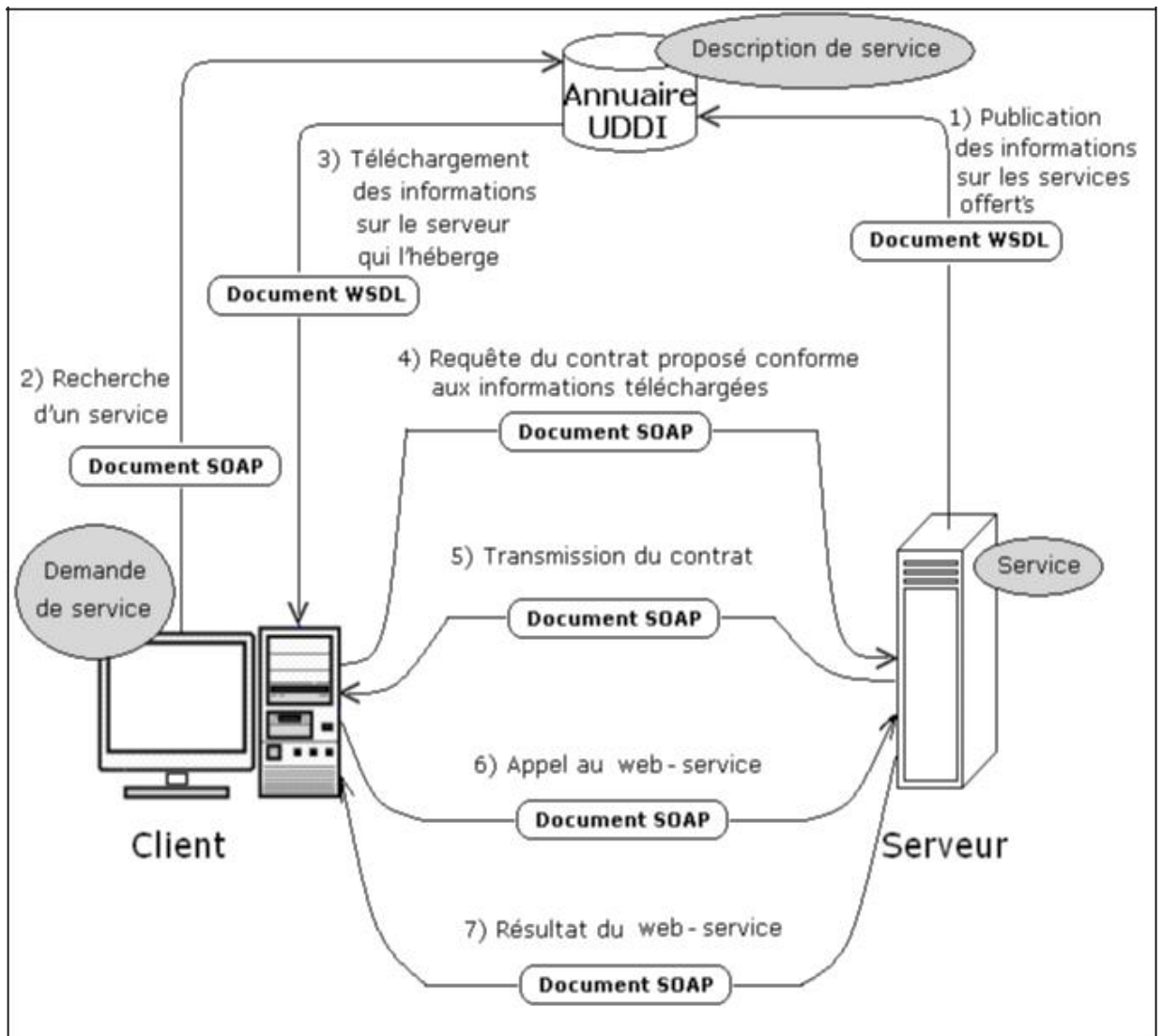


Figure 1 : Modèle d'interaction des Web-Services

### 3. Architecture de Web-Services

Différentes extensions de l'architecture de référence ont été proposées dans la littérature. Le groupe architecture du W3C travaille activement à l'élaboration d'une architecture étendue standard. Une architecture étendue est constituée de plusieurs couches se superposant les unes sur les autres, d'où le nom de pile des Web-services.

La figure qui suit décrit un exemple d'une telle pile (voir figure2). La pile est constituée de plusieurs couches, chaque couche s'appuyant sur un standard particulier. On retrouve, au-dessus de la couche de transport, trois couches s'appuyant sur les standards émergents SOAP, WSDL et UDDI. L'infrastructure de Web-Service de base définit les fondements techniques permettant de rendre les processus métiers (business processes) accessibles à l'intérieur d'une

entreprise et au-delà même des frontières d'une entreprise. Dans ce contexte deux types de couches permettent de la compléter :

Les couches dites transversales (exemple: sécurité, administration, transactions et qualité de services (QoS)) qui rendent viable l'utilisation effective des Web-Services dans le monde industriel :

Une couche Business processus qui permet l'utilisation effective des Web-Services dans le domaine du e-business.

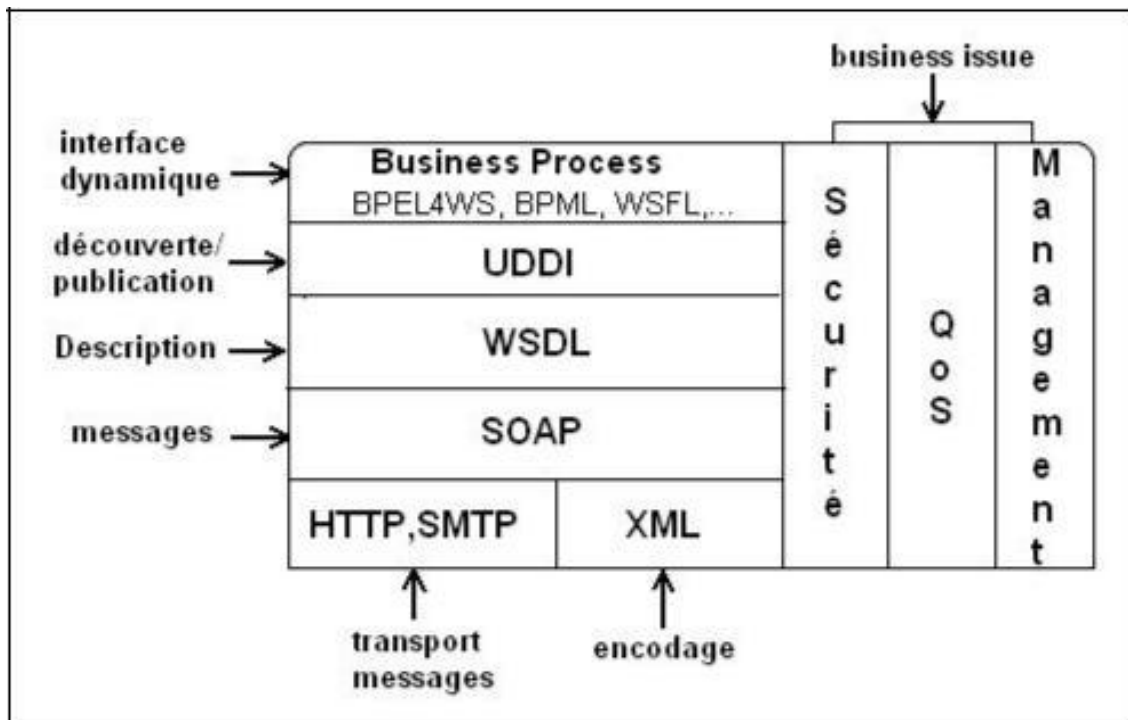


Figure 2 : Pile des Web-Services

#### 4. Les technologies utilisées

Les Web-Services utilisent trois technologies :

- **SOAP** pour le service d'invocation : il permet l'échange de messages dans un format particulier.
- **WSDL** pour le service de description : il permet de décrire les Web Services.
- **UDDI** pour le service de publication : il permet de référencer les Web Services.

##### 4.1 SOAP (Simple Object Access Protocol)

SOAP est un protocole de communication entre applications fondé sur XML, visant à satisfaire un double objectif : servir de protocole de communication sur les intranets, dans une optique d'intégration d'applications d'entreprise, et permettre la communication entre applications et Web-Services, en particulier dans un contexte d'échanges interentreprises.

Conçu pour s'appuyer principalement sur HTTP, SOAP ne requiert pas de modification majeure aux serveurs Web déjà installés sur la Toile, tout au plus des extensions. SOAP permet également, grâce au support de XML Schéma, de s'interfacer avec des applications préexistantes en capturant leurs structures de données quelle que soit la complexité de ces dernières [6].

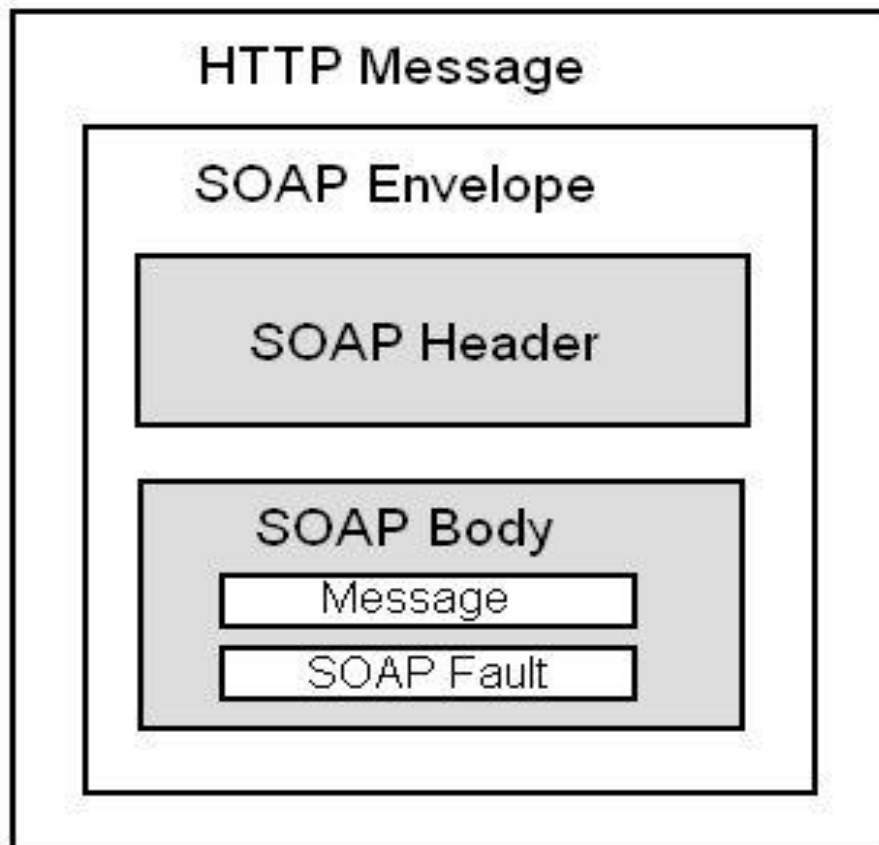
#### **4.1.1 La spécification SOAP se divise en trois parties :**

- La structure de l'enveloppe SOAP définit une structure générale pour exprimer le contenu d'un message, le responsable de son traitement et le caractère facultatif ou obligatoire de ce message.
- Les règles de codage de SOAP définissent un mécanisme de sérialisation qui peut être utilisé pour échanger des instances de types de données définis par l'application.
- La représentation RPC de SOAP définit une convention qui peut être utilisée pour représenter les appels de procédures à distance et leurs réponses.

#### **4.1.2 Structure du Message SOAP**

Un message SOAP comporte plusieurs parties (voir figure3):

- Une enveloppe qui définit la structure du message.
- Un en-tête (optionnel) qui contient les informations d'en-tête (autorisations et transactions par exemple).
- Un corps contenant les informations sur l'appel et la réponse une gestion d'erreur qui identifie la condition d'erreur des attachements ou pièces jointes (optionnel).



**Figure 3 : Structure d'un message SOAP**

#### **4.2 WSDL (Web Services Description Language)**

WSDL est un fichier qui spécifie ce que doit contenir un message de requête et l'apparence du message de réponse dans une notation sans ambiguïté.

La notation utilisée par un fichier WSDL pour décrire les formats de messages est basé sur la norme du schéma XML, ce qui signifie que WSDL est à la fois neutre par rapport au langage de programmation et à la plateforme.

Outre la description du contenu des messages, WSDL définit l'endroit où le service est disponible et le protocole de communications utilisé pour converser avec le service. Cela signifie que le fichier WSDL définit tout ce qui est nécessaire pour écrire un programme fonctionnant avec un Web-Service.

- Le WSDL décrit quatre ensembles de données importantes: Information d'interface décrivant toutes les fonctions disponibles publiquement.
- Information de type de données pour toutes les requêtes de message et requêtes de réponse.
- Information de liaison sur le protocole de transport utilisé.

- Information d'adresse pour localiser le service spécifié.

Une fois qu'un Web-Service est développé, il faut publier sa description WSDL et faire un lien vers elle dans un annuaire UDDI de sorte que les utilisateurs potentiels puissent le trouver. Quand un utilisateur souhaite utiliser le service, il fait une demande de son fichier WSDL afin de connaître son emplacement, les appels de fonctions et comment y accéder. A partir de cela, il peut composer, par exemple, une requête SOAP (Simple Object Access Protocol) vers l'ordinateur du service pour consommer le service.

#### 4.2.1 Structure d'un document WSDL Un document WSDL

Est constitué d'un ensemble de définitions :

```
< ?xml version="1.0" encoding="utf-8"? >
<définitions>
<types>!-- Les types de données </types>
<message>!-- La structure du message </message>
<portType>!--L'interface d'accès au sw </portType>
<binding>! -- Méthode d'accès </binding>
<service>! -- Le fournisseur du service </service>
</definitions>
```

#### 4.2.2 Eléments de WSDL

- **Définitions :**

Élément racine du document, il donne le nom du service, déclare les espaces de noms utilisés, et contient les éléments du service.

- **Message :**

Décrit un message unique, que ce soit un message de requête seul ou un message de réponse seul. L'élément définit les noms et types d'un ensemble de champs à transmettre.

- **Type :**

Décrit tous les types de données utilisés entre le client et le serveur. Contient les définitions de types utilisant un système de typage. Les Schémas XML sont utilisés pour définir les types de données.

- **Types de port (port Type) :**

Combine plusieurs éléments message pour composer une opération. Chaque opération se réfère à un message en entrée et à des messages en sorti.

- **Liaison (binding) :**

Défini les spécifications concrètes de la manière dont le service sera implémenté: protocole de communication et format des données pour les opérations et messages définis par un type de port particulier.

- **Service :**

Défini les adresses permettant d'invoquer le service donné, ce qui sert à regrouper un ensemble de ports reliés.

### 4.3 UDDI (Universal Description Discovery and Integration)

#### 4.3.1 Présentation

UDDI est une technologie d'annuaire basée sur XML et plus particulièrement destinée aux Web-Services, notamment dans le cadre d'architectures de type SOA (Service Oriented Architecture).

Un annuaire UDDI permet de localiser sur le réseau le Web-Service recherché. Il repose sur le protocole de transport SOAP.

Né en 1999 de l'initiative d'un certain nombre d'entreprises, dont SUN, Oracle, Microsoft, HP ou encore SAP, il permet de stocker à la fois des informations techniques et formelles tel que l'adresse pour accéder au Web-Services, mais également des informations beaucoup plus contextuelles, tel le nom de la personne qui s'occupe de leur gestion, la description sommaire de leur fonctionnalités ou encore le nom et la branche d'activité de l'entreprise dont ils dépendent.

#### 4.3.2 L'annuaire UDDI est consultable de différentes manières :

- **Les pages blanches :** comprennent la liste des entreprises ainsi que des informations associées à ces dernières. Nous y retrouvons donc des informations comme le nom de l'entreprise, ses coordonnées, la description de l'entreprise mais également l'ensemble de ses identifiants.
- **Les pages jaunes :** recensent les Web-Services de chacune des entreprises sous le standard WSDL.
- **Les pages vertes :** fournissent des informations techniques Précises sur les services fournis. Ces informations concernent les descriptions de services et de information de liaison ou encore les processus métiers associés.

## **5. Conclusion**

Dans ce chapitre, on a défini l'architecture des services web et les différents protocoles utilisés tel que WSDL pour la description de services et UDDI pour la publication et la découverte de services. Ces protocoles reposent sur une architecture orientée services(SOA). Les limites des services web classiques ont permis aux chercheurs d'exploiter les ontologies pour les rendre plus expressives pouvant par conséquent d'automatiser leurs fonctions (découverte, composition,.. etc.). Dans le chapitre suivant, nous présenterons plus en détails les services web sémantique.

# **CHAPITRE III : LES SERVICES WEB SÉMANTIQUE**

## 1. Introduction

Les services Web sémantiques se situent à l'intersection de deux domaines de recherche importants qui concernent les technologies de l'internet : le Web sémantique et les services Web. Le Web sémantique s'intéresse principalement aux informations statiques disponibles sur Internet (pages Web) et à la manière de les décrire d'une façon compréhensible par les machines. D'autre part, les standards permettant la description des services Web tels que WSDL et UDDI qui utilisent des structures XML pour la description de la fonctionnalité des services et des données qu'ils échangent. De telles structures sont syntaxiques et engendrent une forme d'ambiguïté. Ainsi la même structure XML peut être interprétée différemment d'un utilisateur à un autre. Cette contrainte empêche l'utilisation des services Web par des agents logiciels. Le même problème s'était posé quelques années plus tôt pour les pages Web, et avait engendré l'apparition des solutions pour l'annotation sémantique des pages Web.

De nombreux travaux ont tentés d'adapter ces solutions pour l'annotation sémantique des services Web.

Un service web sémantique est un service web décrit en utilisant des annotations sémantiques dans un langage bien défini (exemple : les ontologies) qui permettent au service d'avoir une interface interprétable sans ambiguïté, facilitant l'automatisation de certaines tâches telles que leurs découverte, sélection, invocation et leurs composition.

- ***Découverte automatique de services web*** : actuellement cette tâche doit être réalisée par un utilisateur humain qui doit utiliser un moteur de recherche ou un annuaire pour trouver le service, lire la page Web qui décrit l'utilité et l'utilisation du service, puis l'exécuter manuellement pour vérifier que celui-ci correspond bien aux attentes de l'utilisateur.
- ***Invocation automatique de services web*** : l'invocation automatique d'un service signifie l'exécution du service par un programme informatique ou un agent logiciel. Cet agent doit être capable d'interpréter les descriptions de services afin de délivrer les données nécessaires à l'exécution du service Web.
- ***Composition automatique de services web*** : l'objectif que l'utilisateur veut atteindre nécessite souvent l'utilisation de plusieurs services Web. L'agent logiciel chargé d'atteindre cet objectif doit disposer de suffisamment de données afin de pouvoir sélectionner, composé et inter opérer automatiquement ces services web.

## 2. Langages de modélisation des services web

De nombreux langages et approches ont été développés dans l'objectif de décrire les services Web sémantiques, les plus connus sont WSDL-S et OWL-S (que nous allons utiliser pour l'adaptation de notre algorithme), et le langage WSMO [7].

### 2.1 Approche de description à base d'annotations

WSDL-S est un langage de description sémantique des services Web. Une description WSDL-S est une description WSDL augmentée avec l'annotation sémantique. Les annotations sémantiques peuvent être des références à des concepts définis dans des ontologies externes. WSDL-S est issu du projet METEOR-S de l'université de Georgia. Il définit un modèle sémantique pour capturer les termes et les concepts utilisés pour décrire et représenter la connaissance, cette sémantique est ajoutée en deux étapes [8] :

La première étape consiste à faire référence, dans la partie définition de WSDL, à une ontologie dédiée au service à publier ;

La deuxième étape consiste à annoter les opérations de la définition WSDL avec la sémantique, en ajoutant deux nouvelles balises ; la balise « Action » qui permet de représenter l'action de l'opération et la balise « Contrainte » qui représente les pré et post conditions d'une opération.

Quatre rôles du modèle sémantique sont distingués :

***InputSemantics*** : La sémantique des paramètres d'entrées ;

***OutputSemantics*** : La sémantique des paramètres de sortie ;

***Precondition*** : Un ensemble d'états sémantiques qui doivent être vrais afin d'invoquer une opération avec succès ;

***Effect*** : Un ensemble d'états sémantiques qui doivent être vrais après qu'une opération accomplisse son exécution.

### SAWSDL

Semantic Annotation for WSDL and XML Schema (SAWSDL) [9] est une recommandation du W3C depuis 2007. Contrairement à OWL-S ou WSMO, le SAWSDL ne définit pas un nouveau langage ou une ontologie de haut niveau pour la description sémantique des Web services. Cependant, Il fournit un mécanisme à travers lequel des concepts appartenant à des modèles sémantiques existants peuvent être référés à partir d'un document WSDL 2.0. SAWSDL utilise l'extensibilité de WSDL 2.0 permettant d'ajouter des attributs aux éléments existants. Ces mécanismes permettent l'annotation sémantique des éléments de WSDL (les structures des messages d'entrées et des sorties, les opérations et les interfaces) et des schémas XML. Les mécanismes d'annotation, ainsi définis, sont indépendants du langage de

représentation de la sémantique (comme OWL et WSML) [9]. SAWSDL est la suite de WSDL-S [10] dans lequel les éléments « precondition » et « effet » sont éliminés. L'élément « pre-condition » indique les pré-conditions à vérifier pour que l'opération s'exécute comme prévu et « effet » représente les effets de cette exécution [11].

SAWSDL définit 3 extensions principales pour annoter un document WSDL 2.0. La première est « model Reference » qui permet d'associer un élément WSDL ou XML Schema à un concept d'un modèle sémantique. Cette extension figure comme attribut de « interface », « opération » et « Fault » de la description WSDL. Elle permet, aussi, l'annotation des schémas XML (les définitions des types de données simples ou complexes, les déclarations d'éléments, etc.). Chaque « model Reference » identifie le concept dans un modèle sémantique qui décrit l'élément auquel il est attaché. Les 2 autres sont « lifting Schema Mapping » et « lowering Schema Mapping » qui remplacent l'élément « SchemaMapping » dans WSDL-S. Ces 2 attributs sont ajoutés à l'ensemble des déclarations d'éléments, les définitions des types de données simples ou complexes dans les schémas XML. Ils permettent d'établir la correspondance entre les données sémantiques (par exemple, les concepts dans une ontologie) référencée par « modelReference » et XML [9] et [11].

## 2.2 Approches basées sur des langages sémantiques

### WSMO

WSMO (Web Service Modeling Ontology) est une ontologie pour la description des services Web sémantiques aux moyens de :

- **Propriétés non-fonctionnelles** : telles que la performance, la fiabilité, la sécurité, la robustesse, et la sociabilité du service Web.
- **les médiateurs** utilisés par le service.
- **la fonctionnalité** : décrite en termes de pré-conditions, post-conditions, hypothèse et effets.
- **les interfaces** : Une interface de description d'un service Web est constituée d'informations telles que les erreurs gérées par le service, une description d'orchestration si le service fait appel à d'autres services, une description de la conversation du service appelée (échange de message), et des stratégies de compensations, utilisées dans le cas où certaines opérations exécutées par le service doivent être annulées.
- **les groundings** : Le grounding spécifie les informations concrètes pour l'accès au service.

## OWL-S

### ➤ Présentation

OWL-S est un langage de description des ontologies pour les services web. Il est basé sur le langage OWL. Le langage OWL est un langage basé sur XML, il a pour objectif d'ajouter des descriptions sémantiques aux services Web (en plus de leur description syntaxique WSDL). Il a été conçu pour faciliter l'automatisation des tâches relatives aux Services Web et plus particulièrement la découverte automatique des Services Web, leurs exécutions, ou bien encore leurs compositions et leurs interopérabilités.

### ➤ Les principaux composants du langage OWL-S

Le langage OWL-S organise la description d'un service en trois zones conceptuelles : le Service Profile, Service Model et le Service Grounding (figure 5).

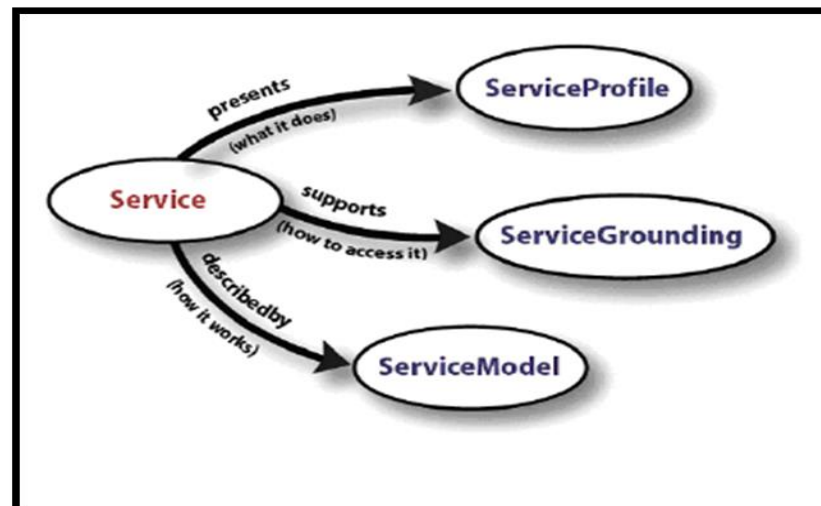


Figure 4 : L'ontologie supérieure de OWL-S

Le langage OWL-S est architecturé autour de quatre entités dont la classe Service fournit le point d'entrée pour la description d'un service Web. Il n'existe qu'une seule instance de la classe Service par Web Service.

- **Le Service Profile** : pour la publication et la découverte des services. Il permet à des agents logiciels d'accéder à l'ensemble des informations nécessaires leur permettant d'évaluer si le Web Service correspond à leurs besoins.
- **Le Service Model** : permet de connaître le fonctionnement du service. Il peut être utilisé de quatre manières différentes :
  - Effectuer une analyse plus complète du Web Service.

- Composer des descriptions de services.
  - Coordonner les activités des différents participants à l'exécution de ce service.
  - Surveiller l'exécution du service.
- **Le Service Grounding** : définit les modalités d'accès au service. Typiquement, le Service Grounding va spécifier un protocole de communication, les formats des messages ainsi que les autres détails spécifiques pour la mise en œuvre technique du service.

#### ➤ **Ontologie de Service Profile**

La figure6 illustre la partie supérieure de l'ontologie de Service Profile. La figure est logiquement divisée en trois parties :

- Le bas de l'ontologie représente la description non-fonctionnelles de service, il constitue la définition de l'acteur (Actor) : il enregistre des informations sur le fournisseur du service.
- La partie centrale décrit les propriétés fonctionnelles comme : QualityRating qui est le taux attribué au service, GeographicRadius qui spécifie les contraintes géographiques de service.etc.
- La partie supérieure de la figure représente la description fonctionnelle du service, il décrit

Les capacités du service en termes d'inputs/outputs, préconditions et effets. Les inputs sont ce qui est requis par un service afin de produire l'output désirée. Les préconditions représentent les conditions dans le monde qui devrait être vraies pour la bonne exécution du service. L'exécution du service va avoir des résultats dans le monde, ces résultats sont décrits par la description effets. [12]

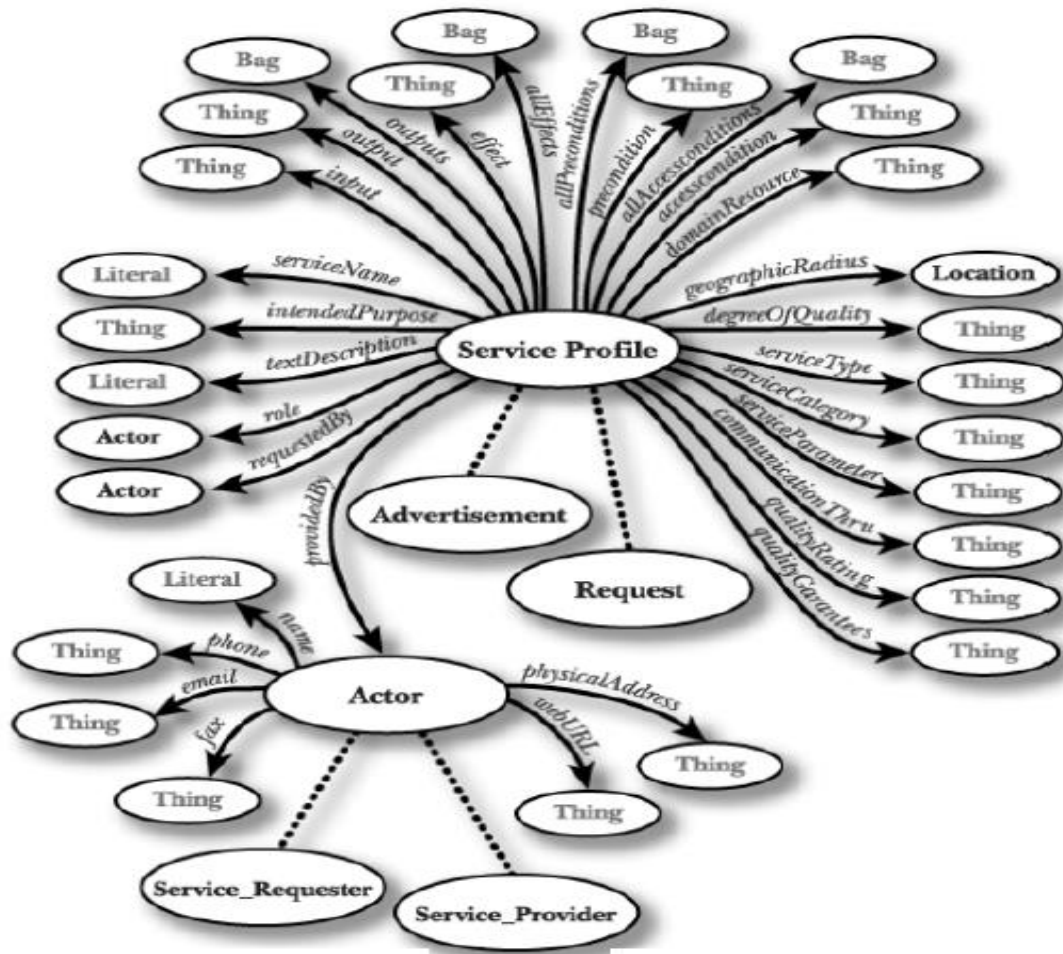


Figure 5 : Ontologie de Service Profile.

### 3. Conclusion

Dans ce chapitre, nous avons présenté les technologies de base concernant notre domaine. Il s'agit des services Web sémantiques, notamment les langages permettant leur description sémantique. Dans le chapitre suivant, nous présenterons la classification, dans lequel nous exposerons les principaux algorithmes effectuant les tâches de classification ou classement.

## **CHAPITRE IV : CLASSIFICATION**

## 1. Introduction

La notion de classification est essentielle en science. Elle permet aux scientifiques de mettre de l'ordre dans les connaissances qu'ils ont sur le monde. Ainsi, depuis longtemps des chercheurs et des scientifiques ont essayé de classer des espèces animales, De nombreuses classifications ont été créées. Face à ces classifications, les scientifiques sont souvent incapable de designer la meilleure d'entre elles. Car chacune présente un intérêt par rapport aux autres et à la tâche considérée. L'importance de la classification dans les sciences se reflète dans la grande variété des domaines ou tant leur nature que leur construction ont fait l'objet des recherches. Dans le cadre des problèmes de classification, on dispose d'un ensemble de données qui reprend une collection d'individus (objet) non étiqueté. Les classes sont encore inexistantes. L'objectif alors d'obtenir des classes d'objets homogènes en favorisant l'hétérogénéité entre ses différentes classes [13]. Dans ce chapitre nous présenterons un panorama des méthodes de classification les plus connues et qui font référence à l'existence de groupes ou classes de données, Elles se divisent en deux groupes:

- Les méthodes de classification automatique (aussi appelées méthodes de clustering): Méthodes basées sur la notion d'apprentissage non supervisé, laquelle consiste à regrouper des objets appartenant à un ensemble T en classes restreintes de telle sorte que les objets d'une même classe soient les moins dispersés possibles.
- Les méthodes d'affectation (aussi appelées classificateurs) basées sur la notion d'apprentissage supervisé : méthodes utilisant un ensemble d'exemples où les classes d'appartenance sont connues au préalable. À partir de cet ensemble, des normes (ou règles) d'affectation seront définies. De même, certains problèmes de classification nécessitent de combiner les deux types d'apprentissages (supervisée et non supervisée) appelé la méthode semi supervisé.

## 2. Domaines d'application de classification :

La classification a un rôle à jouer dans toutes les sciences et techniques qui font appel à la statistique multidimensionnelle. Citons tout d'abord les sciences biologiques : botanique, zoologie, écologie,... Ces sciences utilisent également le terme de "taxinomie" pour désigner l'art de la classification. De même les sciences de la terre et des eaux : géologie, pédologie, géographie, étude des pollutions, font grand usage de classifications. La classification est fort utile également dans les sciences de l'homme : psychologie, sociologie, linguistique, archéologie, histoire, etc. ... et dans les techniques dérivées comme les enquêtes d'opinion, le marketing, etc. ... Ces dernières emploient parfois les mots de "typologie" et "segmentation"

pour désigner la classification, ou l'une de ses innombrables variantes. Citons encore la médecine, l'économie, l'agronomie, et nous en oublions certainement ! Dans toutes ces disciplines la classification peut être employée comme une fin en soi ; mais elle l'est souvent, à juste titre, comme une méthode complémentaire à d'autres méthodes statistiques. Elle peut, en effet, aider efficacement à l'interprétation des graphiques d'analyse factorielle, ou bien déterminer des groupes d'objets homogènes, préalablement à une régression linéaire multiple [14].

### **3. Critères pour une bonne classification**

L'objectif principal des techniques de classification est de trouver une partition où les objets d'une classe devraient être semblables (entre eux), les objets de différentes classes devraient être différents [13], une bonne classification devrait accomplir différents critères:

#### **3.1 Validité**

Elle peut se définir par :

- Chaque classe d'une partition doit être homogène : Les objets qui appartiennent à la même classe doivent être semblables.
- Les classes doivent être isolées entre eux : Les objets de différentes classes doivent être différents.
- La classification doit s'adapter aux données : La classification doit pouvoir expliquer la variation des données.

#### **3.2 Interopérabilité**

Les classes doivent avoir une interprétation substantive c'est-à-dire qu'il est possible de donner des noms aux classes, dans le meilleur des cas les noms doivent correspondre aux types déduits d'une certaine théorie.

#### **3.3 Stabilité**

Les classes doivent être stable, ça veut dire la petite modification dans les données et dans les méthodes ne doivent pas changer les résultats.

#### **3.4 D'autre critère**

Parfois la taille et le nombre de classes sont employés en tant que critères Additionnels : le nombre de classée doit être aussi petit que possible, et la taille des Classes ne doit pas être trop petite [13].

## 4. Classification

### 4.1 Classification supervisée

C'est l'ensemble des techniques qui visent à deviner l'appartenance d'un individu à une classe en s'aidant uniquement des valeurs qu'il prend. Elle crée un modèle représentatif d'un certain nombre de données organisées en classes (ensemble) que l'on appelle généralement le corpus d'apprentissage - puis d'utiliser ce modèle afin de classer de nouvelles données, c'est à dire de prédire leur classe au vu de leur caractéristiques (appelées paramètres ou factures). La construction du modèle relève de l'apprentissage automatique supervisé, l'ensemble des exemples constituant le corpus d'apprentissage étant annotés, c'est à dire qu'ils portent le label de leur classe donné a priori. La plupart des algorithmes d'apprentissage supervisés tentent de trouver un modèle (une fonction mathématique) qui explique le lien entre des données d'entrée et les classes de sortie. Ces jeux d'exemples sont donc utilisés par l'algorithme. Il existe de nombreuses méthodes d'apprentissage supervisé [15] :

- Méthode des k plus proches voisins.
- Réseau de neurones.
- Arbre de décision.
- Classification naïve bayésienne.

La méthode directe de la classification supervisée k plus proches voisins

#### 4.1.1 K plus proches voisins (k-ppv)

La méthode des plus proches voisins (noté parfois k-PPV ou k-NN pour (k-Nearest-Neighbor) consiste à déterminer pour chaque nouvel individu que l'on veut classer, la liste des plus proches voisins parmi les individus déjà classés. L'individu est affecté à la classe qui contient le plus d'individus parmi ces plus proches voisins. Cette méthode nécessite de choisir une distance, la plus classique est la distance euclidienne et le nombre de voisins à prendre en compte. Cette méthode supervisée est souvent performante, cependant, le temps de prédiction est très long, car il nécessite le calcul de la distance avec tous les exemples, mais il existe des heuristiques pour réduire le nombre d'exemples à prendre en compte [16].

#### 4.1.2 Affectation par la méthode bayésienne

Un classificateur probabiliste linéaire simple basée sur le théorème de Bayes qui suppose que les descripteurs (attributs) qui décrivent les objets de l'ensemble d'apprentissage sont indépendants [14]. L'approche bayésienne a pour but de minimiser la probabilité d'erreur de classification, C'est-à-dire la probabilité jointe qu'une observation  $x$  soit en provenance d'une classe  $C_i$  et soit classée dans une autre [17].

### **4.1.3 Analyse discriminante**

Les méthodes d'analyse discriminante ont été largement étudiées; la littérature à ce sujet est très abondante. Le but de ces méthodes est de produire des décisions concernant l'appartenance ou non d'un objet à une classe en utilisant des fonctions discriminantes appelées également fonctions de décisions. Suivant les formes des classes, on peut trouver différents types de discrimination: discrimination linéaire et discrimination quadratique [17].

### **4.1.4 Les réseaux de neurones**

Les réseaux de neurones sont à l'origine d'une tentative de modélisation mathématique du cerveau humain. Le principe général des méthodes utilisant les réseaux de neurones consiste à modifier (ou ajuster) les paramètres comme, par exemple, les poids et les seuils par des algorithmes itératifs afin d'obtenir des réponses correctes [17].

### **4.1.5. Arbres de décision**

Un arbre de décision est une structure simple récursive permettant d'exprimer un processus de classification séquentiel au cours duquel une correspondance est établie entre un objet décrit par un ensemble de caractéristiques (attributs), et un ensemble de classes disjointes. Chaque feuille de l'arbre dénote une classe et chaque nœud intérieur un test portant sur un ou plusieurs attributs, produisant un sous arbre de décision pour chaque résultat possible du test [17].

## **4.2 Classification non supervisée**

### **4.2.1 Définition**

L'objectif de ces méthodes est de regrouper les individus en un nombre restreint de classes homogènes sans connaissances a priori [18]. L'apprentissage non supervisé consiste à inférer des connaissances sur des classes sur la seule base des échantillons d'apprentissage, et sans savoir a priori à quelles classes ils appartiennent [15].

On distingue aussi les approches de classification non hiérarchiques et les méthodes de classification hiérarchiques.

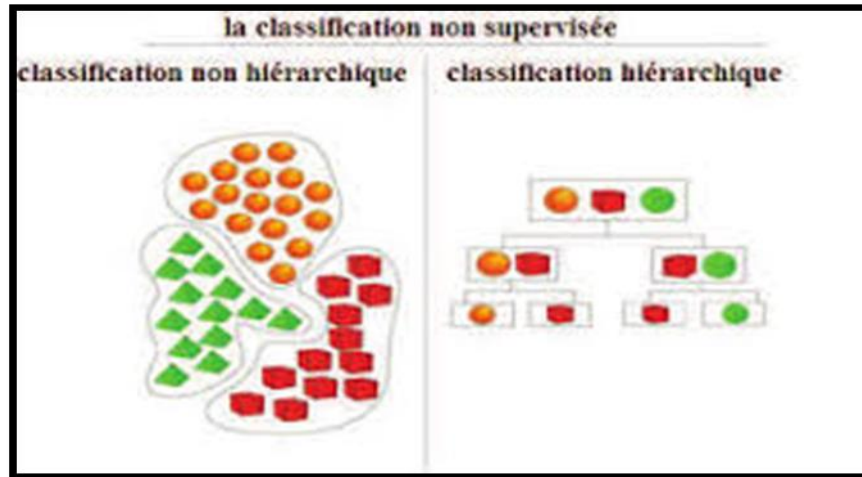


Figure 6 : les deux types de clustering hiérarchique/non hiérarchique.

#### 4.2.2 Méthodes non hiérarchiques

Ce sont des méthodes qui produisent directement une partition en un nombre fixé de classes [17].

Regrouper  $n$  individus en  $k$  classes de telle sorte que les individus d'une même classe soient le plus semblables possible et que les classes soient bien séparées [18]. Parmi ces méthodes, nous retrouvons:

##### a- k-means :

Ces méthodes construisent  $k$  classes à partir d'un ensemble de  $n$  individus, tout en minimisant la quantité :

$$\sum_{r=1}^k \sum_{x_i \in C_r} (x_i - g_r)^2$$

Ou :

- $C_r$  est la classe numéro  $r$ .
- $X_i$  est un individu dans une classe.
- $G_r$  est le centre de classe  $C_r$ .

**a.1 L'algorithme k-means:**

Entrée : k c'est le nombre de groupes recherchés.

Début

- (1) Choisir aléatoirement les centres des groupes.
- (2) Affecter chaque individu au centre le plus proche.
- (3) Recalculer le centre de chacune de ces groupes.
- (4) Répéter l'étape (2) et (3) jusqu'à stabilité des centres.
- (5) Editer la partition obtenue.

Fin.

Nous citons ici deux méthodes connues sur le principe de k-means sont :

- Méthode de centre mobile
- Méthode de nuée dynamique

Ces méthodes donnent la plupart de temps une partition localement optimale, Il est donc conseiller d'effectuer plusieurs exécutions et comparer les différentes résultats obtenues. Une suggestion consiste [21] à appliquer la méthode des k-means dans une première étape à plusieurs sous-ensembles de données extraits de l'ensemble total, et la meilleure partition obtenue fournit les centres à utiliser au départ de l'algorithme appliqué à l'ensemble total des données [13].

➤ **Nuées dynamiques (*isodata*) :**

Cette méthode a été proposée par Diday.E en 1972 [19]. Elle peut être considérée comme une généralisation de la méthode des centres mobiles. Le principe de la méthode est le suivant: on tire au hasard k noyaux parmi une famille de noyaux (chaque noyau contient un sous-ensemble d'individus). Puis chaque point de l'ensemble d'apprentissage est affecté au noyau dont il est plus proche. On obtient ainsi une partition en k classes dont on calcule les noyaux. On recommence le processus avec les nouveaux noyaux et ainsi de suite jusqu'à ce que la qualité de la partition ne s'améliore plus [13].

**Algorithme :**

- (1) Soit  $E$  l'ensemble à classer.
- (2) Soit  $f$  une fonction qui détermine un noyau d'une classe donnée.
- (3) Soit  $g$  une fonction qui détermine une classe autour d'un noyau donné avec Noyau: c'est l'ensemble d'éléments qui agit comme un centre.
- (4) Soit  $W$  une fonction qui mesure l'homogénéité des classes d'une partition donnée et un ensemble de nœuds donné.
- (5) Soit  $K$  le nombre de classes à créer.
- (6) Choisis  $K$  noyaux dans  $E$ .
- (7) Tant que  $W$  n'est pas satisfaisant
- (8) Utilise  $g$  pour déterminer une classe autour de chaque noyau.
- (9) Utilise  $f$  pour déterminer les noyaux de ces classes.
- (10) Fin Tant que

**➤ Méthodes de centres mobiles :**

Cette méthode est développée par Forgy en 1965[20]. Elle consiste à construire une partition en  $k$  classes en sélectionnant  $k$  individus comme centres de classes tirés au hasard de l'ensemble d'individus. Après cette sélection, on affecte chaque individu au centre le plus proche en créant  $k$  classes, les centres des classes seront remplacés par les centres de gravité et nouveaux classes seront créés par le même principe [13].

Donnée : k le nombre maximum de classes désiré.

Début

(1) Choisir k individus au hasard (comme centre des classes initiales)

(2) Affecter chaque individu au centre le plus proche

Ce qui donne une partition en k classes  $P_1 = \{C_1, C_2, \dots, C_k\}$

(3) On calcule les centres de gravité des chacune des classes de  $P_1$  ce qui donne K nouveaux centres de classes.

(4) Répéter l'étape (2) et (3) jusqu'à deux itérations successives donnent la même partition

(5) éditer la partition obtenue.

FIN

Généralement la partition obtenue est localement optimale car elle dépend du choix initial des centres. Pour cela les résultats entre deux exécutions de l'algorithme sont significativement variés [13].

### **b-K-médoïds**

Dans ces méthodes une classe est représenté par un de ces individus (médoïde), c'est une méthode itérative combinant la réaffectation des individus dans des classes avec une intervention des médoïdes et des autres individus, c'est une méthode simple parce qu'elle couvre n'importe qu'elle type de variables, quand des médoïds sont choisis, des classes sont définis comme sous-ensembles des individus près des médoïdes les plus proches par rapport à une mesure de distance choisie. Il est alors plus judicieux de choisir comme centre de groupe un individu présent dans le groupe et non un individu calculé. La médoïde d'un groupe est l'individu possédant la dissimilarité moyenne la plus faible d'avec les autres individus du groupe [13].

- **Avantages**

- ❖ Bonne résistance aux données erronées,
- ❖ Flexibles avec tout type de distance,

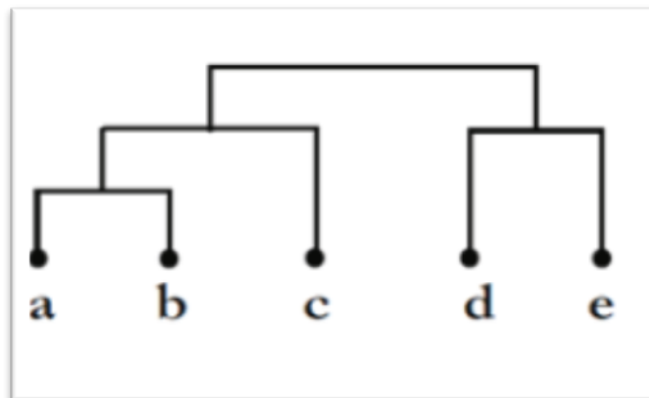
- **Inconvénients**

- ❖ Nécessité de spécifier le nombre de clusters  $k$ .
- ❖ Complexité de chaque itération.

En conclusion, les méthodes non hiérarchiques permettent de traiter rapidement de grands ensembles d'individus, mais elles supposent que le nombre des classes est fixé au départ. Si le nombre de classes n'est pas connu ou si ce nombre ne correspond pas à la configuration véritable de l'ensemble d'individus (d'où le risque d'obtenir des partitions de valeurs douteuses), il faut presque toujours tester diverses valeurs de  $k$ , ce qui augmente le temps de calcul. C'est pourquoi, lorsque le nombre des individus n'est pas trop élevé, on préfère utiliser les méthodes hiérarchiques [17].

### 4.2.3 Méthodes hiérarchiques

La classification hiérarchique consiste à effectuer une suite de regroupements en classes de moins en moins fines en agrégeant à chaque étape les objets ou les groupes d'objets les plus proches. Elle fournit ainsi un ensemble de partitions de l'ensemble d'objets [22]. Cette approche utilise la notion de distance, qui permet de refléter l'homogénéité ou l'hétérogénéité des classes. Ainsi, on considère qu'un élément appartient à une classe s'il est plus proche de cette classe que de toutes les autres. La (Figure7) est une illustration du principe des méthodes hiérarchiques [17]. Dans cette figure, on représente la suite de partitions d'un ensemble  $\{a, b, c, d, e\}$  :



**Figure 7 : La partition hiérarchique.**

### a. Les critères de similarité

On peut mesurer la similarité entre deux objets par [14] :

#### ➤ La distance euclidienne :

Un rapport de clusters analyses en psychologie de la santé a conclu que la mesure de la distance la plus courante dans les études publiées dans ce domaine de recherche est la distance euclidienne ou la distance au carré euclidienne.

$$d^2(x_1, x_2) = \sum_i (x_{1i} - x_{2i})^2 = (x_1 - x_2)(x_1 - x_2)'$$

#### ➤ La distance de Manhattan :(appelée aussi taxi-distance)

$$d^2(x_1, x_2) = \sum_i |x_{1i} - x_{2i}|$$

#### ➤ La distance de Mahalanobis

Corrige les données pour les différentes échelles et des corrélations dans les variables, L'angle entre deux vecteurs peuvent être utilisés comme mesure de distance quand le regroupement des données de haute dimension. Voir l'espace produit scalaire.

Où (C = covariance)

$$d^2(x_1, x_2) = (x_1 - x_2)C^{-1}(x_1 - x_2)$$

#### ➤ La distance de Hamming

Mesure le nombre minimum de substitutions nécessaires pour changer un membre dans un autre. Elle permet ainsi, de quantifier la différence entre deux séquences de symboles, généralement utilisée dans le cas des valeurs discrètes (vecteurs).

$$d(a, b) = \sum_{i=0}^{n-1} (a_i \oplus b_i)$$

Exemple : Considérons les suites binaires suivantes :

a= (0 0 0 1 1 1 1) et b= (1 1 0 1 0 1 1) alors d =1+1+0+0+1+0+0

La distance entre a et b est égale à 3 car 3 bits diffèrent.

#### ➤ La métrique Minkowski :

Pour les données dimensionnelles, c'est la mesure Populaire où d est la dimensionnalité des données.

$$d_p(x_i, x_j) = \left( \sum_{k=1}^d |x_{ik} - x_{jk}|^p \right)^{\frac{1}{p}}$$

La distance euclidienne est un cas particulier où  $p = 2$ , alors que Manhattan  $p = 1$ . Néanmoins, il n'existe pas de directives générales théoriques pour la sélection d'une mesure à une application donnée. Une autre question, est de savoir comment mesurer la distance entre 2 classes  $D (C_1 ; C_2)$  ? Pour cela il y a certaines fonctions permettent de mesurer cette distance comme :

❖ plus proche voisin :  $\min (d(i, j), i \in C_1, j \in C_2)$

❖ Diamètre maximum :  $\max (d(i, j), i \in C_1, j \in C_2)$

❖ Distance moyenne :

$$\frac{\sum_{i,j} d(i,j)}{n_1 n_2}$$

### ➤ Similarité Cosinus

La similarité cosinus est fréquemment utilisée [27] en tant que mesure de ressemblance entre deux documents  $d_1$  et  $d_2$ . Il s'agit de calculer le cosinus de l'angle entre les représentations vectorielles des documents à comparer. La similarité obtenue Sim cosinus

$(d_1, d_2) \in [0,1]$ .

$$\text{Sim cosinus } (d_1, d_2) = \frac{\vec{d_1} \cdot \vec{d_2}}{\|\vec{d_1}\| \|\vec{d_2}\|}$$

$\vec{d_1} \cdot \vec{d_2}$  est le produit scalaire des deux vecteur =  $x_1 * y_1 + \dots + x_n * y_n$

$\|\vec{d_1}\|$  est la longueur du vecteur =  $\sqrt{x_1^2 + x_2^2 + \dots + x_n^2}$

### ➤ Coefficient de Jaccard

L'indice de Jaccard ou coefficient de Jaccard [26] est le rapport entre la cardinalité (la taille) de l'intersection des ensembles considérés et la cardinalité de l'union des ensembles. Il permet d'évaluer la similarité entre les ensembles. Les documents  $d_1$  et  $d_2$  sont donc représentés, non pas comme des vecteurs, mais comme des ensembles de termes. La similarité obtenue Sim Jaccard  $(d_1, d_2) \in [0,1]$ .

$$\text{Sim Jaccard } (d_1, d_2) = \frac{\|d_1 \cap d_2\|}{\|d_1 \cup d_2\|}$$

### **b. Algorithme de la CAH**

Il en existe un bon nombre d'algorithmes pour les méthodes hiérarchiques, On se limite à présenter l'algorithme de la méthode directe CAH (Classification Ascendante Hiérarchique). Ce dernier est celui le plus simple, il est dû à Lance et Wiliam (1967).

#### ➤ *Initialisation*

Construction du tableau des distances, peu importe la formule utilisée pour le construire car l'algorithme de la C.A.H est indépendant de la métrique utilisée. Ainsi, entre chaque couple de points  $(x, y)$  de  $M$ , nous disposons d'une valeur  $d(x, y)$ . La partition initiale est la plus fine  $P_0$  de  $M$ .

#### ➤ *Regroupement*

Parcourir le tableau des distance pour déterminer le couple d'élément  $(x^*, y^*)$  les plus proches :

$$d(x^*, y^*) \leq \min_{x, y \in M} \{d(x, y)\}$$

On réunit les deux éléments dans une même classe  $A = x^* \cup y^*$ , les autres classes restent inchangées. Nous obtenons une nouvelle partition  $p_i$  moins fine que la précédente.

#### ➤ *Tableau des distances*

La classe  $A$  sera vue comme un seul point. Il faut donc calculer les distances qu'il y a entre le point  $A$ , qui est un ensemble de cardinal supérieur à un, et tous les autres points qui ne sont pas dans  $A$  et qui peuvent être des singleton. Par souci de généralité, nous les notons  $B$ .

$$D(A, B); \quad B \not\subseteq A$$

Pour cela, on peut utiliser l'un des six critères de dissimilarité proposés plus haut. Nous disposons alors d'un nouveau tableau des distances ayant une ligne et une colonne de moins que le précédent dont il ne diffère que par la ligne et la colonne qui correspond au point  $A$ .

#### ➤ *Condition d'arrêt*

Si nous avons atteint la partition du niveau souhaité, généralement c'est la partition grossière, celle qui ne comporte qu'une seule classe réunissant la totalité des points, alors, c'est terminé. Dans le cas contraire, nous repartons de l'étape « Regroupement » à partir du tableau des distances calculé à la suite du précédent regroupement.

Malgré le nombre important des méthodes de classification, il n'est existé pas un algorithme qui répond à tous les demandes et plusieurs problématiques restent encore ouvertes dans le cadre de la classification.

## 5. Conclusion

Plusieurs solutions sont proposées pour résoudre le problème de classification, chacune de ces solutions se différencie par les mesures de similarité qu'elles utilisent, la nature des données qu'elles traitent et l'objectif final souhaité, ainsi que chacune possède ses points forts et ses points faibles. Les méthodes hiérarchiques ascendantes sont souvent utilisées en cas de données de petite taille à cause de leurs complexités élevées, par conséquent des sérieux problèmes du temps et d'espace se posent en cas où les données sont de grande taille. Dans notre contexte nous proposons l'utilisation de l'algorithme k-means, malgré il est sensible à l'initialisation (nombre du cluster, et sélection des points de départ) mais généralement il donne des bons résultats dans plusieurs domaines d'applications [25]. Dans le chapitre qui se suit nous présentons la partie conception de notre application.

## **CHAPITRE V : CONCEPTION**

## 1. Introduction

Dans ce chapitre, nous étudions le coté conceptuel qui constitue une étape fondamentale avant l'implémentation. Dans notre application on a utilisé le langage UML. Les diagrammes UML utilisés sont les suivants : le diagramme de cas d'utilisation, diagramme d'activité. Ces diagrammes sont conçus à l'aide de l'outil StarUml.

## 2. Présentation d'UML

### Définition :

UML (Unifie Modeling Language) est un langage visuel constitue un ensemble de schémas, appelés diagrammes, qui donnent chacun une vision différente du projet à traiter. UML nous fournit donc des diagrammes pour représenter le logiciel à développer (son fonctionnement, sa mise en route, les actions susceptibles d'être effectuées par le logiciel, etc.).

Un diagramme UML est une représentation graphique, qui s'intéresse à un aspect précis du modèle ; c'est une perspective du modèle. Chaque type de diagramme UML possède une structure (les types des éléments de modélisation qui le composent sont prédéfinis) et véhicule une sémantique précise (il offre toujours la même vue d'un système).

Combinés, les différents types de diagrammes UML offrent une vue complète des aspects statiques et dynamiques d'un système. Les diagrammes permettent donc d'inspecter un modèle selon différentes perspectives et guident l'utilisation des éléments de modélisation (les concepts objet), car ils possèdent une structure.

Une caractéristique importante des diagrammes UML, est qu'ils supportent l'abstraction. Cela permet de mieux contrôler la complexité dans l'expression et l'élaboration des solutions objet.

UML opte en effet pour l'élaboration des modèles, plutôt que pour une approche qui impose une barrière stricte entre analyse et conception. Les modèles d'analyse et de conception ne diffèrent que par leur niveau de détail, il n'y a pas de différence dans les concepts utilisés. UML n'introduit pas d'éléments de modélisation propres à une activité (analyse, conception.) le langage reste le même à tous les niveaux d'abstraction.

Cette approche simplificatrice facilite le passage entre les niveaux d'abstraction. L'élaboration encourage une approche non linéaire, les "retours en arrière" entre niveaux d'abstraction différents sont facilités et la traçabilité entre modèles de niveaux différents est assurée par l'unicité du langage.

UML favorise donc le prototypage, et c'est là une de ses forces. En effet, modéliser une application n'est pas une activité linéaire. Il s'agit d'une tâche très complexe, qui Nécessite une approche itérative, car il est plus efficace de construire et valider par étapes, ce qui est difficile à cerner et maîtriser [28].

### 3. Diagrammes UML utilisé pour la conception de notre application

#### 3.1 Diagramme des cas d'utilisation

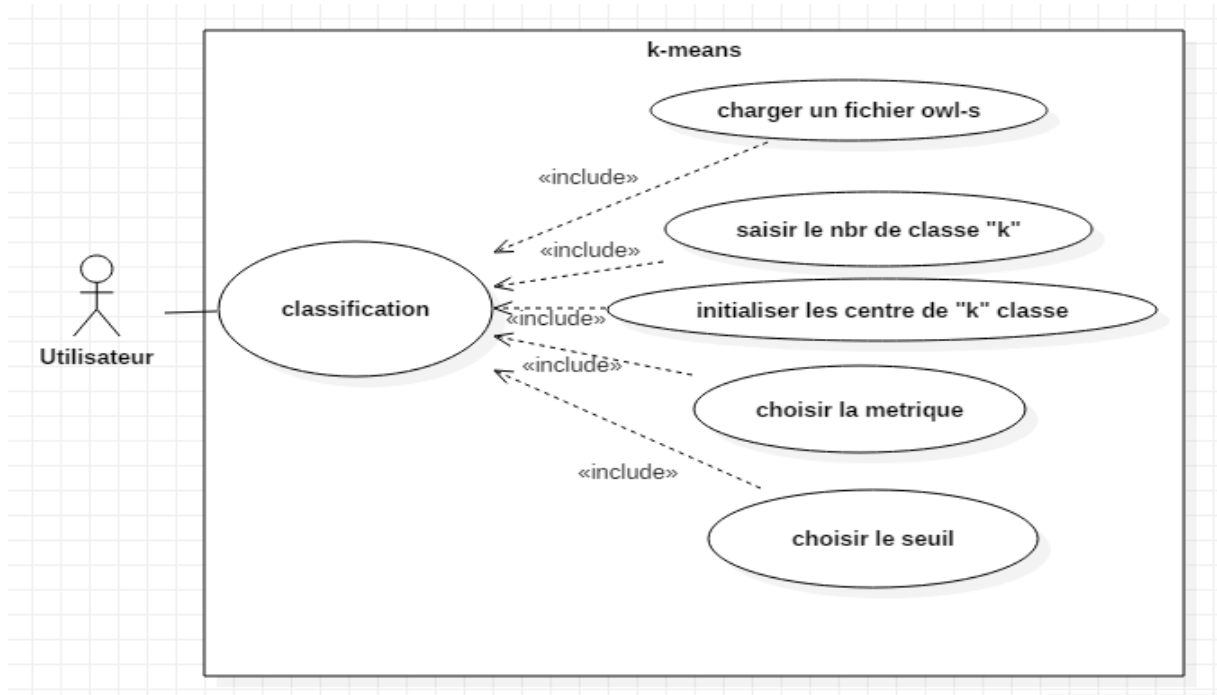
Il permet d'identifier les possibilités d'interaction entre le système et les acteurs (intervenants extérieurs au système), c'est-à-dire toutes les fonctionnalités que doit fournir le système.

**Acteur:** Utilisateur

➤ Classification :

- Charger les fichiers owl-s.
- Saisir le nombre de classe « k ».
- Initialiser les centres de « k » classe.
- Choisir la métrique.
- Choisir le seuil.

Dans la figure ci-dessous nous présentons le diagramme de cas d'utilisation qui regroupe les fonctionnalités de notre système (voir figure8).



**Figure 8: diagramme de cas d'utilisation.**

### 3.2 Diagramme d'activité :

. Le diagramme d'activité permet de représenter le comportement interne d'un use case ou processus, représenté le déroulement des traitements en les regroupant dans des étapes appelées « Activité», la figure ci-dessous représente le diagramme d'activité de notre application (Voir figure 9).

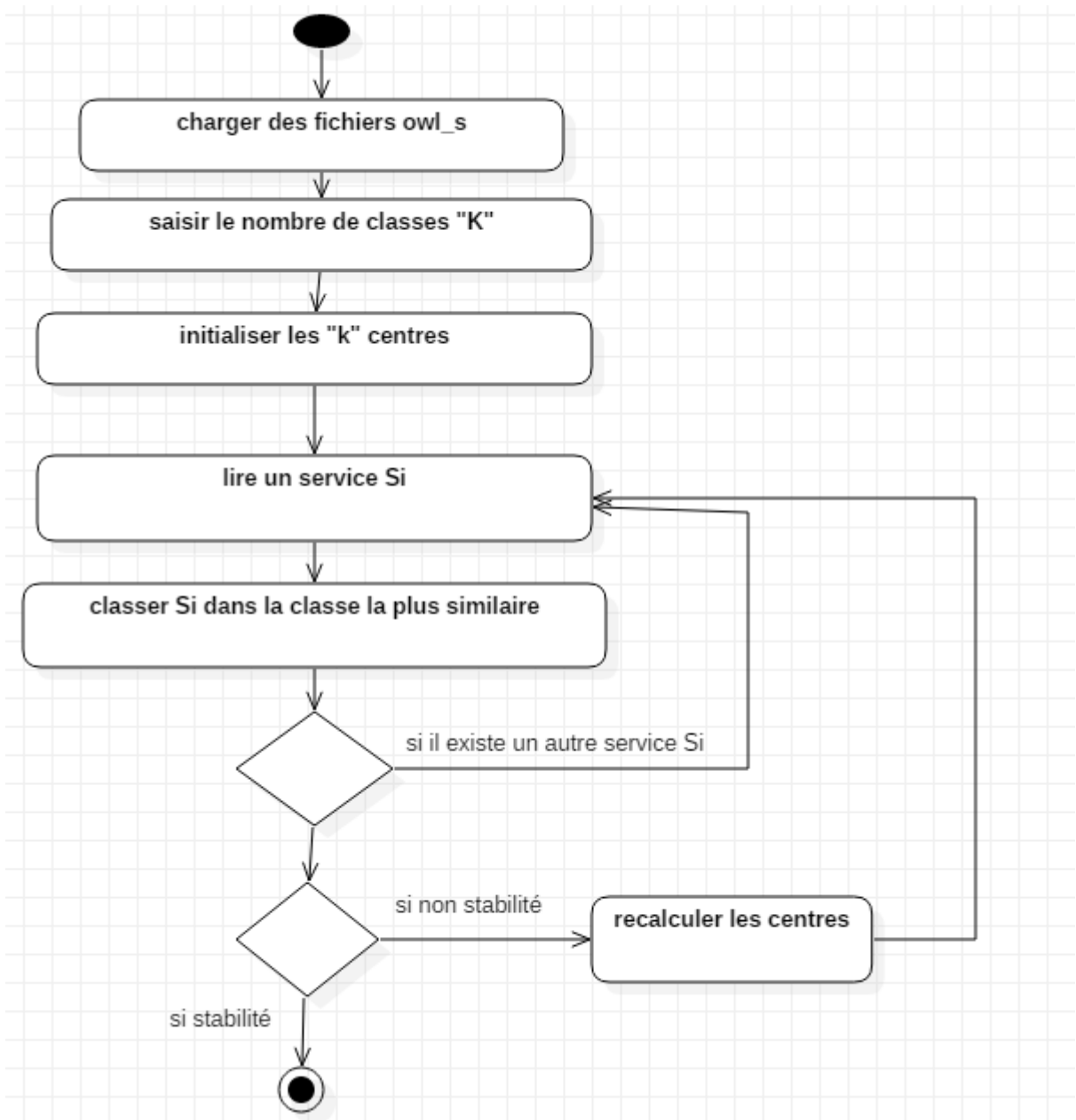


Figure 9:diagramme d'activité

#### **4. conclusion**

Dans ce chapitre, on a essayé de donner une vue conceptuelle du travail réalisé, à travers les différents diagrammes décrits en UML (diagramme de cas d'utilisation, diagramme d'activité de la classification des services web). Dans le chapitre qui se suit nous passons à détailler la phase implémentation.

# **CHAPITRE VI : IMPLEMENTATION**

## 1. Introduction

Nous allons commencer par présenter les différents outils et techniques utilisées dans l'implémentation, ensuite, nous présentons l'interface de notre algorithme, les principales étapes d'utilisation de notre application et un exemple illustratif.

## 2. Outils et environnement de développement

Nous avons développé notre application sur une machine ayant un processeur Intel Core i3, avec une vitesse de 2.40GHz, dote d'une capacité mémoire de 4GB de RAM sous Windows 10. Avant de commencer l'implémentation de notre application, nous allons tout d'abord spécifier les outils utilisés qui nous ont semblé être bon vu les avantages qu'ils Offrent :

### • Langage Java

Pour le langage de programmation notre choix s'est porté sur le langage JAVA, et cela parce que JAVA est un langage orienté objet simple ce qui réduit les risques D'incohérence; il est portable, il peut être utilisé sous Windows, sous Linux, sous Macintosh et sur d'autres plateformes sans aucune modification, enfin, il possède une riche bibliothèque de classes comprenant des fonctions diverses telles que les fonctions standards, le système de gestion de fichiers, les fonctions multimédia et beaucoup d'autres Fonctionnalités.

### • Eclipse

Pour le choix de l'environnement de développement, on a opté pour Eclipse car il Possède de nombreux points forts qui sont à l'origine de son énorme succès dont les Principaux sont :

- Une plateforme ouverte pour le développement d'applications et extensible grâce à un Mécanisme de plugins.
- Supporte plusieurs plates-formes d'exécution : Windows, Linux, Mac OS.
- Malgré son écriture en Java, Eclipse est très rapide à l'exécution grâce à l'utilisation de la Bibliothèque SWT.

## 3. Matching et mesure de similarité

Afin de calculer le degré de similarité entre la requête de l'utilisateur et les concepts des ontologies du domaine, et avec les (entrées/sorties) des services web, nous avons employé le Wordnet.

### 3.1. WordNet API

Notre application utilise les mesures de similarité à base WordNet [23]. Ces mesures sont calculées, puis normalisées. La normalisation consiste généralement à convertir la valeur de

mesure pour obtenir une nouvelle valeur comprise entre 0 et 1. La valeur 1 indique qu'il existe une équivalence sémantique complète entre les deux entités. Notre application interagit avec l'API Java WordNet pour calculer les mesures de similarité sémantique entre les entités.

#### 4. Corpus utilisés pour la découverte

Nous avons utilisé un extrait du corpus open source OWLS-TC version 4.0. Ce dernier est développé par le centre allemand pour la recherche en intelligence artificielle [24]. Cette version décrit un ensemble de services web à travers des documents owls, elle dispose de 1083 services web segmentés en 09 classes différents : éducation, soins médicaux, Nourriture, voyage, communication, économie, armes, géographie et simulation. La majorité de ces services sont extraits de l'annuaire UDDI d'IBM, et sont traduits du format WSDL en format OWLS de façon semi-automatique. La base propose aussi un ensemble de requêtes réparties sur les 09 classes, ces requêtes sont modélisées sous forme de document OWLS. Chaque document OWLS (service ou requête) comporte dans sa partie «Profile » des éléments « profile : has input » et « profile : has output », ces derniers sont employés comme entrées pour le module de découverte des services web. Chaque service web (i.e. document OWLS) est étiqueté manuellement par des experts humains comme étant relevant ou non par rapport à une requête donnée. En d'autres termes, chaque service web possède une étiquette binaire (relevant ou non) par rapport à une requête donnée. La base offre aussi un ensemble d'ontologies pour décrire les services et les requêtes, chaque classe de services possède une ou plusieurs ontologies. Dans ce qui suit, nous donnons un exemple d'une description de service web et d'une description de requête à l'aide des éléments profile : has input » et « profile : has output ».

- **Requête**

Cette requête cherche les moyens de diagnostics offerts par un hôpital donné.

Concepts d'entrée : Hospital

Concepts de sortie: Investigating

- **Service**

Ce service donne les prix des véhicules ayant trois roues.

Nom de service : 3wheeledcar\_price

Concepts d'entrée: 3 wheeled car

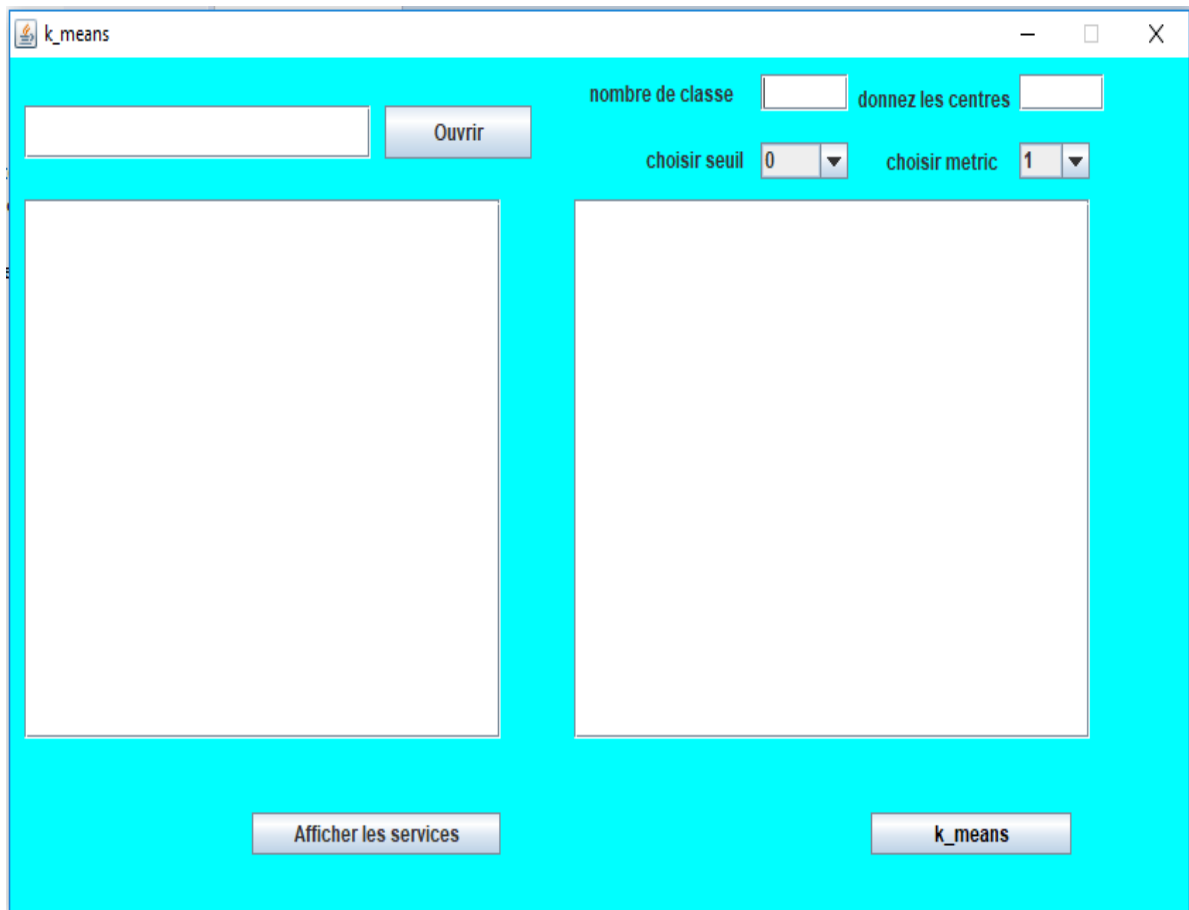
Concepts de sortie : price

### 5. Présentation de L'application réalisée :

L'interface homme/machine (IHM) représente l'élément clé dans l'utilisation de tout système informatique. Les interfaces de notre système sont conçues de manière à être simples, compréhensible et faciles à utilisées.

Afin d'illustrer les fonctionnalités de notre application, nous avons effectué quelques prises d'écrans montrant les principaux étapes d'utilisation de notre application :

- L'interface principale de notre application permet l'utilisateur de faire plusieurs choses Parmi eux charger un fichier de services OWL-S, Remplir le champ (nombre de classe(k), les centre de (k) classe), choisir la métrique et le seuil et enfin faire la classification (voir Figure 10).



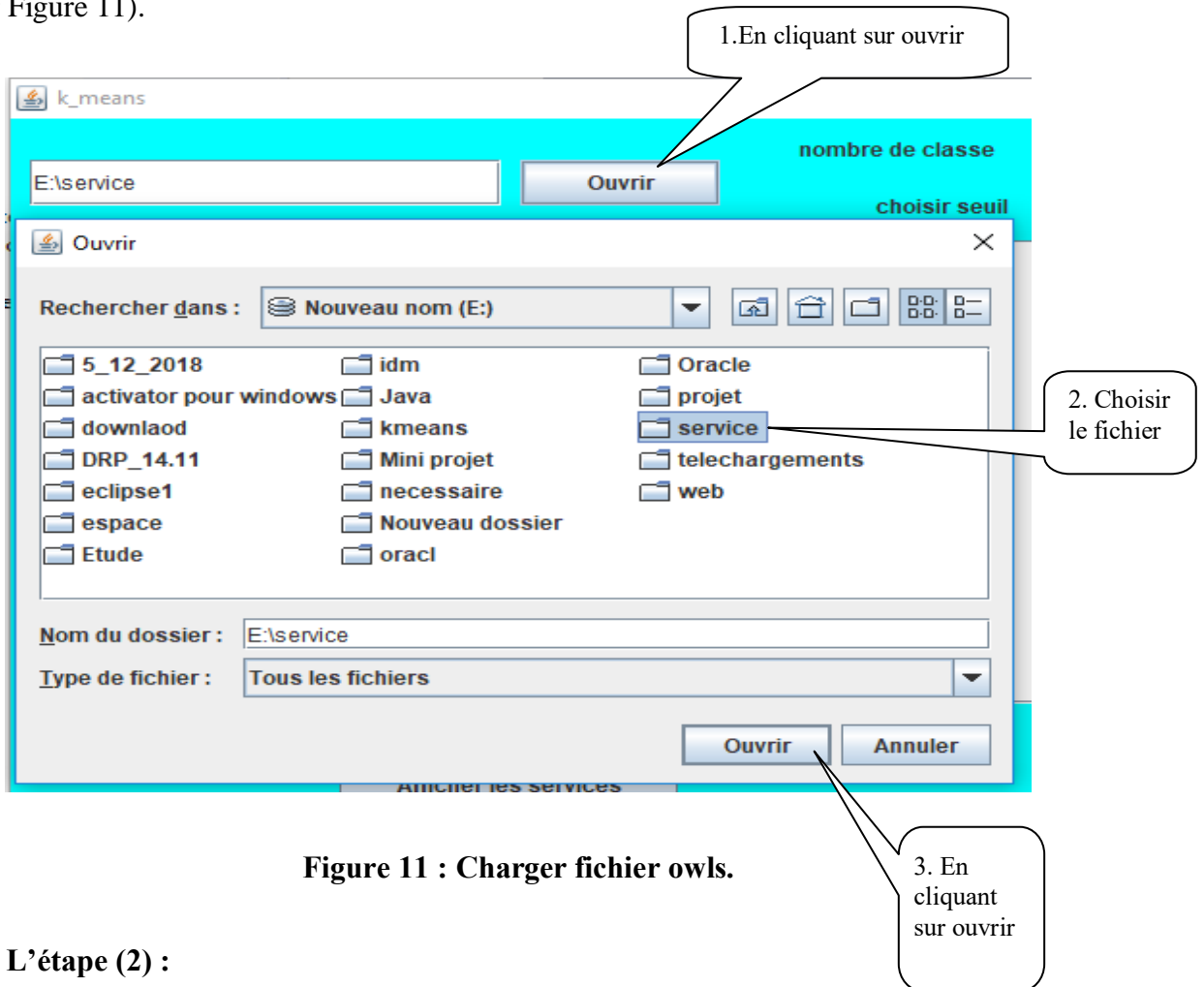
**Figure 10 : Interface principale de la classification des SWS avec k\_means**

**5.1 Les étapes de démarche :**

• **L'étape (1) :**

Pour charger un fichier (OWLS) clic sur le bouton ouvrir et choisir le fichier

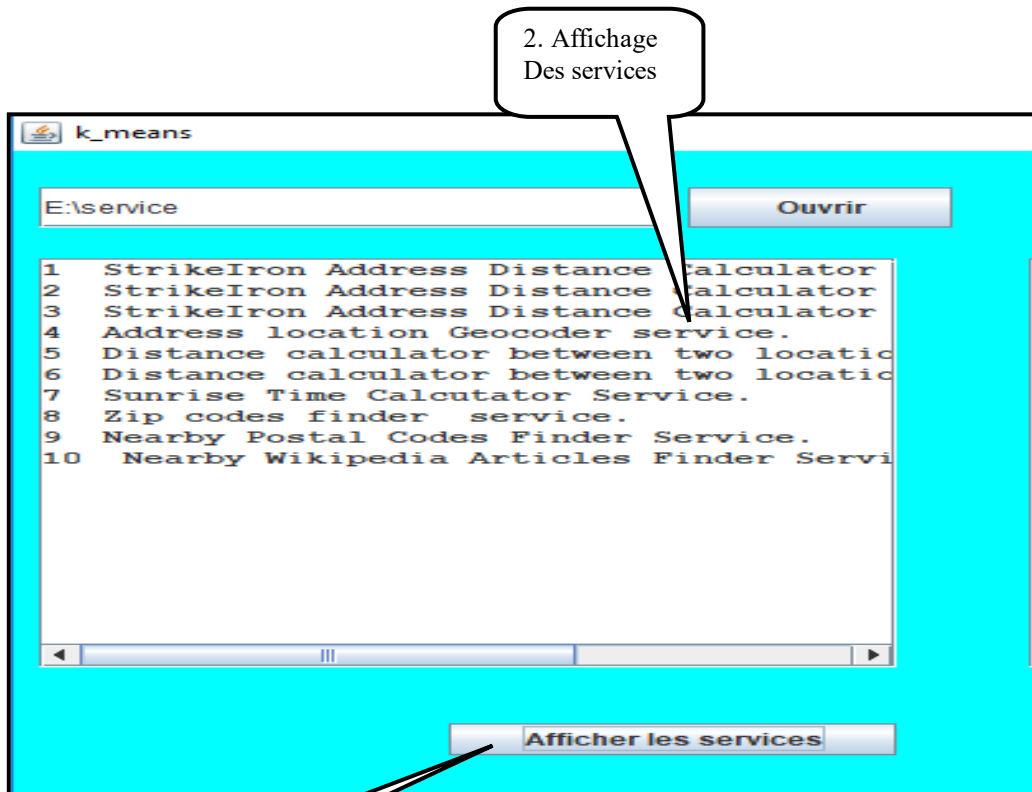
(Voir Figure 11).



**Figure 11 : Charger fichier owls.**

• **L'étape (2) :**

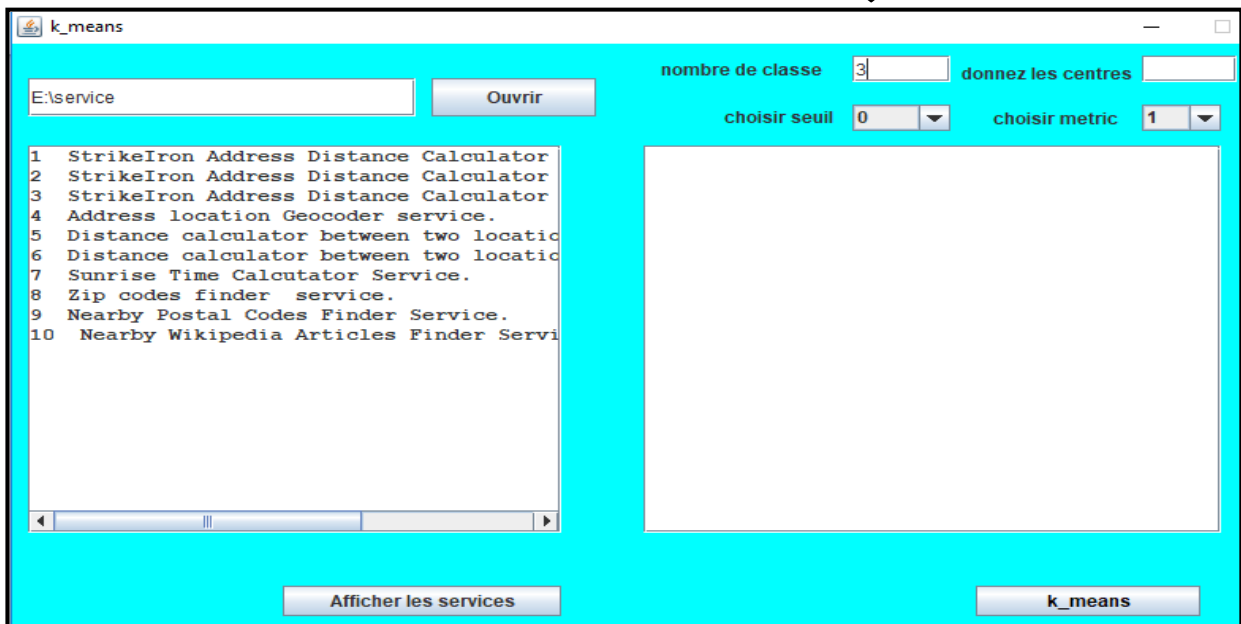
Pour afficher les services clic sur le bouton Afficher les services (voir Figure 12).



• L'étape (3) :

Saisir le nombre de classe (Figure 13).

Saisir le nombre de classe



- **L'étape (4) :**

La saisir des centres des classes (Figure 14).

Saisir les centres des classes.

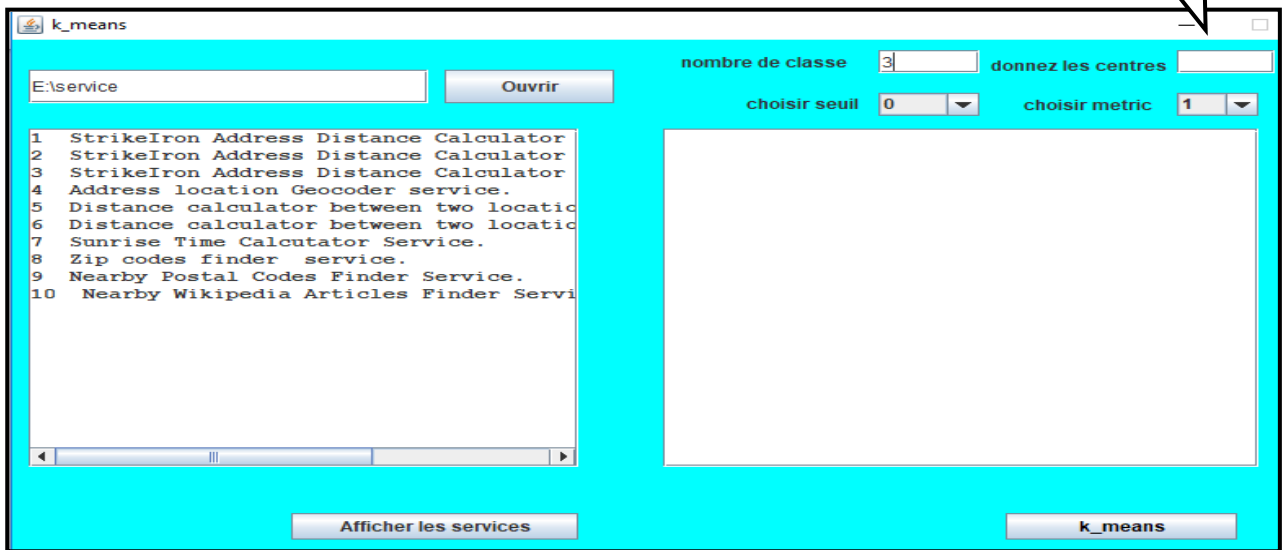


Figure 14 : Saisir les centres des classes.

- **L'étape (5) :**

Choisir un seuil entre 0 et 1 (voir Figure 15).

Choisir le seuil

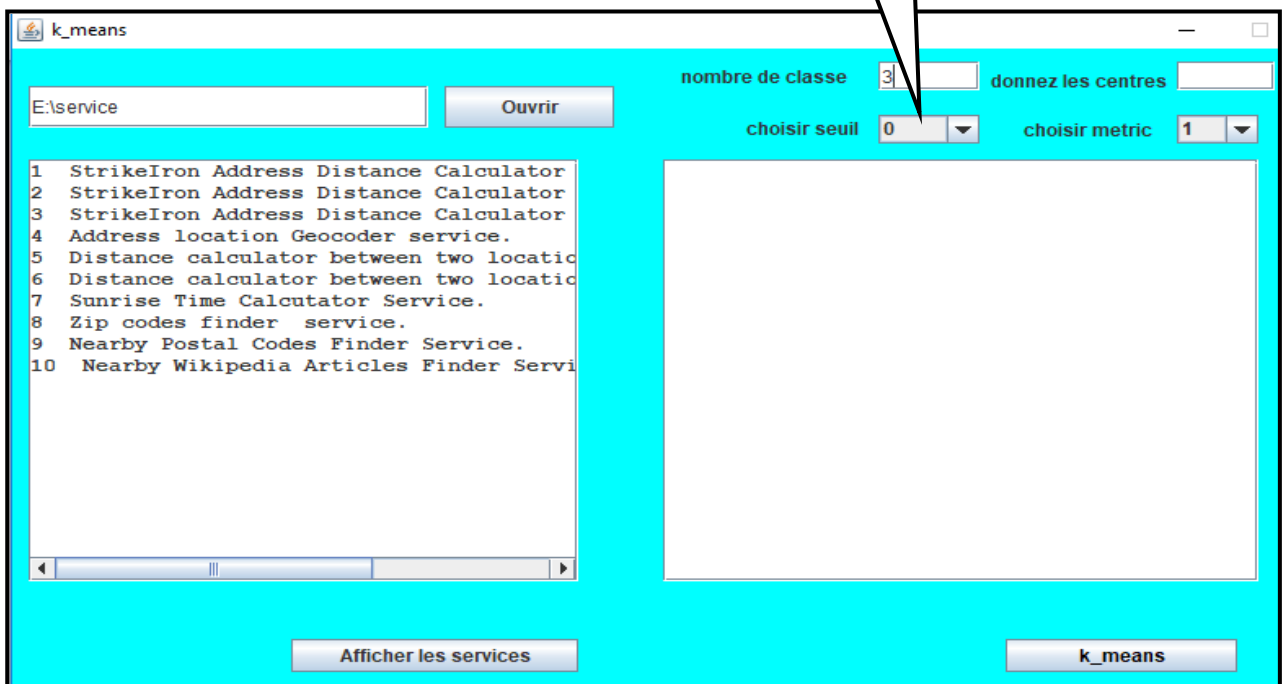


Figure 15 : choisir le seuil.

- **L'étape (6) :**  
Choisir une métrique entre 1 et 4. (Voir Figure 16)

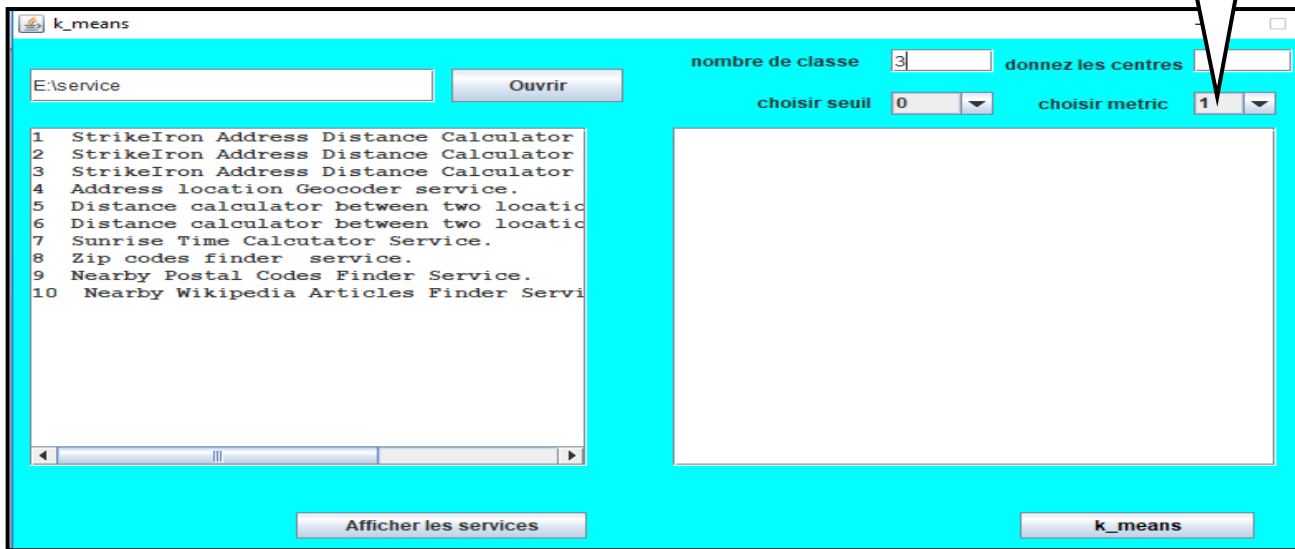


Figure 16 : choisir la métrique.

- **L'étape (7) :**  
Pour faire la classification des services qu'on a sélectionnés, on clique sur le bouton k\_means. Les résultats de la classification seront affichés dans une fenêtre d'affichage (voir figure 17).

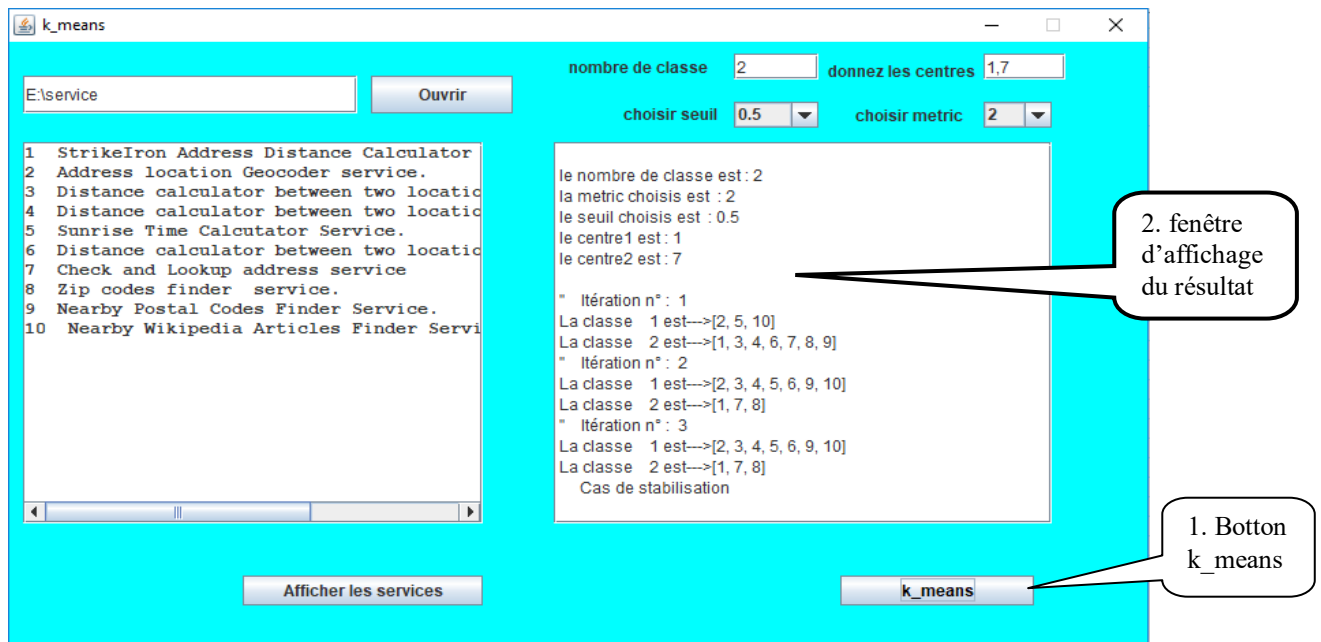


Figure 17: les résultats de la classification en 3 classes.

## 6. Exemple illustratif :

Nous voulons regrouper un ensemble de services web sémantique dans des groupes similaires avec l'algorithme k-means, Les services web sémantique sont sous forme (owls) qui comporte dans sa partie «Profile » des éléments « profile : has input » et « profile : has output », Nous voulons classer quatre services en deux groupes utilisant l'algorithme k-means, Basé sur la mesure de similarité Coefficient de Jaccard :  $\text{Sim Jaccard}(d1, d2) \in [0,1]$ .

$$\text{Sim Jaccard}(d1, d2) = \frac{\|d1 \cap d2\|}{\|d1 \cup d2\|}$$

Service [has input] [has output]

1. [Address, Address, City, City, State, State] [Distance]
2. [Address] [Address, latitude, longitude]
3. [latitude, longitude, latitude, longitude] [Distance]
4. [latitude, longitude, latitude, longitude] [Distance]

Initialisation : Nous avons deux groupes et nous choisissons des centres aléatoires, la métrique et le seuil.

- Les centres est : (1, 4).
- La métrique est : 2.
- Le seuil est : =0,7.

### L'étape 1 :

❖ Calculer la distance entre les centres (C1, C2) et les services (S1, S2, S3, S4) :

1. le première centre (C1) est : 1 [Address, Address, City, City, State, State][Distance].
2. le deuxième centre (C2) est : 4 [latitude, longitude, latitude, longitude][Distance].

❖  $\text{Sim Jaccard}(C1, S2) = 0.166666666666666666$ .

❖  $\text{Sim Jaccard}(C2, S2) = 0.0$ .

➔  $\text{Sim Jaccard}(C1, S2) > \text{Sim Jaccard}(C2, S2)$ , donc le deuxième service (S2) appartient à la première classe et on note : [C1, S2].

❖  $\text{Sim Jaccard}(C1, S3) = 0.5$ .

❖  $\text{Sim Jaccard}(C2, S3) = 1.0$ .

➔  $\text{Sim Jaccard}(C1, S3) < \text{Sim Jaccard}(C2, S3)$ , donc le troisième service appartient à la deuxième classe et on note : [C2, S3].

❖ Classer chaque service avec le centre le plus similaire :

➤ La classe 1 est [1, 2].

➤ La classe 2 est [3, 4].

**L'étape 2 :**

Recalculer le centre de chaque classe et répéter la première étape, Pour calculer le centre on utilise l'union des inputs et outputs des services de même classe.

- L'Union de classe [1,2] = [Address] [Address, latitude, longitude].
  - L'Union de classe [3,4] = [latitude, longitude] [Distance].
1. le nouveau centre 1 (C1) est : [Address] [Address, latitude, longitude].
  2. le nouveau centre 2 (C2) est : [latitude, longitude] [Distance].
- ❖ Sim Jaccard (C1, S2) =: 1.0.
  - ❖ Sim Jaccard (C2, S2) =: 0.0.
  - ➔ Sim Jaccard (C1, S2) > Sim Jaccard (C2, S2), donc le deuxième service (S2) appartient à la première classe et on note : [C1, S2].
  - ❖ Sim Jaccard (C1, S3) = 0.0.
  - ❖ Sim Jaccard (C2, S3) = 1.0.
  - ➔ Sim Jaccard (C1, S3) < Sim Jaccard (C2, S3), donc le troisième service (S3) appartient à la deuxième classe et on note : [C2, S3].
  - ❖ Classifier chaque service avec le centre le plus similaire :
    - La classe 1 est [1, 2].
    - La classe 2 est [3, 4].

Nous remarquons que les services de chaque classe ne changent pas, donc c'est un cas de stabilisation et les classes sont comme suit :

- La classe 1 est [1, 2].
- La classe 2 est [3, 4].

**7. Conclusion**

Dans ce chapitre, nous avons décrit les grands axes de la réalisation de notre application, ainsi que les résultats obtenus. Nous allons commencer par présenter les différents outils utilisés dans l'implémentation, ensuite, nous avons présenté l'interface de notre application et les principales étapes d'utilisation de notre application, enfin nous avons terminé par un exemple illustratif.

# Conclusion générale

---

## Conclusion Générale

Avec le développement rapide des technologies de l'information, la recherche de l'interopérabilité est de nos jours une problématique centrale des systèmes distribués. Ce domaine de recherche est favorisé par l'adoption de l'architecture orientée service comme modèle de développement, et particulièrement, par les services Web qui combinent les avantages de ce modèle aux langages et technologies développés pour Internet. Les services Web sont adoptés pour l'intégration et l'interopérabilité des systèmes répartis. Ils sont caractérisés par leurs indépendance aux plateformes et aux systèmes d'exploitation, ce qui a impliqué leur adoption par les différentes organisations commerciales et industrielles offrant leurs services à travers le Web, et par conséquent l'augmentation du nombre de services offerts.

Au début, les travaux traitants les services web se mettaient face au problème d'automatisation de la découverte. En effet, l'utilisation des technologies du Web sémantique afin de rendre les services interprétable par la machine est considérée comme un facteur de réussite majeur en vue de faciliter les fondations sur les services web. Dans un deuxième temps, les questions posées étaient comment améliorer la qualité de différents processus de la technologie des services web, en terme de degré de similarité avec la demande. Dans ce contexte, a défini les trois facteurs influant sur le processus de découverte : le degré de description des services, le degré de description des besoins, et l'efficacité de l'algorithme matchmaking. Actuellement, et en conséquence de la demande croissante de l'utilisation de la technologie des services web, les travaux se concentrent plus sur les solutions qui supportent la distribution et la catégorisation des services web et ce pour que la découverte soit rapide en terme du temps réponse et efficace en terme de résultat.

Dans ce dernier contexte et à travers ce projet, nous proposons de réaliser une catégorisation des services web à fin d'assister le processus de découverte. Pour cela on a employé l'algorithme K-means, malgré qu'il soit sensible à l'initialisation (nombre du cluster, et sélection des points de départ) mais généralement, il donne des bons résultats dans plusieurs domaines d'applications. Dans notre travail, les résultats obtenus doivent être comparés avec d'autres réalisés dans le même contexte et devront être évalués pour montrer leurs efficacités. Cette phase d'évaluation va être réalisée dans un futur travail.

# Références bibliographiques

---

- [1] A. Mullenders, e-DRH: Outil de gestion innovant, de Boeck Université, 2009.
- [2] [www.servicearchitecture.com](http://www.servicearchitecture.com), consulter le 18/06/2019.
- [3] M.P.Papazoglou, “What’s in a Service?”, VLDB, Springer-Verlag, 2007, pp. 11-28.
- [4] B. Manouvrier, L. Ménard, Intégration applicative EAI, B2B, BPM et SOA, Hermès Lavoisier, 2007.
- [5] <http://www.w3.org/TR/ws-arch/>, consulter le 20/06/2019.
- [6] JM. Chauvet, “Web Services avec SOAP, WSDL, UDDI, ebXML”, Eyrolles, 2002.
- [7] WSMO Web Service Modeling Ontology, 2005. <http://www.w3.org/Submission/WSMO/>
- [8] R. Akkiraju, J. Farrell, J. Miller, M. Nagarajan, M.T. Schmidt, A.Sheth, and K. Verma. Web service semantics - WSDL-S, Novembre 2005.
- [9] W3C. Joel Farrell , Holger Lausen. Semantic Annotations for WSDL and XML Schema. <http://www.w3.org/TR/sawSDL/>. W3C Recommendation. August 2007
- [10] W3C. Akkiraju, R., Farrell, J., Miller, J., Nagarajan, M., Schmidt, M.-T., ShethA., Verma, K. Web Service Semantics – WSDL-S. W3C Member Submission . <http://www.w3.org/Submission/WSDL-S>. 2005 .
- [11] Dieter Fensel, Mick Kerrigan and Michal Zaremba. Implementing Semantic Web Services: The SESA Framework. Springer-Verlag Berlin Heidelberg. 2008.
- [12] Mr. Ayad Soheyb. Une approche basée agents pour la découverte sémantique des services web. Mémoire magister. Université Mohamed Khider de Biskra., 2009
- [13] Mounzer BOUBOU : "contribution aux méthodes de classification non supervisée via des approches prétopologiques et d'agrégations d'opinion", thèse de doctorat, université Claud Bernard –Lyon1, 2007.
- [14] KOUDRI MOHAMMED “ modèle de mélange gaussien : application sur les images cytologique” Master , université Abou bakr belkaid Tlemcen ,2011
- [15] Belhabib Abdelkader ,Lagha Omar : “ Développement d’une application à base de l’algorithme de classification k-means”, Mémoire de fin d’études pour l’obtention du diplôme de Licence en Informatique, université Abou bakr belkaid, Tlemcen ,2012
- [16] Berrani, S.-A., Amsaleg, L., & Gros, P. « Recherche par similarités dans les bases de données multidimensionnelles : panorama des techniques d’indexation. » Ingénierie des systèmes d’information (RSTI série ISI-NIS), 7(5-6), pp 65-90.2002.
- [17] Lamri Laoumer : “Approche exploratoire sur la classification appliquée aux images”, Mémoire , Université du Québec à Trois-Rivières , Avril 2006
- [18] Licence Professionnelle Géomatique et Environnement : “TRAITEMENT NUMÉRIQUE DES IMAGES Classifications non supervisées”

# Références bibliographiques

---

- [19] Diday .E, " Optimisation en classification automatique et reconnaissance de formes".  
Note Scient. IRIA nO 6, 1972.
- [20] E, W Forgy cluster analysis of multivariate data : efficency versus interpretability of  
classification (abstract), Biometrics 21:768-769 ,1968
- [21] S . P . Bradley ,U.M. Fayyad ,and C. Reina . Scaling clustering algorithms to large  
databases. In knowledge Discovery and Data Mining, pages 9-15,1998.
- [22]Cel eux .G, Diday .E, Govaert .G, " Classification automatique de données  
environnement statistique et informatique". Dunod, Informatique, 1989.
- [23] Pedersen T., Patwardhan S. and Michelizzi J., "WordNet: Similarity - Measuring the  
Relatedness of Concepts". Proceedings of the Nineteenth National Conference on Artificial  
Intelligence (AAAI-04): 1024-1025. 2004
- [24] <http://www.dfki.de/scallops>, consulter le 10/07/2019.
- [25] [https://www.saedsayad.com/clustering\\_kmeans.htm](https://www.saedsayad.com/clustering_kmeans.htm), consulter le 05/09/2019.
- [26] Jaccard, P. (1901). Étude comparative de la distribution \_orale dans une portion des alpes  
et desjura. Bulletin del la Société Vaudoise des Sciences Naturelles, 37 :547\_579.
- [27] Baeza-Yates, R. A. and Ribeiro-Neto, B. (1999). Modern Information Retrieval.  
Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- [28] COURS UML – janv. 2003, <http://uml.free.fr>.