



N° Réf :.....

Centre Universitaire  
Abd elhafid Boussouf Mila

Institut des sciences et de la technologie

Département de Mathématiques et Informatiques

**Mémoire préparé en vue de l'obtention du diplôme de  
Master  
En : Informatique**

**Spécialité: Sciences et Technologies de l'Information et de la  
Communication (STIC)**

**Classification et découverte de services  
web**

**Préparé par :** - HAMMOUDI Fatima zahra  
- BOUREDIOUS Chama

**Soutenu devant le jury :**

**Encadré par :** MERARBET Adil..... M.A.A  
**Président :** LALOU Mohammed ... .. M.A.A  
**Examineur:** BOUBAKIR Mohamed .....M.A.A

**Année universitaire : 2015/2016**

# *Remerciement*

*En premier lieu,*

*Nous tenons à remercier Dieu le tout puissant, qui nous a donné la force et la patience d'accomplir ce modeste travail.*

*Nous remercions vivement Mr Merabet Adil notre encadreur pour sa présence, son aide et surtout pour ses précieux conseils qui nous ont beaucoup aidés pour l'accomplissement de ce projet.*

*Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à ce travail en acceptant de l'examiner et de l'enrichir par leurs propositions.*

*Nous tenons à exprimer nos sincères remerciements à tout le personnel de l'institut de sciences et de la technologie du Centre Universitaire de Mila surtout les enseignants qui nous ont enseigné durant ces cinq années d'étude.*

*Nous remercions également toutes les personnes qui nous ont aidés, de près ou de loin pour la réalisation de ce travail en particulier à nos parents, et nos frères, sœurs, amis qui nous ont encouragés, soutenu durant tout notre cursus.*

*Nous remercions tous les étudiants de la promotion 2015/2016 pour avoir été liés et unis tout au long de cette année et tous ceux qui ont collaborés de près ou de loin à l'élaboration de ce travail.*

*Fatima Zohra et Chama*

# *Dédicace*

*A ma chère maman que j'aime le plus au monde qui m'a donnée*

*le*

*Courage et l'espoir pour atteindre mon but,*

*Et mon père qui est toujours dans mon cœur*

*A mon frère Abd elkader*

*, Et*

*Mes sœurs Leila et Rima*

*Et toute la famille Hammoudi et Bouderbane que j'estime*

*Beaucoup.*

*Et a tous mes amis chama, Nor el houda, rokia, hamida et Ikram.*

*A mes cousines sarah et mitchi*

*A mon prof Zimamouche Riyad*

*A mon encadreur Mr Merabet Adil*

*A tous ceux et celles qui me sont chers.*

*Nous Remercions tous ceux qui nous ont*

*Aidés à réaliser ce travail*

*Fatima zohra*

# *Dédicace*

*Je dédie ce modeste travail aux êtres les plus  
chers dans ma vie,*

*À mes chers parents, ma mère Badiaa et mon père*

*Mouloud pour leur*

*amour, leurs encouragements et leurs sacrifices.*

*A mes frères Abd el djalil,*

*Amar, Yaakoub, Mousaab, Sohaib, et Mouaad*

*Et ma sœur Donia,*

*À mon cher mari Houssam et sa famille,*

*À ma famille, mes cousins et cousines*

*À mes amies*

*A mon encadreur Mr Merabet Adil*

*A tous ceux et celles qui me sont*

*Chers.*

*Chama*

# Sommaire

Remerciement .....	I
Dédicace .....	II
Dédicace .....	III
Sommaire.....	IV
Liste des figures .....	VIII
Liste des acronymes .....	X
Résumé .....	XII
Introduction générale.....	1

## *Chapitre1 : Généralités sur les Services Web*

Introduction .....	3
Partie 1.Les Services Web .....	4
1.1. Qu'est-ce qu'un Service Web ? .....	4
1.2. Pourquoi les Services Web ? .....	5
1.3. L'Architecture <i>Orientée Service</i> .....	5
1.3.1. Le concept de SOA .....	5
1.3.2. Pour quoi SOA?.....	5
1.4. Architecture des Services Web.....	5
1.4.1. Opérations dans l'architecture des Services Web .....	6
1.4.2. Les couches technologiques des Services Web .....	7
1.5. Les technologies des Services Web .....	7
1.5.1. Le XML .....	7
a. Définition.....	7
b. Structure de XML.....	7

1.5.2. Le protocole SOAP .....	8
a. Définition .....	8
b. Structure d'un message SOAP .....	8
1.5.3. Langage WSDL .....	9
a. Définition .....	9
b. Structure du fichier WSDL .....	9
c. Exemple d'un fichier WSDL .....	10
1.5.4. L'annuaire UDDI .....	10
a. Définition .....	10
b. Structure de l'annuaire UDDI .....	11
1.6. Avantage des Services Web .....	12
1.7. Inconvénients des Services Web .....	12
Partie 2. Web Sémantique .....	13
2.1. Qu'est-ce qu'un Web sémantique .....	13
2.2. Définition du service web sémantiques .....	13
2.3. Approches proposées pour la réalisation des Services Web Sémantiques .....	13
2.3.1. Le standard WSDL-S (Web Service Description Language-Semantic).....	13
2.3.2. OWL-S (Ontology Web Language for Services) .....	14
2.3.3. IRS-II (Internet Reasoning Service) .....	14
2.3.4. WSMF (Web Service Modeling Framework) .....	15
2.3.5. WSMO (Web Service Modeling Ontology) .....	15
Conclusion .....	16

## ***Chapitre 2 : Classification et Distance sémantique***

Introduction .....	17
Partie 1 : Classification des services web .....	18
1. Qu'est-ce qu'une classification des services Web ? .....	18

2. Les types de classification des services web .....	18
2.1. Le premier type de classification des services web .....	18
2.1.1. Le comportement .....	18
2.1.2. L'Ontologie et la sémantique .....	18
2.1.3. Le contexte du domaine.....	19
2.1.4. Qualité de service (QoS) .....	19
2.2. Le deuxième type de classification des services web.....	20
2.2.1. Classification manuel .....	20
2.2.2. Classification automatique .....	20
Partie 2 : Distance Sémantique.....	21
2.1. Définition de la distance sémantique.....	21
2.2. Mesure de similarité sémantique .....	21
2.2.1. Définition.....	21
2.2.2. Les différentes approches de mesure de similarité.....	22
a. Approches basées sur les arcs .....	22
b. Approches basées sur les nœuds.....	22
c. Approches hybrides .....	23
d. Approches basées sur l'espace vectoriel .....	23
e. Mesure de similarité intentionnelle .....	23
2.3. Domaine d'application .....	24
Conclusion.....	25

## ***Chapitre3 : Modélisation***

Introduction .....	26
Partie 1. Présentation de l'architecture générale de système.....	27
Partie 2 : la construction du graphe .....	28
1. Extraction des services web à partir des fichiers WSDL .....	28
1.1. Parcoure, Extraction et Prétraitement des informations d'un document WSDL .....	28
1.2. Les messages WSDL .....	29

1.3. Les opérations.....	30
1.4. Le Prétraitement du contenu d'un document WSDL .....	30
2. Le calcul de la distance sémantique .....	31
2.1.1. La distance sémantique entre deux concepts.....	31
2.1.2. La distance sémantique entre deux listes de concepts.....	32
2.1.3. La distance sémantique entre deux services .....	33
3. La construction de graphe .....	36
Partie 3 : la classification des services.....	38
Partie 4 : la découverte des services .....	40
Conclusion.....	43

## ***Chapitre4 : Implémentation***

Introduction .....	44
1. Environnements utilisés pour le développement .....	45
1.1. Langage de programmation.....	45
1.2. Environnement de programmation « NetBeans ».....	46
1.3. Les services web (WSDL).....	46
2. Implémentation .....	47
2.1. Présentation de l'application classification .....	47
2.2. Présentation de l'application découverte.....	49
Conclusion .....	52
Conclusion générale .....	53
Référence bibliographique.....	54

# *Liste des figures*

## *Chapitre 1 : Généralités sur les Services Web*

Figure 1.1 : Exemple de Service Web .....	4
Figure 1.2 : Découverte, publication et invocation des Services Web .....	6
Figure 1.3 : Exemple de la structure XML .....	7
Figure 1.4 : Structure d'un message SOAP .....	8
Figure 1.5 : Exemple d'un message SOAP Requête et Réponse .....	9
Figure 1.6 : Exemple d'un fichier WSDL .....	10
Figure 1.7 : l'annuaire UDDI .....	11
Figure 1.8 : Schéma général pour les Services Web .....	11

## *Chapitre 2 : Classification et Distance sémantique*

Figure 2.1 : Taxonomie des approches de mesure de similarité.....	22
---	----

## *Chapitre 3 : Modélisation*

Figure 3.1 : L'architecture du système .....	27
Figure 3.2 : Schéma du processus d'extraction de l'information.....	29
Figure 3.3 : Les messages WSDL .....	29
Figure 3.4 : Les opérations WSDL.....	30
Figure 3.5 : Exemple de la distance sémantique entre deux concepts.....	31
Figure 3.6 : Exemple de la distance sémantique entre deux services.....	33
Figure3.7 : Exemple d'une Correspondance entre deux services web .....	35
Figure3.8 : Exemple d'un graphe des services web .....	36
Figure3.9 : Exemple d'une Correspondance entre une requête et un service.....	41

## ***Chapitre 4 : Implémentation***

Figure 4. 1 : l'interface administrateur .....	47
Figure 4. 2 : l'interface graphe .....	48
Figure 4. 3 : l'interface classification .....	49
Figure 4. 4 : l'interface utilisateur .....	50
Figure 4. 5 : exemple de découverte .....	51

# *Liste des acronymes*

HTML: Hypertext Markup Language.

HTTP: Hypertext Transfer Protocol.

IC: Contenu Informationnel.

IRS-II: Internet Reasoning Service.

OWL: Web Ontology Language.

OWL-S: Ontology Web Language for Services.

PSMs: Problem Solving Methods.

QOS: Quality of service.

SESA: Syndicat des entreprises de sûreté aéroportuaire.

SMTP: Simple Mail Transfer Protocol.

SOA: Architecture Orientée Service.

SOAP: Simple Object Access Protocol.

SWSS: Cisco Software Support Service.

TF-ITF: Term Frequency-Inverse Document Frequency.

TKC: Top-Construction k.

UDDI: Universal Description Discovery and Integration.

UNSPSC: United Nations Standard Products and Services Code.

URI: Uniform Resource Identifier.

W3C: World Wide Web Consortium.

WSDL: Web Service Description Language.

WSDL-S: Web Service Description Language-Semantic.

WSMF: Web Service Modeling Framework.

WSMO: Web Service Modeling Ontology.

**WSQMS:** Web Services Registry for Quality Assurance.

**WSSN:** World Standards Services Network.

**XML:** eXtensible Markup Language.

# Résumé

Comme toute technologie, les *services web* génèrent beaucoup d'intérêt et promettent de nombreux avantages. Ils fournissent des fonctionnalités accessibles via les protocoles standards du web qui permettent aux entreprises d'échanger des données à distance, et une exploration facile entre les entreprises, leurs partenaires et leurs clients.

La classification des *services web* est le processus qui permet d'organiser un ensemble de *services web* selon des critères de similitude catégorielle et fonctionnelle. Elle peut permettre de faciliter, d'optimiser, d'automatiser l'efficacité et l'efficience des processus de découverte, de composition, d'exécution et de gestion des *services web*.

Dans ce travail, nous concentrons sur la façon de regrouper les *services web* en plusieurs *classes* en utilisant un algorithme de classification qui prend en entrée un graphe de services, qui est conçu à l'aide de la distance sémantique, et un algorithme de découverte pour la recherche de ces services.

**MOTS-CLES :** *Web sémantiques, SOAP, WSDL, UDDI, services web, découverte des services, la classification des services.*

## المخلص

مثل أي تكنولوجيا، خدمات الويب تولي اهتماما كبيرا للعديد من المزايا. بحيث توفر وظائف يمكن الوصول إليها عبر بروتوكولات الانترنت القياسية التي تمكن الشركات من تبادل البيانات عن بعد. وذلك بتقديم وظائف تسهل ايجاد المعلومات أو الخدمات بين الشركات والعملاء.

تصنيف خدمات الويب هو عملية تنظيم مجموعة من خدمات الويب وفقا لمعايير التشابه الفئوية و الوظيفية. ويمكن أن تساعد على تسهيل و تحسين فعالية و كفاءة عملية اكتشاف ، تكوين ، تنفيذ و تسيير خدمات الويب بطريقة الية.

في هذا السياق، تركز هذه الاطروحة على كيفية الجمع بين خدمات الويب إلى مجموعات باستخدام خوارزمية التصنيف المبنية على الرسم البياني للخدمات التي تم تصميمه باستخدام المسافة الدلالية. و خوارزمية الاكتشاف للبحث عن هذه الخدمات.

**كلمات البحث:** الويب الدلالي، SOAP، WSDL، UDDI، خدمات الويب ، اكتشاف الخدمات، تصنيف الخدمات.

# *Introduction générale*

Comme toute technologie, les services web génèrent beaucoup d'intérêt et promettent de nombreux avantages. L'architecture des services web repose sur trois standards : *WSDL* qui permet de décrire un service, *UDDI* qui permet de référencer ce service et *SOAP* qui décrit la communication avec ce service. Ils sont basés sur *XML* qui constitue la technologie utilisée pour développer les services web. Il existe aussi la notion de web sémantique qui représente une extension du Web actuel, dans lequel l'information a un sens bien défini, et permet une meilleure coopération entre les humains et les ordinateurs.

A l'heure actuelle, les travaux de recherche se concentrent principalement sur la composition, la classification et la découverte des services web.

L'idée de la composition des services web est de créer un nouveau service en combinant des services existants pour répondre aux besoins de l'utilisateur, puisque un seul service peut ne pas suffire à satisfaire tous les besoins. La plupart du temps il est nécessaire de composer plusieurs services web.

La découverte des *services web* représente un axe de recherche émergent. Au début, la découverte est faite au niveau du registre *UDDI*, elle est basée essentiellement sur la recherche syntaxique des descriptions *WSDL* des services Web. Mais avec le développement des technologies du Web sémantique, les techniques de découverte sont devenues essentiellement sémantiques.

La classification est l'opération qui consiste à organiser les *services web* publiés dans les annuaires. Elle a pour objectif d'améliorer les activités de publication, de découverte, d'invocation ou encore de maintenance. C'est une tâche qui s'avère complexe et coûteuse si elle est réalisée manuellement, étant donné le nombre important et croissant de services. Les travaux sur la classification visent à proposer des solutions pour l'automatisation de ce processus. La plupart des méthodes de classification proposées sont basées sur la similitude entre les services. Pour notre travail, la notion de similitude porte sur les opérations contenues dans les services et plus particulièrement sur les paramètres d'entrée/sortie de ces opérations.

Ce mémoire est structuré de la manière suivante :

Le premier chapitre est constitué de deux parties. Nous allons décrire en première partie les définitions les plus liées aux services Web. Ensuite nous allons expliquer la notion du Web sémantique en deuxième partie.

Dans le deuxième chapitre, nous allons définir dans une première partie la classification des services web en expliquant ses intérêts principaux ainsi que présenter les différents types de classifications. Dans une deuxième partie de ce chapitre, nous allons concentrer sur le traitement sémantique de l'information en présentant les différentes méthodes de calculs des distances sémantiques.

Dans le troisième chapitre, partie principale de notre travail, nous allons aborder la conception de notre application, en détaillant la méthode de regroupement des services web, qui est basée sur un graphe de services web sémantique.

Le quatrième chapitre présente l'implémentation de notre application en définissant dans une première partie les différents outils utilisés et dans une deuxième partie, les fonctionnalités et le mécanisme de fonctionnement de l'application à l'aide des différentes interfaces disponibles.

Enfin, nous allons terminer ce mémoire par une conclusion générale et des perspectives pour cette étude.

---

# *Chapitre 1*

## *Généralités sur les services web*

---

### ***Introduction :***

L'accès aux systèmes d'information s'appuie aujourd'hui de plus en plus sur la technologie Internet. Les efforts de standardisation dans ce contexte ont permis l'émergence des services Web comme support de développement des applications accessibles par Internet. Ainsi, les technologies associées aux services Web sont devenues incontournables pour le développement d'applications interagissant les unes avec les autres par le biais de l'Internet.

Le Web sémantique constitue le point de départ pour le développement futur de services Web. Les services Web sémantiques sont une évolution nécessaire des services Web. Ils se basent totalement sur les langages du Web sémantique, en particulier sur les ontologies.

Dans ce chapitre nous allons présenter d'une façon générale en première partie les technologies les plus liées aux services Web, ensuite nous allons expliquer la notion du Web sémantique en deuxième partie.

## ***Partie 1. Les Services Web***

### ***1.1. Qu'est-ce qu'un Service Web ?***

Un service Web est un composant logiciel représentant une fonction applicative (ou un service applicatif). Il est apparu vers 2000. Il peut être accessible depuis une autre application (un client, un serveur ou un autre service Web) à travers le réseau Internet en utilisant les protocoles de transports disponibles (i.e. SOAP sur HTTP). Ce service applicatif peut être implémenté comme une application autonome ou comme un ensemble d'applications (liées ensemble par une infrastructure d'intégration) [1].

Le consortium W3C définit un service Web comme étant : « une application, ou un composant logiciel qui vérifie les propriétés suivantes » :

- ✓ Il est identifié par un URI.
- ✓ Ses interfaces et ses liens peuvent être décrits en XML.
- ✓ Sa définition peut être découverte par d'autres services Web.
- ✓ Il peut interagir directement avec d'autres services Web à travers le langage XML en utilisant des protocoles Internet standards [2].

#### ***Exemple de Service Web:***

Un client demande la météo d'une région en envoyant un message sur le Web, ce message contient le nom d'une région et la date du jour. Le service web va recevoir le nom et la date, effectuer le traitement du service et renvoyer la météo au client via un autre message.



***Figure 1.1 : Exemple de Service Web***

### ***1.2. Pourquoi les Services Web ?***

- ✓ Faciliter l'accès aux applications entre entreprises.
- ✓ Simplifier les échanges de données.
- ✓ Réutilisation dans un environnement ouvert.
- ✓ Permettre l'interopérabilité entre des applications diverses.

### ***1.3. L'Architecture Orientée Service :***

#### ***1.3.1. Le concept de SOA :***

Actuellement, sous le vocable de SOA, se développe un style d'architecture orientée service permettant de construire des systèmes informatiques évolutifs et adaptables, en améliorant leur qualité et en simplifiant leur intégration dans l'infrastructure informatique de l'entreprise, par recours à des composants réutilisables appelés services [3].

#### ***1.3.2. Pourquoi SOA?***

Une architecture orientée services permet d'obtenir tous les avantages d'une architecture client-serveur et notamment:

- ✓ Une modularité permettant de remplacer facilement un composant (service) par un autre.
- ✓ De meilleures possibilités d'évolution (il suffit de faire évoluer un service ou d'ajouter un nouveau service).
- ✓ Une plus grande tolérance aux pannes.
- ✓ Une maintenance facilitée.

### ***1.4. Architecture des Services Web :***

Dans cette section, nous allons présenter l'architecture de base des services web. Cette architecture comporte trois entités: le fournisseur de service, l'annuaire de services et le client ou utilisateur du service [4].

- ✓ Le fournisseur de services (service providers)

Qui publie son service en fournissant la description au format WSDL dans l'annuaire de services et réalise les opérations propres au service.

- ✓ Le client (service requestor)

Qui obtient la description du service grâce à l'annuaire et utilise le service. Dans l'architecture des services web, le Client est défini comme le consommateur du service. Il peut accéder à ce dernier en échangeant avec le fournisseur des messages SOAP. Cet échange de messages se fait généralement à l'aide des protocoles Internet standards tels que HTTP, SMTP, etc. Techniquement, le Client peut être une simple application Windows ou web, comme il peut être un autre service web.

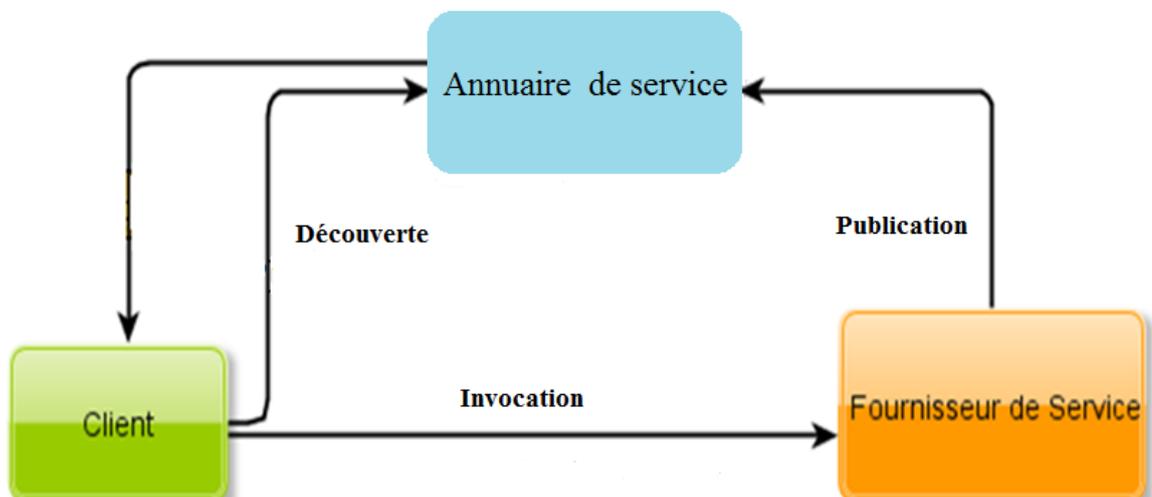
- ✓ l'Annuaire de services (discovery agency)

Qui, d'une part reçoit et enregistre les descriptions de services publiées par les fournisseurs, et d'autre part reçoit et répond aux recherches de services lancées par les clients. C'est donc un registre de description qui offre aux fournisseurs le moyen de publier et d'indexer leurs services web sur le réseau. Il permet également aux clients de rechercher ces services selon plusieurs critères.

### *1.4.1. Opérations dans l'architecture des Services Web :*

Les trois opérations majeures entourent les services Web sont :

- ✓ Publish (publication) : faire un service disponible.
- ✓ Find (recherche et découverte) : la localisation des services Web.
- ✓ Bind (invocation) : utilisant le service web.



*Figure 1.2 : Découverte, publication et invocation des Services Web*

### *1.4.2. Les couches technologiques des Services Web :*

L'architecture des Web Services repose essentiellement sur les technologies suivantes :

- ✓ **WSDL** (Web Service Description Language): Langage de description de l'interface du Service Web.
- ✓ **UDDI** (Universal Description, Discovery and Integration): Annuaire pour le référencement du Service Web.
- ✓ **SOAP** (Simple Object Access Protocol) : Protocole de communication aux Services Web par échange de messages XML.

### *1.5. Les technologies des Services Web:*

#### *1.5.1. Le XML:*

##### *a. Définition :*

XML (Extensible Markup Language, ou Langage Extensible de Balisage) est le langage destiné à succéder HTML. Comme HTML (Hypertext Markup Language) c'est un langage de balisage (markup). XML est un métalangage, ce qui veut dire que contrairement à HTML qui possède un ensemble de balises de présentation prédéfinies, il va permettre d'inventer de nouvelles balises d'isolement d'informations que peut contenir une page Web [5].

##### *b. Structure de XML :*

```
<livre>
  <titre> le super livre </titre>
  <chapitre>
    <numero> 1 </numero>
    <titre> titre du chapitre 1 </titre>
    <contenu> aaaaaaaaaaaaaa </contenu>
  </chapitre>
  <chapitre>
    ..
  </chapitre>
</livre>
```

*Figure 1.3 : Exemple de la structure XML*

### 1.5.2. Le protocole SOAP :

#### a. Définition :

SOAP est un protocole de communication entre applications basé sur le langage XML. On utilise un protocole de transport pour véhiculer les messages SOAP de l'émetteur au récepteur comme http [6].

L'objectif de SOAP est:

- ✓ Assurer la communication entre les applications d'une même entreprise (intranet).
- ✓ Assurer les échanges interentreprises entre les applications et les services web.

#### b. Structure d'un Message SOAP :

Un message SOAP est un document XML constitué d'une enveloppe composée de deux parties :

- ✓ Un en-tête (header) qui peut être facultatif : est utilisé pour transmettre des informations supplémentaires sur ce même message.

L'en-tête est défini par la balise `<SOAP-ENV: Header>`.

- L'élément peut être facultatif
- Doit être placé avant le corps
- ✓ Un corps (body) : est constitué d'un élément `<SOAP-ENV: Body>`.

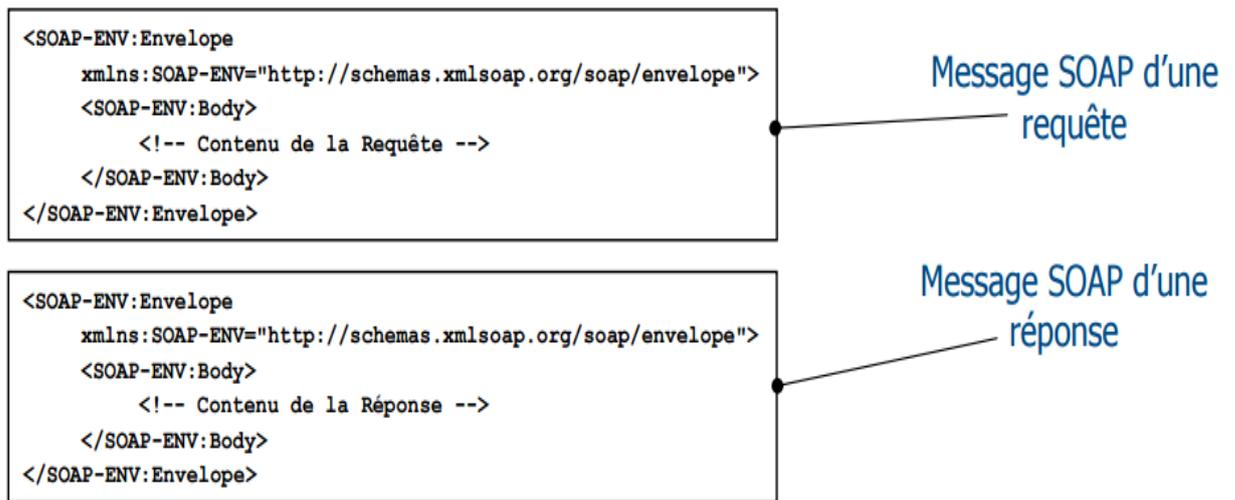
L'élément `<SOAP-ENV: Body>` peut contenir soit :

- Une erreur en réponse à une requête (élément).
- Des informations adressées au destinataire du message SOAP respectant un encodage déterminé.



Figure 1.4 : Structure d'un message SOAP.

La requête et la réponse ont la même structure.



*Figure 1.5 : Exemple d'un message SOAP Requête et Réponse*

### 1.5.3. Langage WSDL :

#### *a. Définition :*

WSDL (*Web Services Description Language*) est un langage de description standard. C'est l'interface présentée aux utilisateurs. Il indique comment utiliser le service web et comment interagir avec lui. WSDL est basé sur XML et permet de décrire de façon précise les détails concernant le service Web tels que les protocoles, les ports utilisés, les opérations pouvant être effectuées, les formats des messages d'entrée et de sortie et les exceptions pouvant être envoyées [7].

#### *b. Structure du fichier WSDL :*

- ✓ Définition : racine d'un document WSDL.
- ✓ Type : contient la définition des types de données exprimées sous forme XML.
- ✓ Message : décrit des messages à transmettre.
- ✓ PortType: décrit un ensemble d'opérations ou chacune a 0 ou plusieurs messages en entrée, 0 ou plusieurs messages en sortie ou de fautes.
- ✓ Binding : spécifie une liaison portType à un protocole (soap, http).
- ✓ Service : regroupe l'ensemble des ports [8].

### *c. Exemple d'un fichier WSDL :*

```
----- HelloWorld.wsdl -----
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="HelloWorld"
  targetNamespace="http://hello.jaxrpc.samples/"
  xmlns:tns="http://hello.jaxrpc.samples/"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
  <types />
  <message name="sayHello">
    <part name="String_1" type="xsd:string" />
  </message>
  <message name="sayHelloResponse">
    <part name="result" type="xsd:string" />
  </message>
  <portType name="Hello">
    <operation name="sayHello" parameterOrder="String_1">
      <input message="tns:sayHello" />
      <output message="tns:sayHelloResponse" />
    </operation>
  </portType>

```

*Figure 1.6 : Exemple d'un fichier WSDL*

### **1.5.4. L'annuaire UDDI:**

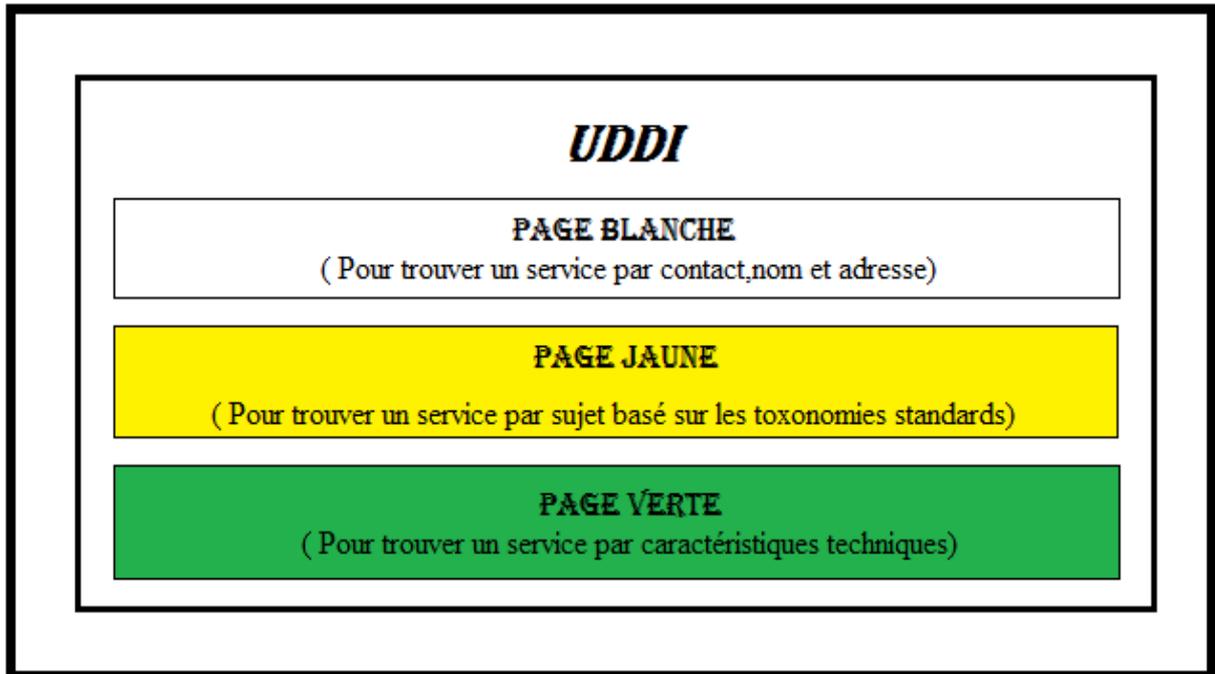
#### *a. Définition :*

Maintient un référentiel qui stocke des informations sur les services disponibles et leurs emplacements. Il est contacté par le service fournisseur, qui annonce ses services et informations de contact. Annuaire de services est interrogé par les consommateurs de services pour obtenir l'emplacement d'un service.

Les informations qu'il contient peuvent être séparées en trois types [9] :

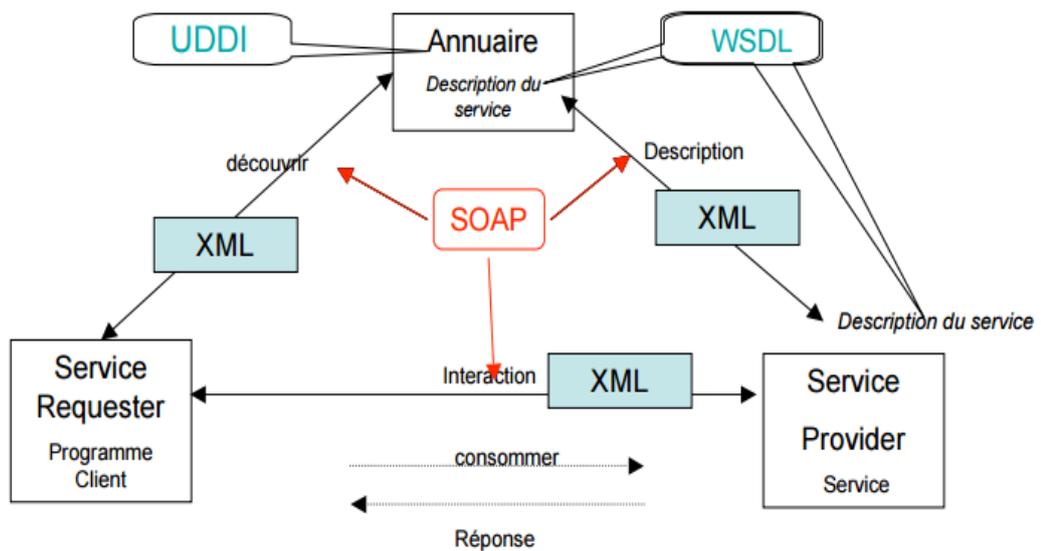
- ✓ Les pages blanches (White Paper). Ce composant permet de connaître les informations à propos de l'organisation proposant le service. Cette description contient toutes les informations pour identifier l'organisation (telles que son nom, son adresse physique).
- ✓ Les pages jaunes (Yellow Paper). Les pages jaunes d'UDDI détaillent la description de l'organisation : la catégorie de l'entreprise, le type de services et les conventions d'utilisation (prix, qualité de service,...etc.).
- ✓ Les pages vertes (Green Paper). Les pages vertes comportent les informations techniques comme comment accéder à un service Web.

*b. Structure de l'annuaire UDDI :*



*Figure 1.7 : l'annuaire UDDI*

Ce schéma represente généralement les services web.



*Figure 1.8 : Schéma général pour les Services Web*

### ***1.6. Avantages des Services Web:***

- ✓ La productivité.
- ✓ Le franchissement des Firewalls.
- ✓ La normalisation des protocoles et des interfaces par les consortiums W3C.
- ✓ L'appel synchrone (bloquant) et asynchrone. [10].

### ***1.7. Inconvénients des Services Web:***

- ✓ La faible performance à cause du codage des données au format XML, car les fichiers XML sont souvent de très gros fichiers et ceci entrainera une lourdeur considérables.
- ✓ Le manque de la sémantique (le format WSDL décrit les services d'une manière syntaxique.)
- ✓ La sécurité n'est pas garantie à 100%.Par exemple : L'utilisation du protocole HTTP, les services Web peuvent contourner les mesures de sécurité mise en place à travers des Firewalls. [10].

## ***Partie 2. Web Sémantique***

### ***2.1. Qu'est-ce qu'un Web sémantique :***

Le Web Sémantique est une extension du Web actuel, dans lequel l'information a un sens bien défini, et permet une meilleure coopération dans le travail entre les humains et les ordinateurs... « Les données web qui peuvent être traitées par les ordinateurs ».

Le Web sémantique est donc une nouvelle approche pour l'organisation du contenu du Web instaurée dans le but d'améliorer l'interopérabilité, la découverte et la récupération des ressources [11].

### ***2.2. Définition du service web sémantiques :***

Les services web sémantiques sont des services web décrits de telle sorte qu'un agent logiciel puisse interpréter les fonctionnalités offertes par le service web. Un agent logiciel doit être capable de lire la description d'un service web pour déterminer si le service web fournit les fonctionnalités désirées, et s'il est lui-même capable d'utiliser ce service. Pour aboutir à cette fin, la description du service web doit être complétée en information sémantique interprétable par machine. Les paramètres du service web doivent être décrits de façon qu'un agent logiciel puisse connaître leur signification. Cela peut s'accomplir en définissant des vocabulaires organisés en ontologies [12].

### **2.3. Approches proposées pour la réalisation des services web sémantiques:**

En littérature, diverses approches ont été proposées pour permettre la réalisation des services web sémantiques. Parmi ces approches nous avons : WSDL-S, OWL-S, IRS-II, WSMF et WSMO [12].

#### ***2.3.1. Le standard WSDL-S (Web Service Description Language-Semantic) :***

WSDL-S est un langage de description sémantique des services web. Une description WSDL-S de service web est une description WSDL augmentée de sémantique, cette sémantique est ajoutée en deux étapes : La première étape consiste à faire référence (dans la partie définition de WSDL), à une ontologie dédiée au service à publier. La seconde étape

consiste à annoter les opérations de la définition WSDL de sémantique en ajoutant deux nouvelles balises [13]: balise action et balise contrainte.

- ✓ La balise Action : représente l'action de l'opération.
- ✓ La balise Contrainte : représente les prés et post conditions d'une opération.

### **2.3.2. OWL-S (*Ontology Web Language for Services*):**

OWL-S est un langage d'ontologie destiné aux services web. Il est basé sur le langage OWL. Cette ontologie a pour objectif de d'écrire de façon non ambiguë les services web, de telle sorte qu'un agent logiciel puisse exploiter automatiquement ces informations. OWL-S permet : La découverte automatique, la composition et l'interopérabilité de services web ainsi que la surveillance automatique de leurs exécutions [14].

OWL-S décrit un service à l'aide des trois classes définie ci-après:

- ✓ ServiceProfile : définit le service web.
- ✓ ServiceModel : définit le fonctionnement du service web.
- ✓ ServiceGrounding : définit la manière d'accéder au service web.

### **2.3.3. IRS-II (*Internet Reasoning Service*):**

IRS-II est une architecture pour les services web sémantiques. Où diverses ontologies sont définies [15] :

- ✓ Ontologie du domaine (Domain model) : permet de décrire le domaine d'une application (véhicules, maladies.....).
- ✓ Ontologie de tâche à résoudre (Task models) : fournit une description générique de la tâche à résoudre, spécifie les types d'entrées (input) et sortie (output), le but à atteindre et les prés conditions à satisfaire.
- ✓ Ontologie des méthodes de résolution d'un problème (Problem Solving Methods (PSMs)): sépare la description de ce qu'un service fait des paramètres et des contraintes d'une mise en œuvre particulière.
- ✓ Liens (bridges) : permettent la correspondance entre les différents modèles d'une application.

Les principaux composants de l'architecture IRS-II sont : le serveur IRS-II (IRS-II server), l'éditeur de services (IRS-II Publisher) et la partie client (IRS-II Client). Ces trois composants interagissent entre eux via le protocole SOAP.

### **2.3.4. WSMF (Web Service Modeling Framework):**

WSMF est un modèle de représentation des divers aspects relatifs aux services web. L'objectif principal de cette approche est de permettre le développement du commerce électronique par application des technologies du web sémantique aux services web. Le WSMF se base sur quatre éléments : les ontologies fournissant la terminologie utilisée par les autres éléments ; les répertoires d'objectifs définissant les problèmes qui doivent être résolus par les services web ; les descriptions des services web et un ensemble de médiateurs contribuant à outrepasser les problèmes d'interopérabilités [16].

### **2.3.5. WSMO (Web Service Modeling Ontology):**

Le WSMO est un projet de l'union européenne qui constitue un cadre compréhensible pour SESA et définit un modèle conceptuel avec un langage de spécification, comme, il fournit une implémentation avec plusieurs outils [17].

### ***Conclusion :***

Les services web sont suffisamment développés pour que les développeurs les utilisent maintenant dans tous les domaines de l'informatique, afin de récolter les divers bénéfices de la technologie. La diversité de ces infrastructures et des organismes qui les déploient entraîne des hétérogénéités.

Un service web sémantique est un service web décrit, en utilisant des annotations sémantiques dans un langage du Web sémantique bien défini, qui permettent au service web d'avoir une interface compréhensible par les humains et les machines. Ces services web sémantiques s'appuient en général sur les langages du web sémantique, pour décrire leurs fonctionnalités et les données qu'ils échangent.

---

## *Chapitre 2*

# *Classification et Distance Sémantique*

---

### ***Introduction***

Trouver des services web adaptés aux clients ou aux développeurs de systèmes orientés services exige certaines compétences jusqu'à maintenant, et prend un temps remarquable, même si le nombre de services web n'est pas très grand.

De plus, il existe un grand nombre de résultats retournés, et qui n'ont aucun rapport avec les besoins de l'utilisateur. Par exemple, si un utilisateur est à la recherche d'un service web gratuit pour la prévision de la météo, les résultats récupérés sont des services web avec le terme "free" dans la description, sans confirmer que ce service web est en fait gratuit d'utiliser ou non dans le sens de libre. D'ailleurs, l'utilisateur a besoin de lire la description des services web récupérés afin de confirmer s'il est gratuit ou commercial.

La classification des services web est une méthode utilisée aussi afin d'améliorer la vitesse du processus de découverte. En plus, elle augmente la précision de la découverte des services appropriés à des besoins spécifiés.

Dans ce chapitre nous allons présenter dans la première partie pour quoi faire une classification des services web et expliquer brièvement les différents types de classification, ensuite nous allons expliquer la notion de la distance sémantique en deuxième partie.

## ***Partie 1 : Classification des services web***

### ***1. Qu'est-ce qu'une classification des services Web ?***

La classification des services web est l'acte de groupement des services web similaires en différents groupes. La similitude entre les services web d'un groupe dépend de différents critères, ce qui conduit à différentes méthodes de classification.

Voici les critères utilisés pour la classification des services web dans des publications récentes [18].

### ***2. Les types de classification des services web :***

#### ***2.1. Le premier type de classification des services web :***

##### ***2.1.1. Le comportement :***

C'est une classification des services Web qui se base sur les fonctionnalités fournies par chaque service. Par exemple, les services Web d'information sont une catégorie de services web qui fournissent des informations aux utilisateurs sans entrées requises. Comme les prévisions météorologiques, taux de change de l'argent, et les services internationaux de temps.

*Hongbing* propose un algorithme de classification automatique, ce dernier dépend de certaines caractéristiques fonctionnelles telles que les entrées et les sorties extraites des services web (extraites à partir des fichiers de description des services). Les critères de classification de *Hongbing* dépendent d'une taxonomie normalisée par *UNSPSC* [19].

##### ***2.1.2. L'Ontologie et la sémantique :***

*Crasso* [20] propose une classification des services web qui est basée sur les ontologies et inspirée par un modèle vectoriel. Chaque mot dans le fichier de description de services web

a une pondération basée sur le schéma *TF-ITF*. Son travail montre une flexibilité dans la gestion de la description des services web, même si les sources de mots représentant les services web ne sont pas déterminé avec précision.

Un inconvénient de la classification à base d'ontologies réside sur le fait que les différentes ontologies qui existent peuvent provoquer des classifications différentes pour les mêmes services web, ce qui peut conduire à l'ambiguïté dans la découverte des services web.

*Kehagias* [21] a proposé une classification basée sur la sémantique des fichiers *WSDL* à travers 3 couches de catégorisation. Son travail a déployé *WordNet* [22] comme une des références de l'ontologie et *PorterStemmer* [23] pour l'extraction des mots-clés à partir des fichiers de description *WSDL*.

### **2.1.3. Le contexte du domaine :**

*Abujarour et al* [24] ont proposé un algorithme automatique de classification qui permet de récupérer à partir du Web les fichiers de description de services web analysés, puis traiter les caractéristiques de chaque service web en utilisant la fonction *SimHash* [25] et classer les services web en fonction de leurs domaine.

Cette classification a montré de bons résultats dans le regroupement des services Web. C'est parce que les utilisateurs en général recherchent un service web en recherchant son domaine.

### **2.1.4. Qualité de service (QoS):**

*Lee et al* [26] dans leurs travaux ont proposés un système de gestion de la qualité de service web, appelé *WSQMS*. Ils ont intégrés ce facteur avec le registre *UDDI* comme un nouveau composant pour relier chaque service web avec le paramètre de sa qualité de service, et ensuite utiliser et déployer ces paramètres pour la classification des services web.

Le problème de la catégorisation *QOS* est l'absence de la sémantique, ce qui est très nécessaire à l'algorithme de découverte. Cependant, la catégorisation *QOS* est très utile pour le regroupement et le filtrage des services web, ce qui aide beaucoup l'utilisateur final à la

prise de décision dans le but de choisir un service web parmi un groupe de services ayant des fonctionnalités similaires.

### ***2.2. Le deuxième type de classification des services web :***

#### ***2.2.1. Classification manuel :***

Chaque service web publié sur le web doit être enregistré dans le registre *UDDI*. Une fonction de classification manuelle des services web est implémentée au niveau de ce dernier. Cependant, avec l'augmentation des services web, le coût de cette classification manuelle des services est de plus en plus élevé, et la classification est devenue de plus en plus difficile. Donc, La communauté scientifique a commencé la recherche sur des méthodes automatique de classification [27].

#### ***2.2.2. Classification automatique :***

Un effort considérable a été fait pour développer des méthodes automatiques ou semi-automatiques pour classer les services Web selon leur domaine d'application. Les solutions proposées peuvent faire appel à des techniques d'apprentissage supervisé ou d'apprentissage non supervisé communément appelées techniques de « clustering » [28].

## ***Partie 2 : Distance Sémantique***

La sémantique c'est le sens, c'est ce qu'on appelle aussi parfois le « fond » par opposition à la forme. Par exemple si nous comparons les mots « voiture » et « véhicule » d'un point de vue sémantique ce sont des mots relativement proches par leur sens. Nous pouvons en général remplacer le mot voiture par le mot véhicule, cela ne changera pas le sens du discours. La sémantique est naturelle et facile pour un humain, par contre un ordinateur est incapable de voir la proximité de sens entre ces deux mots.

### ***2.1. Définition de la distance sémantique :***

La distance sémantique est une valeur attribuée à deux concepts qui dépend de la similarité entre ces derniers. Cette distance sémantique est calculée selon plusieurs moyens comme nous allons voir dans cette partie.

Notez que cela ne veut pas dire que l'ordinateur va comprendre réellement le sens du texte, mais il sera capable de les comparer et de mesurer leurs similitudes ou différences sémantiques.

### ***2.2. Mesure de similarité sémantique :***

#### ***2.2.1. Définition :***

L'objectif des mesures de similarité sémantique est d'évaluer la proximité sémantique entre les concepts. Le calcul de similarité entre deux concepts permet de déterminer s'ils sont similaires ou dissimilaires c'est-à-dire s'ils atteignent un certain niveau de ressemblance. Ces deux concepts peuvent être également liés sémantiquement par des relations lexicales comme : les synonymes, l'antonymie, la spécialisation, ... etc [29].

### 2.2.2. Les différentes approches de mesure de similarité :

Dans cette partie nous définissons les approches principales de la mesure de similarité. La figure suivante montre une classification de différentes approches ainsi que quelques méthodes concrètes [29].

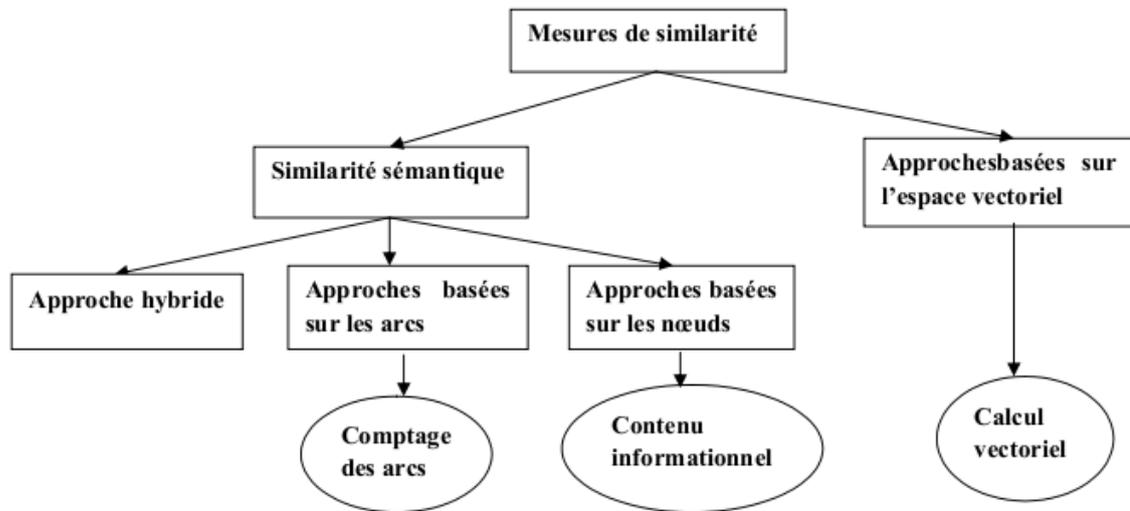


Figure 2.1 : Taxonomie des approches de mesure de similarité

#### a. Approches basées sur les arcs :

Ce type de mesures se sert de la structure hiérarchique de l'ontologie qui est représentée par un graphe dont les nœuds sont des concepts, et les arcs sont les liens entre ces concepts, et cela, pour déterminer la similarité sémantique entre les concepts qui par exemple peut être calculée à partir du nombre des arcs qui séparent les deux concepts dans le graphe de l'ontologie.

Parmi les travaux classifiés sous cette approche on peut citer : Mesure de *Wu & Palmer*, Mesure de *Rada et al*, Mesure de *Ehrig et al*, Mesure de *Hirst-St.Onge*, et la mesure de *Zargayouna*.

#### b. Approches basées sur les nœuds :

Cette approche prend en considération le contenu informationnel (IC) des concepts de l'ontologie.

## Chapitre:02 ..... Classification et Distance sémantique

---

La similarité est alors calculée à partir de l'information partagée par les concepts. Le contenu informatif est défini par :

$$IC(C) = -\log(P(c))$$

Où nous avons  $c$  un concept, et  $p(c)$  la probabilité de lui trouver un de ses descendants dans le corpus.

Parmi les mesures basées sur le contenu informationnel on peut citer : Mesure de *Resnik*, Mesure de *Lin*, Mesure de *Seco et al.*

### ***c. Approches hybrides :***

Ces approches sont fondées sur un modèle mixte qui combine les approches basées sur les arcs et les approches basées sur les nœuds c'est-à-dire le contenu informationnel qui est considéré comme facteur de décision.

On peut citer : Mesure de *Jiang et Conrath*, Mesure de *Leacock et Chodorow*, Mesure de *Li et al* et Mesure de *FaITH*.

### ***d. Approches basées sur l'espace vectoriel :***

Ces approches utilisent un vecteur caractéristique, dans un espace dimensionnel pour représenter chaque objet et calculent la similarité en se basant sur la mesure de *cosine* ou la distance *euclidienne*. La définition de la similarité entre deux vecteurs d'objets est obtenue par leurs contenus internes.

Parmi les approches citées dans la littérature on peut noter : Similarité de *Cosine*, Similarité de *Jaccard* et Similarité de *Dice*.

### ***e. Mesure de similarité intentionnelle :***

D'un point de vue ensembliste, deux entités sont similaires si le cardinal de l'intersection des ensembles de leurs caractéristiques est plus grand que celui des sous-ensembles restants.

On peut citer : Mesure de *Tversky*.

### ***2.3. Domaine d'application :***

Afin de connaître l'importance de la mesure de similarité, il est nécessaire de rappeler les indications de son utilisation [29].

- ✓ Traitement du langage naturel.
- ✓ Bioinformatique.
- ✓ Services Web.

### *Conclusion*

Le nombre des services web accessible par le public est récemment augmenté de façon considérable. Le premier problème posé est la vitesse du processus de découverte.

La Classification des services web ajoute des avantages remarquables au processus de découverte où elle peut aider à catégoriser les services Web dans le but de minimiser le champ de découverte ce qui contribue à l'amélioration de la vitesse de ce processus.

Dans cette dernière partie nous avons présenté les grandes familles et les différentes approches pour le calcul de la similarité sémantique.

---

*Chapitre 3*  
*Modélisation*

---

## ***Introduction***

La classification des *services web* est le processus qui permet d'organiser un ensemble de services web selon des critères de similitude.

Dans ce chapitre, nous construisons un graphe des services web sémantique basé sur les caractéristiques extraites à partir des documents *WSDL* et le calcul de la distance sémantique entre les différents services.

Et puis utiliser une méthode de regroupement des services web pour créer un ensemble de classes. Ensuite utiliser ces classes résultats de l'étape précédente et la requête client comme entrées au processus de découverte des services web.

**Partie 1. Présentation de l'architecture générale du système**

Afin d'assurer une modélisation cohérente de notre système, nous avons commencé par la création d'un certain nombre de modules où chacun d'eux assure des fonctionnalités distinctes, le regroupement de ces modules nous a permis par la suite de construire l'architecture générale du système. Ces modules sont en coopération et en collaboration entre eux où chaque module assure ses tâches et fournit ses sorties comme des entrées à un autre module.

Là (figure 3.1) présente les différents modules composants de notre architecture ainsi que la liaison entre eux. C'est l'architecture du système que nous proposons.

- ✓ Le Premier module prend en charge l'extraction de l'information à partir des documents WSDL.
- ✓ Le deuxième module s'intéresse essentiellement à construire un graphe des services en utilisant les différents caractéristiques et le calcul des distances sémantiques.
- ✓ Ensuite, le troisième module est l'algorithme de classification, qui donne comme résultats un ensemble de classes des services web.
- ✓ Et enfin, nous expliquons comment la classification des services web est utilisée au niveau de l'algorithme de découverte.

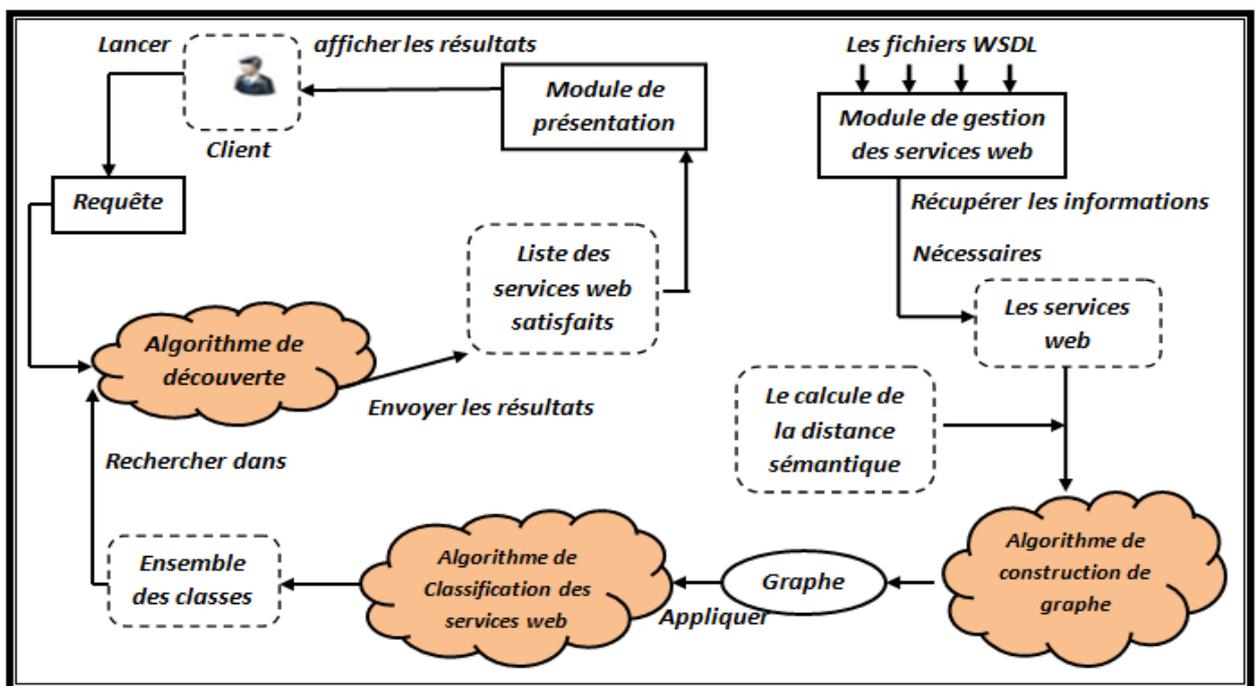


Figure 3.1 : L'architecture du système.

## ***Partie 2 : la construction du graphe***

Dans cette partie nous allons construire le graphe selon l'algorithme suivant :

<b>Algorithme</b> : Construction du graphe des services web.
<b>Entrés</b> : les fichiers WSDL.
<b>Sorties</b> : un graphe des services web.
<ol style="list-style-type: none"><li>1. Extraction des services web à partir des fichiers WSDL.<ol style="list-style-type: none"><li>1.1. Parcoure et extraction des informations d'un document WSDL.</li><li>1.2. Prétraitement des informations extraites.</li></ol></li><li>2. Le calcul de la distance sémantique entre les services web.</li><li>3. L'assemblage des services web dans un graphe.</li></ol>

### ***1. Extraction des services web à partir des fichiers WSDL :***

Dans cette partie, nous allons expliquer comment extraire les caractéristiques des services web à partir d'un document WSDL.

#### ***1.1. Parcoure, Extraction et Prétraitement des informations d'un document WSDL:***

C'est l'extraction des informations utiles : cette étape consiste, grâce à des règles, à identifier et extraire les termes pertinents d'un fichier *XML* où se trouve la définition du service web.

Dans cette étape, nous décrivons comment nous extrairons les détails des quatre informations proposées par le document WSDL.

La *figure 3.2* illustre les étapes de l'extraction des informations.

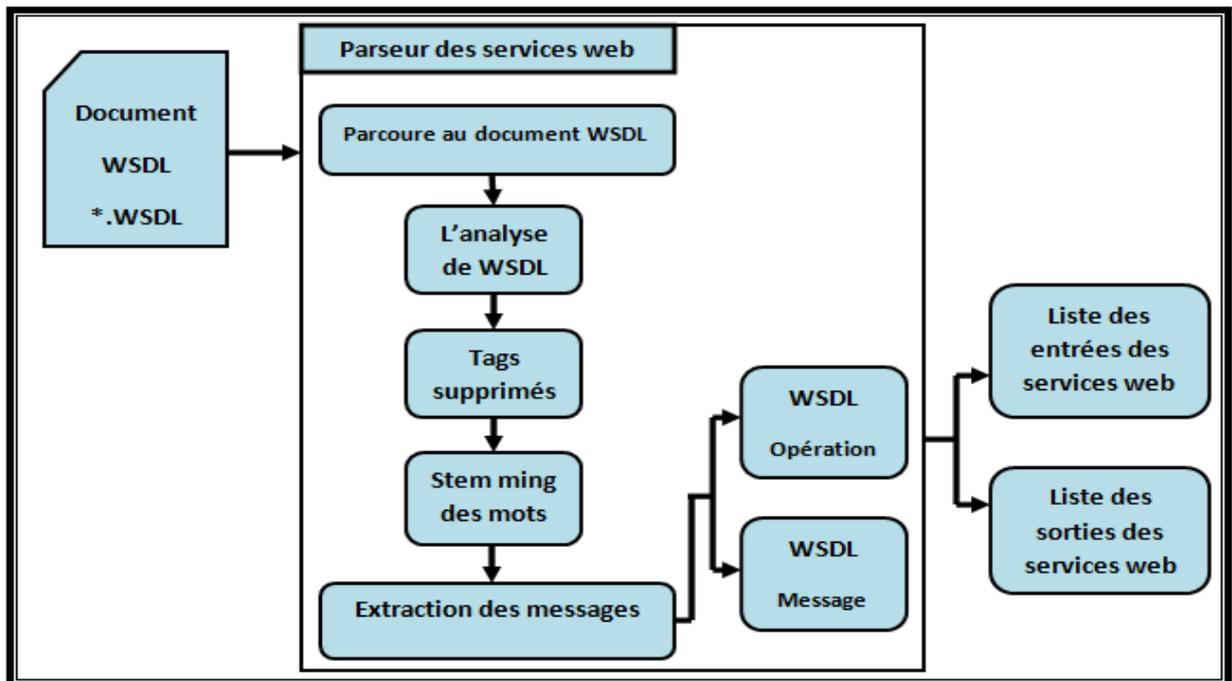


Figure 3.2 : Schéma du processus d'extraction de l'information

### 1.2. Les messages WSDL :

Cet élément dans les documents WSDL décrit les noms et les types des messages échangés par les services.

**Exemple** (figure 3.3): le nom et le type du message :

*nom="\_PRICE" type="tns:PriceType"*

*nom="\_USER" type="tns:UserType"*

La définition du type se trouve aussi dans le même document WSDL en utilisant la balise *type* (Il peut être entier, réel, ... ou même des types complexes).

```

25 <wsdl:message name="get_PRICEResponse">
26   <wsdl:part name="_PRICE" type="tns:PriceType">
27 </wsdl:part>
28 </wsdl:message>
29 <wsdl:message name="get_PRICERequest">
30   <wsdl:part name="_USER" type="tns:UserType">
31 </wsdl:part>
32 </wsdl:message>
  
```

Figure 3.3 : Les messages WSDL

### 1.3. Les opérations:

Le format des messages échangés est :

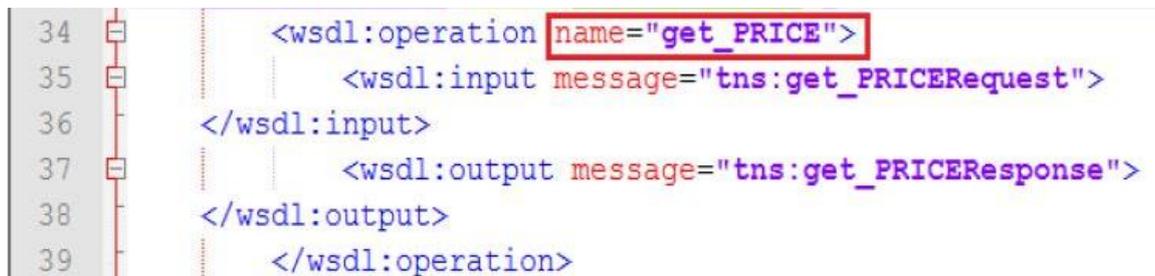
**Format Request-response (Demande-réponse):** le service reçoit un message de requête (entrée) et renvoie un message de réponse (sortie).

En *WSDL*, pour chaque *service web* il existe une séquence d'entrées et une séquence de sorties.

**Exemple** (figure 3.4):

*L'entrée du message="tns:get\_PRICERequest"*

*La sortie du message="tns:get\_PRICEResponse"*



```
34 <wSDL:operation name="get_PRICE">
35   <wSDL:input message="tns:get_PRICERequest">
36 </wSDL:input>
37   <wSDL:output message="tns:get_PRICEResponse">
38 </wSDL:output>
39 </wSDL:operation>
```

**Figure 3.4 : Les opérations WSDL**

### 1.4. Le Prétraitement du contenu d'un document WSDL:

Nous traitons le contenu du document WSDL afin d'extraire un vecteur des mots d'un service Web  $S_i$ .

Pour construire ce vecteur nous suivons trois étapes:

*a. L'Analyse du document WSDL:*

Le contenu du document est analysée par un parcours simple pour construire un vecteur de mots  $T_i$  tel que tous les concepts trouvés doivent appartenir au dictionnaire des mots.

*b. La Suppression des étiquettes :*

L'étape suivante est de supprimer tous les mots de  $T_i$  qui font partie d'une balise XML ou les mots de séparation comme : de, dans, à, ... afin que seulement les mots avec un sens valide restent dans le vecteur.

*c. Stemming des mots :*

Dans cette étape, tous les mots restants de  $T_i$  sont réduits à leurs mots de base ou racine en utilisant l’algorithme de *Porter* [30]. Les mots de même origine ont généralement le même sens.

Par exemple « *pay* » « *paid* » « *paying* » et « *payment* » tous sont de même origine « *pay* »

**2. Le calcul de la distance sémantique :**

La distance sémantique est une valeur attribuée à deux concepts (deux termes ou mots) qui dépend de la similarité entre ces deux derniers pour permettre d’évaluer à quel point sont proches.

**2.1.1. La distance sémantique entre deux concepts :**

Plusieurs formules de calcul de similarité sont disponible dans la littérature, nous avons utilisé les travaux de « *Elias Iosif, Alexandros Potamianos* » [31].

Les auteurs de ce papier ont utilisé leur formule pour créer *semsim*, une base de similarité entre les concepts.

Les espaces sémantiques sont représentés comme des matrices de similarité. *SeMSim* fournit une matrice carrée de 8752 lignes et 8752 colonnes où chaque ligne et chaque colonne représente un mot (un concept). L’intersection de la ligne  $I$  avec la ligne  $J$  donne la similarité entre le concept  $C_I$  et le concept  $C_J$ . Donc, la matrice donne toutes les distances sémantiques possibles.

Par exemple la distance sémantique entre les deux concepts « *abacement* » et « *abandon* » est 0.293512 (figure 3.5)

concept1 ▲	concept2	value
abacement	abandon	0.293512
abacement	abandonment	0.326023
abacement	abacement	1.000000

**Figure 3.5 : Exemple de la distance sémantique entre deux concepts**

**2.1.2. La distance sémantique entre deux listes de concepts :**

Nous allons calculer la distance sémantique entre deux listes de concepts  $L_1$  et  $L_2$  comme suit :

A chaque fois on prend un concept  $C_1$  de  $L_1$  et calculer la distance entre  $C_1$  et tous les concepts  $C_i$  de  $L_2$  et puis prend la moyenne entre les distances obtenues.

Exemple:

$L_1$  (urban, area)

$L_2$  (future, position)

Soit la fonction  $Dist\_C1\_C2$  qui permet de calculer la distance sémantique entre deux concepts :

La distance  $Dist\_C1\_C2$  (urban, future) +  $Dist\_C1\_C2$  (urban, position)  $\rightarrow$  Moyd = 0.005

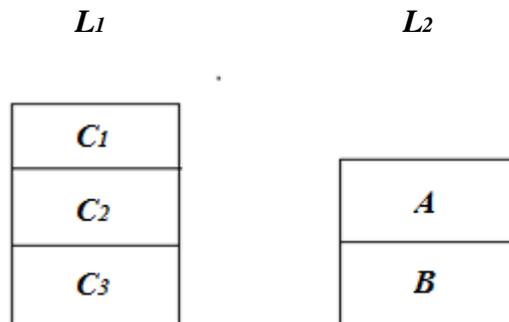
La distance  $Dist\_C1\_C2$  (area, future) +  $Dist\_C1\_C2$  (area, position)  $\rightarrow$  Moyd = 0.0015

De façon générale on applique la formule suivante [32]:

$$distance\_moyenne(L1, L2) = \sum_{x=1}^m \sum_{y=1}^n \frac{Dist\_C1\_C2(Cx, Cy)}{(n*m)}$$

\* Explication de la formule par un exemple :

Si nous voulons calculer la distance sémantique entre la liste  $L_1$  et la liste  $L_2$  tel que :



$$\frac{dist(C1,A) + dist(C1,B)}{2} = \frac{x}{2}$$

$$\frac{dist(C2,A)+ dist(C2,B)}{2} = \frac{y}{2}$$

$$\frac{dist(C3,A)+ dist(C3,B)}{2} = \frac{z}{2}$$

Puis :

$$\frac{\frac{x}{2} + \frac{y}{2} + \frac{z}{2}}{3} = \frac{\frac{x + y + z}{2}}{3} = \frac{x + y + z}{2 \cdot 3}$$

Enfin pour obtenir la distance entre les deux listes on va prendre la moyenne, nous allons calculer la moyenne des deux moyennes

$$distance\_moyenne = (0.005 + 0.0015) / 2 = 0.00325$$

Soit la fonction *Dist\_L1\_L2* qui permet de calculer la distance sémantique entre deux listes de concepts, donc, *Dist\_L1\_L2* (« urban, area », « future, position ») = 0.00325

### 2.1.3. La distance sémantique entre deux services :

Pour calculer la distance sémantique entre deux services  $S_1$  et  $S_2$  nous allons rendre chaque service sous forme d'une liste de concepts en entrée et une liste de concepts en sortie et puis appliquer la formule qui calcule la distance entre deux listes comme nous avons expliqué précédemment et comme montré dans l'exemple suivant (figure 3.6)

description	input	output
Kohl service returns list of prices of the given ...	car person bicycle	price
This service returns renting price of the pair of...	car person bicycle	price

Figure 3.6 : Exemple de la distance sémantique entre deux services

*Le service  $S_1$  : 4WheeledCar1PersonBicyclePrice service*

*[wheeled, car, person, bicycle] → [price]*

*Le service  $S_2$  : Kohl Car1PersonBicyclePrice service*

*[car, person, bicycle] → [price]*

Dans notre exemple :

- *Le service 1 est « 4WheeledCar1PersonBicyclePrice service » qui retourne une liste des prix des bicyclettes et des voitures à quatre roues.*
- *Le service 2 est « Kohl Car1PersonBicyclePrice service » qui retourne une liste des prix des bicyclettes et des voitures.*

De nouveaux algorithmes sont proposés pour mesurer la distance sémantique entre deux services. Dans cette étape nous allons expliquer brièvement la formule que nous avons utilisée :

Pour faire une correspondance entre ces deux services, il y a quatre listes de concepts comme montré dans la (figure 3.7)

Ces quatre listes sont:

- *$S_1\_Input$*  : La liste des concepts en entrée du service  $S_1$ ,
- *$S_1\_Output$*  : La liste des concepts en sortie du service  $S_1$ ,
- *$S_2\_Input$*  : La liste des concepts en entrée du service  $S_2$ ,
- *$S_2\_Output$*  : La liste des concepts en sortie du service  $S_2$ .

Soit la fonction  *$Dist\_S1\_S2$*  qui permet de calculer la distance sémantique entre deux services web. L'objectif principal est d'obtenir la distance sémantique moyenne.

Nous utilisons la fonction  *$Dist\_L1\_L2 (L_1, L_2)$*  pour calculer la distance sémantique entre deux listes, tel que la première liste est :  *$L_1 = S_1\_Input$*  et  *$L_2 = S_2\_Input$*

$$Distance\_input = Dist\_L1\_L2 (S_1\_Input, S_2\_Input)$$

De la même façon nous utilisons  *$Dist\_L1\_L2 (L_1, L_2)$*  pour calculer la distance sémantique entre deux liste, tel que la première liste est :  *$L_1 = S_1\_Output$*  et  *$L_2 = S_2\_Output$*

$$Distance\_output = Dist\_L1\_L2 (S1\_Output, S2\_Output)$$

Les valeurs *Distance\_input* et *Distance\_output* (les distances sémantique des entrées et les distances sémantiques des sorties) seront utilisés pour obtenir la mesure finale de la relation sémantique entre le service  $S_1$  et le service  $S_2$ .

Nous calculons la distance finale par la formule suivante :

$$dist\_S1\_S2(S_i,S_j) = \frac{Distance\_Input + Distance\_Output}{2}$$

Nous choisissons une valeur pour le paramètre « *seuillage* » des distances sémantique, par exemple on prend en considération que les distances supérieures à un *seuil* = 0.5

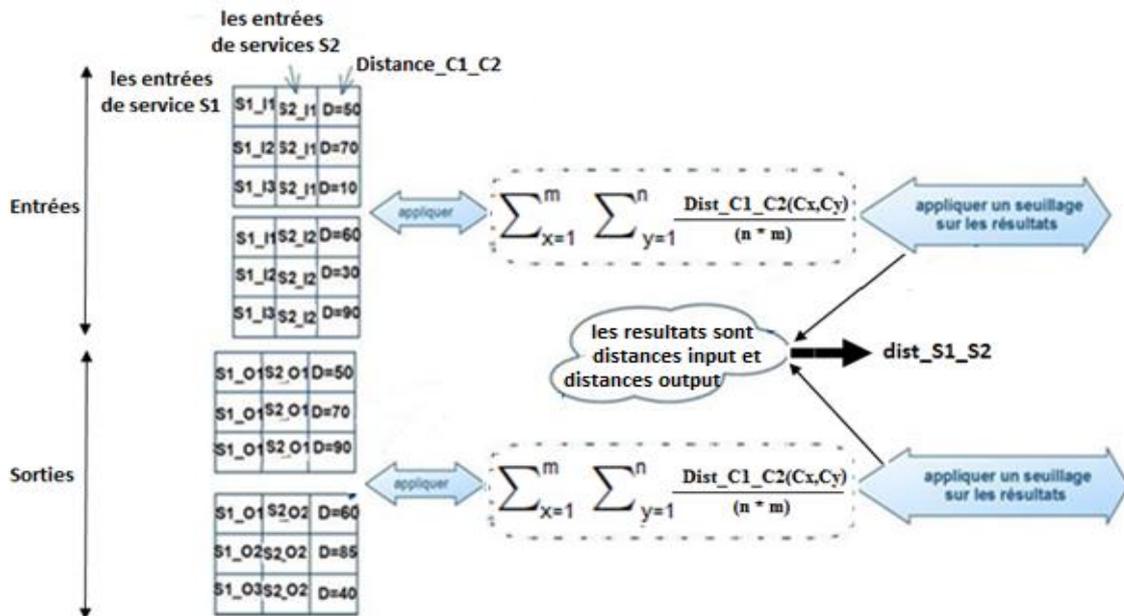


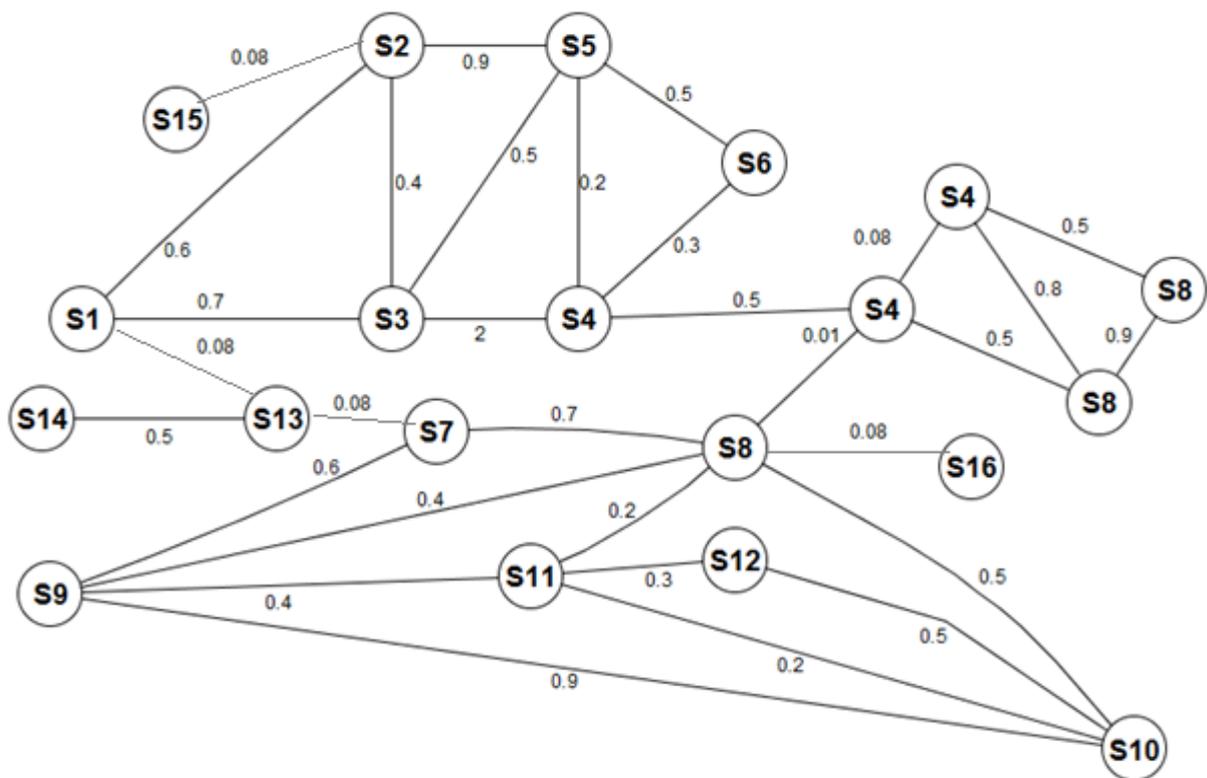
Figure3.7 : Exemple d'une Correspondance entre deux services web

**3. La construction de graphe :**

Un graphe des services web est un graphe valué qui rassemble tous les services disponibles dans lequel nous respectons les présentes règles :

- Chaque service web est un nœud,
- Une arrête relie deux services,
- Le poids de l'arrête représente la distance sémantique entre les deux services reliés,

La figure suivante schématise un exemple de ce graphe :



*Figure3.8 : Exemple d'un graphe des services web*

Dans ce qui suit, nous utilisons l'algorithme *TKC* (*Top-Construction k*) [33] pour construire notre graphe :

<b>Algorithme</b> : Construction du graphe des services web.
<b>Entrés</b> : les fichiers WSDL.
<b>Sorties</b> : un graphe de services web.

```

1.   E = null;
2.   For  $\forall S_i \in S$  Do
    a.   Similarlist = null ;
    b.   For  $\forall S_i \in S$  Do
        If  $S_i \neq S_j$  Then
            get dist_S1_S2( $S_i, S_j$ ) and Similarlist=( $S_j, \text{dist\_S1\_S2}(S_i, S_j)$ );
    c.   Sort Similarlist descendant by similarity;
    d.   get top k items from Similarlist to Similarlist0;
    e.   For all  $S_k \in \text{Similarlist0}$  Do
        If  $(S_i, S_k) \notin E$  Then add  $E=(S_i, S_k)$ ;
3.   Return WSSN;

```

**Explication de l’algorithme :**

Pour un *service web*  $S_i$ , si le *service web*  $S_j$  est l’un des **Top-k** de la liste des services web semblables à  $S_i$ , donc un arc entre  $S_i$  et  $S_j$  est créé dans le graphe des services.

L’algorithme **TKC (Top-Construction k)** a comme paramètre «  $k$  » qui représente le nombre des voisins dans la liste.

On note le graphe de services :  $WSSN=\{ N, E\}$ ,

- où  $N = \{ S_1, S_2, \dots \}$  est l’ensemble des services web est :
- et  $E = \{ (S_i, S_j), \dots \}$  est l’ensemble des arcs, tel que  $S_i, S_j \in N$

### ***Partie 3 : la classification des services***

Dans cette section, nous utilisons l'algorithme **TopLPA** (*Top Label Propagation Algorithm*) [34] pour classer les services web en groupes de services où chaque groupe contient l'ensemble des services du même domaine.

L'algorithme ci-dessous prend en entrée le graphe de services web résultat de la partie précédente, et donne comme sortie un ensemble de classes :

<b>Algorithme</b> : classification des services web
<b>Entrées</b> : graphe des services
<b>Sorties</b> : l'ensemble des classes
<b>Paramètres</b> : $k_c$ : nombre de classe et $\beta$ : le seuil
<pre> 1. <math>\forall S_i \in S</math> <b>Do</b>     a. randomly choose a class from <math>[1, k_c]</math> and assign it to <math>S_i</math> 2. nodechanged = N; 3. <b>While</b> nodechanged &gt; <math>(1-\beta)*N</math> <b>Do</b>     a. nodechanged = 0;     b. <b>For</b> <math>\forall S_i \in S</math> <b>Do</b>         Select <math>S_i</math> neighbors from WSSN <math>\rightarrow</math> neiList         Sort neiList descendant by similarity ;         Choose the first half neighbor in neiList <math>\rightarrow</math>         neiList0;         Choose the most frequent class c from neiList0;         <b>If</b> class of <math>S_i \neq c</math> <b>Then</b>             Assign c to <math>S_i</math> and nodechanged ++; </pre>

*Explication de l'algorithme :*

Tout d'abord, nous devons attribuer au hasard une classe à chaque service Web. Puis, nous devons changer la classe de chaque service Web en prenant en considération les voisins les plus similaires.

Tant que les nœuds qui ne changent pas leur classe dépassent le seuil  $\beta$ , l'algorithme est en cours d'exécution.

Cet algorithme utilise les paramètres:  $k_c$  nombre de classe et le seuil  $\beta$ .

### **Partie 4 : la découverte des services**

Dans cette partie de notre système nous allons expliquer l’algorithme de découverte, la partie responsable de chercher les services satisfaisants la requête du client.

L’algorithme de découverte prend en entrée la requête de l’utilisateur composée d’un ensemble de concepts en entrée et un ensemble de concepts en sortie, ainsi que l’ensemble des classes disponibles dans notre système (les classes résultats de la partie précédente).

La structure de la requête est :

- **R\_Input** : La liste des concepts en entrés de la requête.
- **R\_Output** : La liste des concepts en sortie de la requête.

La structure de chaque service dans les différentes classes est :

- **S\_Input** : La liste des concepts en entrée du service.
- **S\_Output** : La liste des concepts en sortie du service.

Pour notre système, nous proposons l’algorithme de découverte suivant :

<b>Algorithme</b> : découverte des services web.
<b>Entrés</b> : une requête de l'utilisateur.
<b>Sorties</b> : un ensemble des services web.
 <b>1-</b> Déterminer la classe <b>C</b> appropriée. <b>a-</b> Chercher 3 services de chaque classe. <b>b-</b> Calculer la distance entre la requête et les 3 services. <b>c-</b> Calculer la moyenne de la requête avec les 3 services. <b>d-</b> Ordonnancement décroissant du résultat dans une liste. <b>2-</b> Chercher les services dans la classe <b>C</b> . <b>3-</b> Afficher les résultats pertinents.

La première étape de l’algorithme est de trouver dans quelle classe il faut faire notre recherche parmi les différentes classes en entrées de l’algorithme (dans notre système il existe 7 classes).

Nous allons commencer par choisir 3 services au hasard de chaque classe en calculant la distance sémantique entre la requête du client et chacun de ces services.

La figure 3.8 résume les différentes étapes de calcul de la distance sémantique entre la requête client et un service:

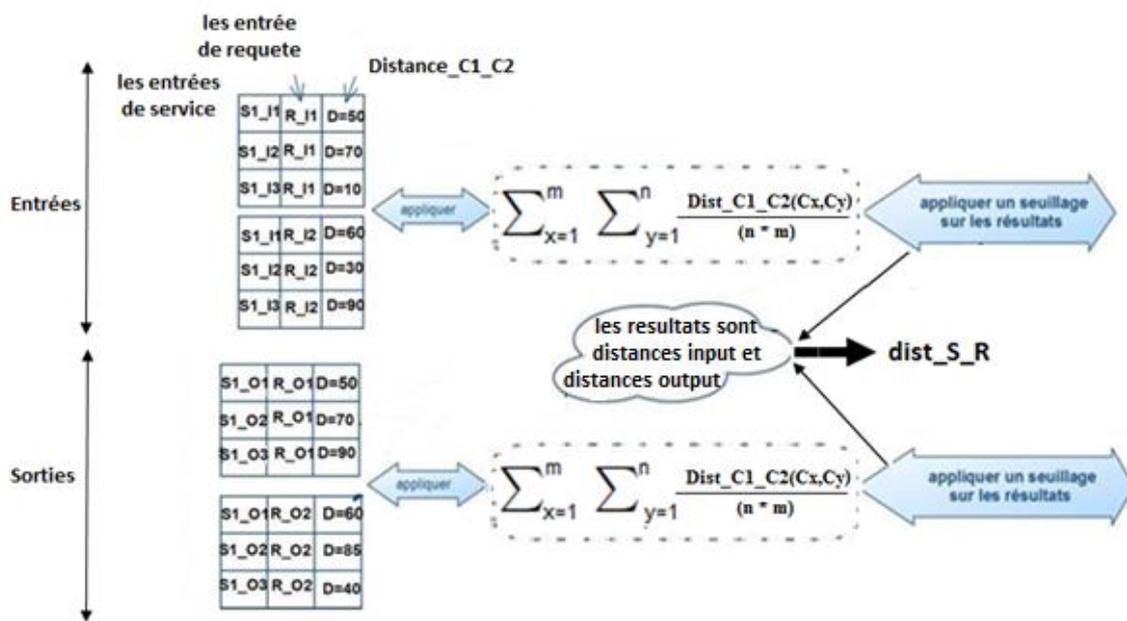


Figure3.9 : Exemple d’une Correspondance entre une requête et un service

La fonction *Dist\_S1\_S2* permet aussi de calculer la distance sémantique entre une requête et un service web. Puisque la requête a la même structure qu’un service (un ensemble de concepts en entrée et un ensemble de concepts en sortie)

Nous utilisons aussi, la fonction *Dist\_L1\_L2* ( $L_1, L_2$ ) pour calculer la distance sémantique entre deux listes,

$$Distance\_input = Dist\_L1\_L2 (R\_Input, S\_Input)$$

$$Distance\_output = Dist\_L1\_L2 (R\_Output, S\_Output)$$

Les valeurs *Distance\_input et Distance\_output* (les distances sémantique des entrées et les distances sémantiques des sorties) seront utilisés pour obtenir la mesure finale de la relation sémantique entre le requête *R* et le service *S*.

Nous utilisons aussi le paramètre « *seuillage* » des distances sémantique.

Puisque nous calculons la distance avec 3 services de chaque classe, et puisque il existe 7 classes, le résultat de cette étape est 7 tableaux, chaque tableau contient 3 valeurs.

Nous calculons la moyenne de chaque tableau (la moyenne des trois distances d'un tableau), ensuite, nous allons faire un ordonnancement décroissant des résultats obtenus dans une liste *L*, le résultat de cette étape est une liste qui contient 7 valeurs.

Dans l'étape suivante nous allons chercher le service web dans la classe au top de la liste c'est-à-dire qui a la plus grande moyenne (cette distance représente la première valeur de la liste *L*)

Si cette classe ne contient aucun service, nous allons refaire notre recherche dans la classe suivante c'est-à-dire qui a la deuxième plus grande valeur dans la liste *L*, et nous allons continuer par le même principe jusqu'à la fin.

Enfin, Nous affichons les résultats pertinents qui représentent l'ensemble des services satisfaisants à la requête client.

### ***Conclusion***

Dans ce chapitre, nous avons présenté d'abord une méthode pour extraire les caractéristiques d'un service web à partir d'un document *WSDL*.

Puis, nous avons construit le graphe des services web en utilisant l'algorithme *TKC* qui se base sur le calcul de la distance sémantique entre les services Web.

Ensuite, l'algorithme de classification prend en entrée ce graphe pour classer les *services web* en groupes.

---

*Chapitre 4*  
*Implémentation*

---

***Introduction :***

Nous avons vu dans le chapitre précédent l'architecture de notre système ainsi que la description et le rôle des modules qui le constituent.

Ce chapitre est organisé de la façon suivante : nous commençons tout d'abord par une présentation des environnements utilisés pour le développement de notre système, à savoir : le langage de programmation, les outils utilisés pour l'implémentation, et les bibliothèques utilisées. Et par la suite, nous allons présenter la description et les fonctionnalités de l'application en décrivant sa structure.

## ***1. Environnements utilisés pour le développement :***

Dans cette section nous présentons les principes d'implémentation de notre application, pour cela nous définissons les différents outils de programmation que nous avons utilisés, afin d'atteindre le but de la recherche sémantique des services web. L'environnement de développement est constitué de :

- ✓ Le langage de programmation java.
- ✓ L'environnement de programmation NetBeans.
- ✓ La base de données :
  - Les services web.

### ***1.1. Langage de programmation :***

Java est un langage orienté objet, multi-plate-forme, conçu pour être facilement utilisable et maniable par une large majorité de développeurs. Développé par *Sun microsystems*. Java rassemble aujourd'hui derrière lui une large communauté d'acteurs informatiques majeurs tels que HP, IBM, Oracle, Borland.

Quand on aborde ce langage, on met souvent en exergue : « Java est un langage simple, orienté objet, réparti, interprété, robuste, sécurisé, indépendant de l'architecture, portable, efficace, multithreads et dynamique ».

Java a été choisie pour notre travail pour les raisons suivantes [35] :

- ✓ Java est absolument basée sur la technologie Objet et emprunte de nombreux éléments au C++. En revanche, les concepteurs de Java ont supprimé les concepts les plus difficiles.
- ✓ Java s'affranchit des plates-formes : il fonctionne en mode interprété. Nous pouvons développer le programme sous n'importe quel système d'exploitation : Unix, Macintosh, Dos, Windows, DOS, etc. Cet avantage nous permet de réutiliser au maximum toutes les fonctions ou classes en diminuant la durée de développement.

### ***1.2. Environnement de programmation « NetBeans » :***

NetBeans est un environnement de développement intégré (IDE) pour Java, placé en open source par Sun en juin 2000 sous licence CDDL (Common Development and Distribution License). En plus de Java, NetBeans permet également de supporter différents autres langages, comme Python, C, C++, XML et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages web). NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X et Open VMS [36].

### ***1.3. Les services web (WSDL) :***

C'est un ensemble de documents WSDL où chaque document est un service web. Ces services sont utilisés pour le test du bon déroulement de notre système.

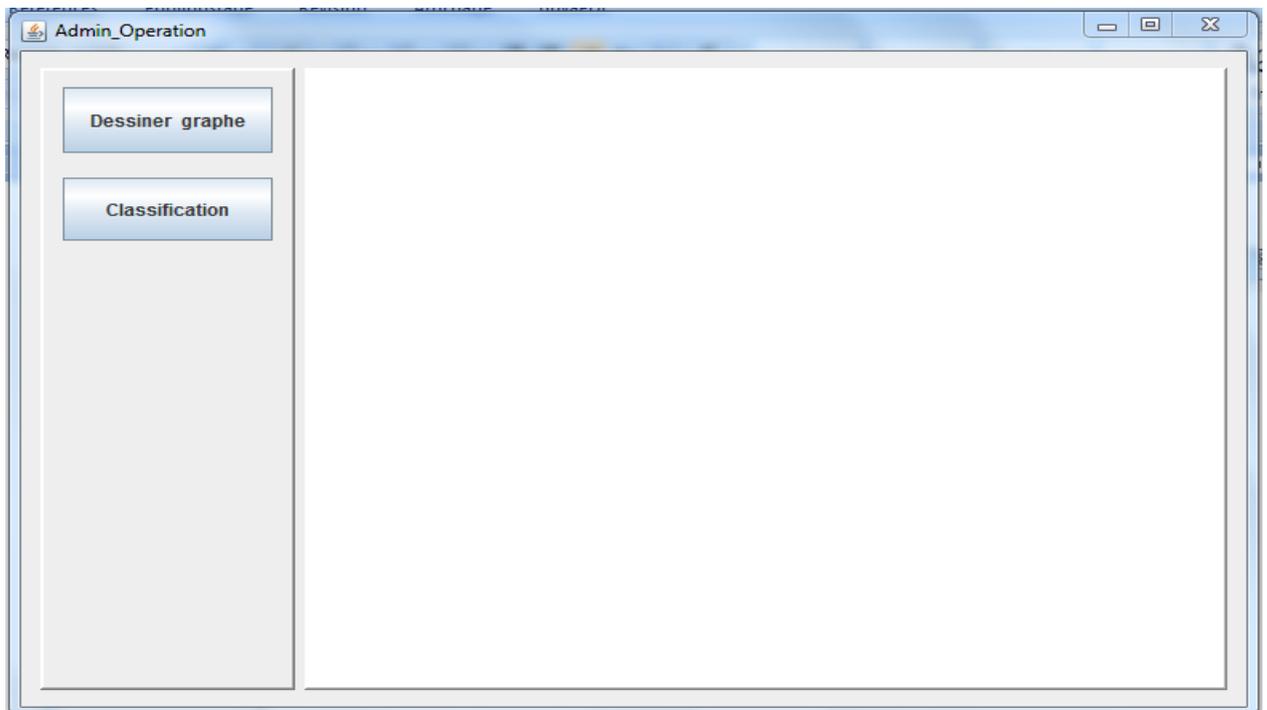
## **2. Implémentation :**

Le système que nous avons développé est un système a deux sous-systèmes principales : un système de classification de services web basé sur un *algorithme de classification* qui prend en entrée un graphe de services et donne comme résultats un ensemble de classes des *services web*. Et le deuxième sous-système est la recherche sémantique des services web basé sur un algorithme de découverte appliqué à l'ensemble des classes obtenus.

Dans ce qui suit nous présentons et nous décrivons les fonctionnalités offertes par l'application.

### **2.1. Présentation de l'application classification :**

Au lancement de l'application classification, l'interface de l'administrateur s'affiche, il s'agit d'une interface simple d'utilisation pour permettre à l'administrateur d'effectuer la classification des services web par groupes. La figure 4.1 présente la fenêtre de l'administrateur :



**Figure 4. 1 : l'interface administrateur**

## Chapitre:04.....Implémentation

L'interface utilisateur se compose de plusieurs zones :

Dans *la zone graphe* (au niveau de l'interface Dessiner graphe) : permet d'afficher le graphe de *services web* après le calcul de la distance sémantique.

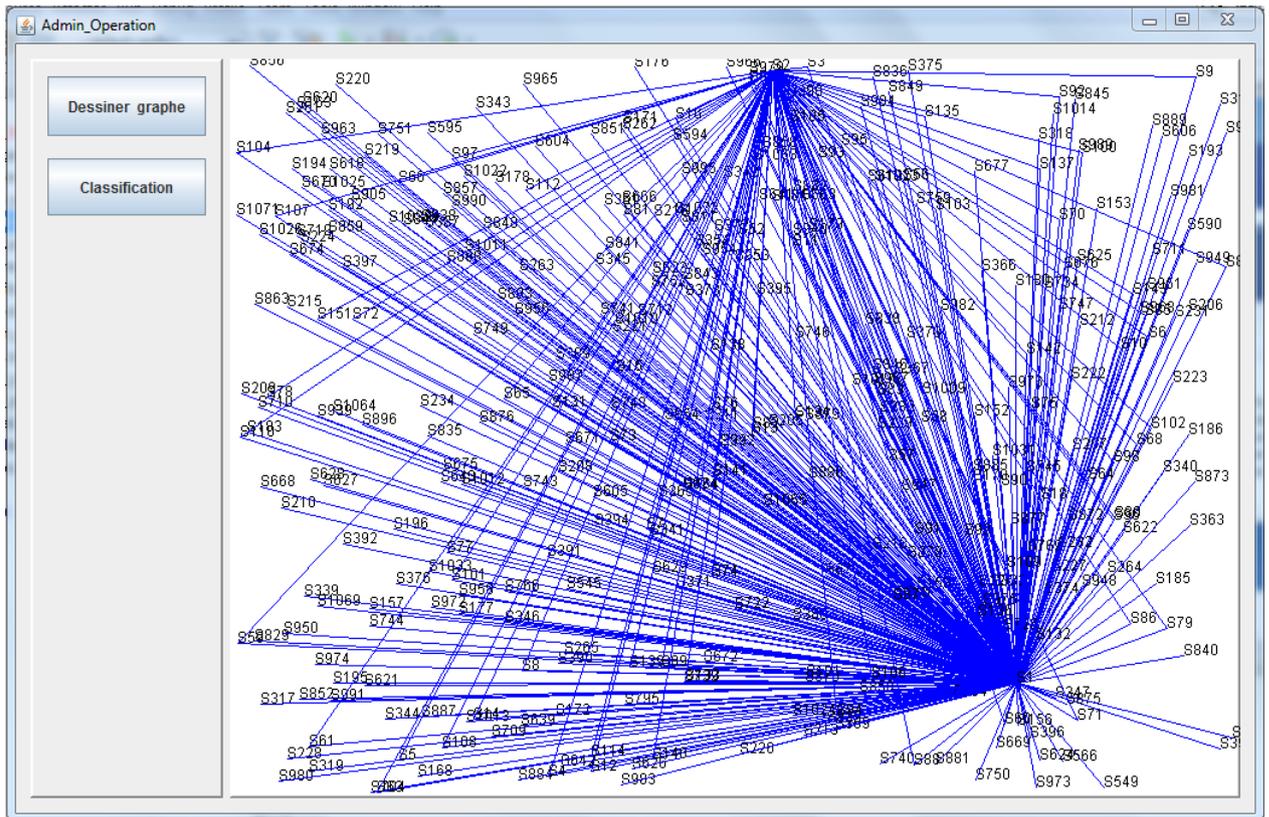
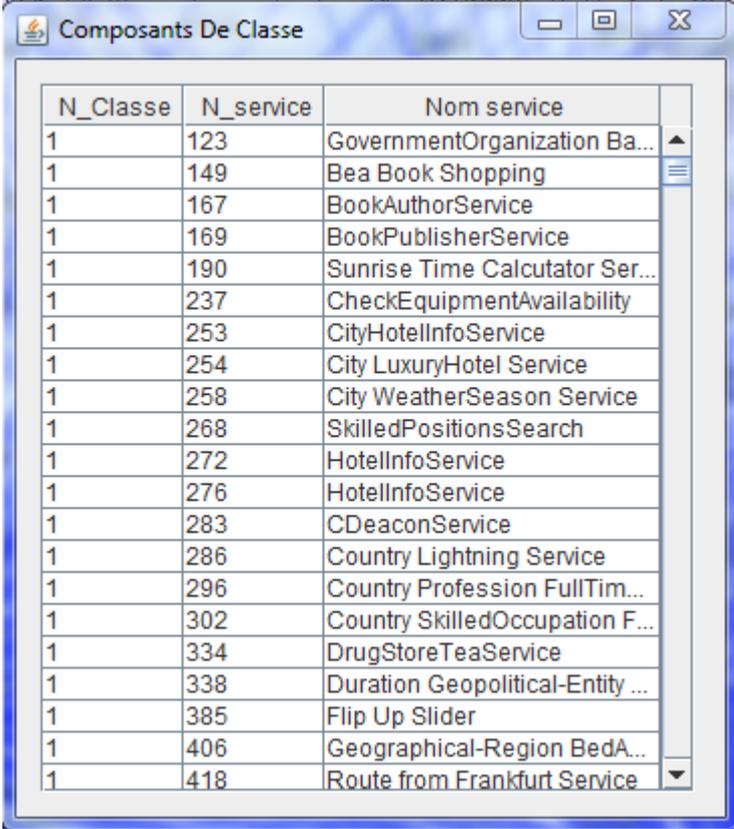


Figure 4. 2 : l'interface graphe

La zone classification permet d'afficher les *services web* selon leur domaine.



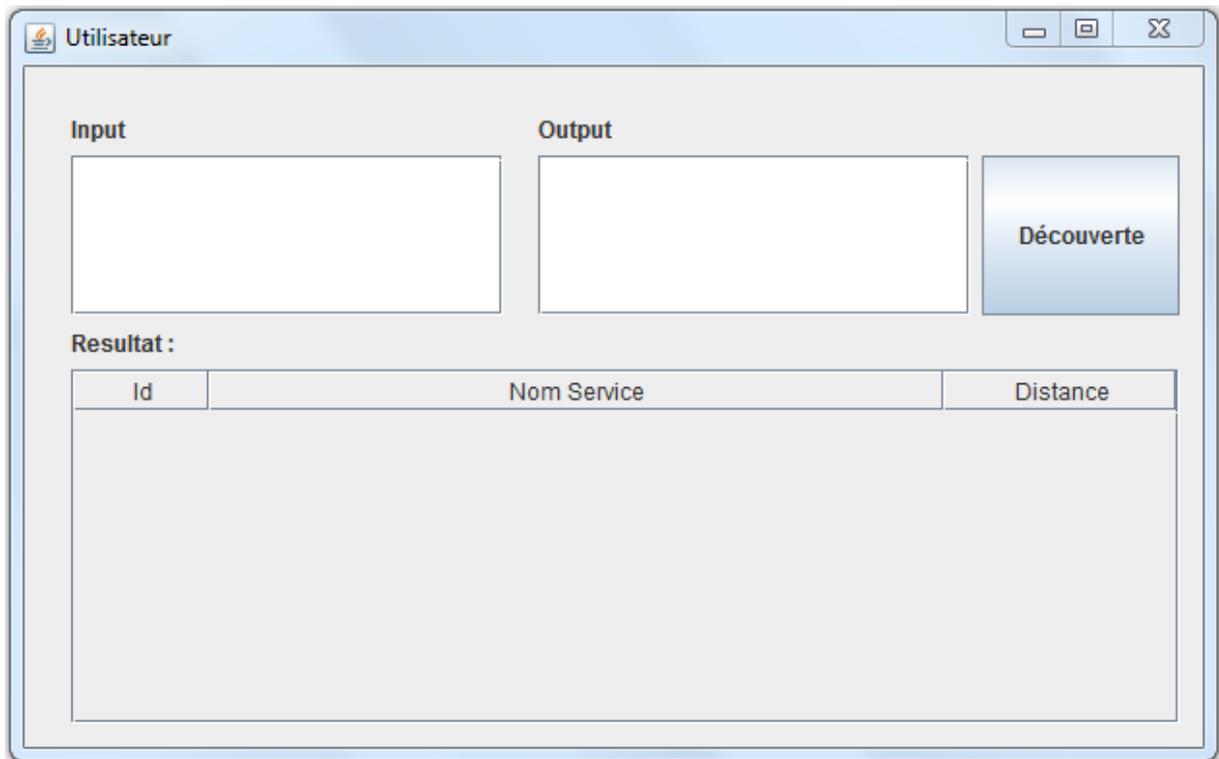
The screenshot shows a window titled "Composants De Classe" with a table containing the following data:

N_Classe	N_service	Nom service
1	123	GovernmentOrganization Ba...
1	149	Bea Book Shopping
1	167	BookAuthorService
1	169	BookPublisherService
1	190	Sunrise Time Calcutator Ser...
1	237	CheckEquipmentAvailability
1	253	CityHotellInfoService
1	254	City LuxuryHotel Service
1	258	City WeatherSeason Service
1	268	SkilledPositionsSearch
1	272	HotellInfoService
1	276	HotellInfoService
1	283	CDeaconService
1	286	Country Lightning Service
1	296	Country Profession FullTim...
1	302	Country SkilledOccupation F...
1	334	DrugStoreTeaService
1	338	Duration Geopolitical-Entity ...
1	385	Flip Up Slider
1	406	Geographical-Region BedA...
1	418	Route from Frankfurt Service

*Figure 4. 3 : l'interface classification*

## ***2.2. Présentation de l'application découverte :***

Au lancement de l'application découverte, l'interface de l'utilisateur s'affiche, il s'agit d'une interface simple d'utilisation pour permettre à l'utilisateur d'effectuer les découvertes sémantique. La figure 4.1 présente la fenêtre de l'utilisateur :



*Figure 4. 4 : l'interface utilisateur*

L'interface utilisateur se compose de plusieurs zones :

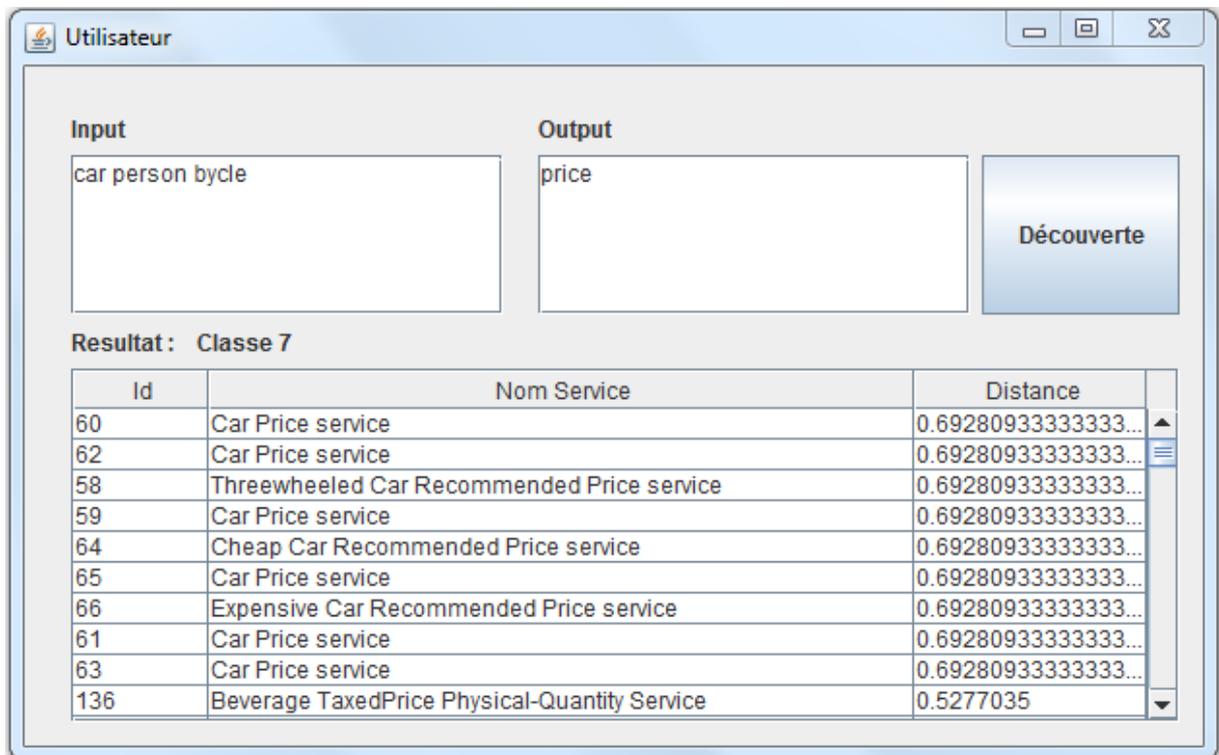
**Zone de recherche** : elle permet à l'utilisateur de rechercher une requête selon la classe appropriée.

**Zone de résultats** : c'est le volet dans lequel le système affiche les résultats rendus par l'algorithme « découverte ».

**Zone Input** : cette zone contient la liste des concepts en entrée de la requête.

**Zone Output** : cette zone contient la liste des concepts en sortie de la requête.

*Exemple de découverte :*



*Figure 4. 5 : exemple de découverte*

***Conclusion :***

Nous avons présenté dans ce chapitre une implémentation de notre application nommée « classification ». Le travail réalisé est un système de classification des services web à base de graphe. Pour construire ce graphe nous avons calculé la distance sémantique entre les différents services.

L'implémentation a été réalisée en utilisant le langage de programmation Java et l'environnement développement NetBeans.

# *Conclusion générale*

L'utilisation des services web est augmentée de manière explosive au cours des dernières années, et les utilisateurs doivent définir leurs exigences fonctionnelles et non fonctionnelles de façon claire pour obtenir des *services web* appropriés en utilisant les mots-clés qui décrivent l'objectif de la requête.

La classification des services web est le processus qui permet d'organiser un ensemble de *services web* selon des critères de similitude catégorielle et fonctionnelle. Elle peut permettre aussi de faciliter, d'optimiser, d'automatiser l'efficacité et l'efficacité des processus de découverte, de composition, d'exécution et de la gestion des *services web*.

Dans ce travail, nous avons présenté une conception et une réalisation d'un système composé d'une classification des services web qui prend en compte la sémantique et un processus de découverte basé sur cette classification.

Au début, nous avons construit un graphe qui rassemble tous les services web en utilisant les caractéristiques extraites à partir des documents *WSDL* et le calcul de la distance sémantique pour relier deux services dans le graphe.

Ensuite, nous avons présenté une méthode de regroupement des services web basée sur le graphe de services web sémantique, le résultat de cette étape est un ensemble de classes de services web.

L'ensemble des classes est utilisé au niveau de la découverte dans le but de minimiser le champ de la recherche et viser seulement la classe concernée et non pas la totalité des services.

# *Références bibliographiques*

- [1] : Chafik Berdjouh vers Une approche basée agent pour la découverte de services Web, université kasdi merbah ouargla, 2009.
- [2]: Belkhiter Khalid vers Une approche Sélection des services Web à base d'algorithmes génétiques multi-objectifs, Université Abou Bakr Belkaid– Tlemcen, 2011/2012(Ethan Cerami, Web Services Essentials, édition O'Reilly, Février 2002)
- [3] : Issam RABHI vers Une approche Testabilité des services Web, Université Blaise Pascal - Clermont-Ferrand II, 2012. Français. [09 janvier 2012].
- [4] : Sofiane Chemaia vers une approche de composition de services web a l'aide des réseaux de petri orientés objet, université Abdelhamid Mehri Constantine 2, 2014.
- [5] : [xml.chez.com/initiation/](http://xml.chez.com/initiation/).
- [6] : [miageprojet2.unice.fr/@api/deki/files/2111/=soap.pdf](http://miageprojet2.unice.fr/@api/deki/files/2111/=soap.pdf).
- [7] : <https://openclassrooms.com/courses/les-services-web>.
- [8] : [https://project.inria.fr/Services\\_Web/files/2012/01/WSDL.pdf](https://project.inria.fr/Services_Web/files/2012/01/WSDL.pdf).
- [9] : boudjellaba hakim vers une approche de Sélection des Web Services Sémantiques, Université A/Mira de Bejaia, 2012.
- [10] : BOUDJELLAL Aïcha et BESSADAT Nor El Houda vers une approche de Sélection de Service Web à base d'Algorithme Culturel Université Abou Bakr Belkaid – Tlemcen, 2012/2013.
- [11] : Mohamed Gharzouli vers une approche de Composition des Web Services Sémantiques dans les systèmes Peer-to-Peer ,25/09/2011.
- [12] : Hamida Souraya vers une approche basée agent mobile pour le m-service web sémantique, Université Mohamed Khider – BISKRA, juin 2014.
- [13]: R. Akkiraju,J. Farrell, J.Miller, M. Nagarajan, M. Schmidt, A. Sheth et K. Verma. «Web service semantics: Wsdl-s». Technical report, IBM, 18 avril 2005.

- [14] : David Martin, Mark Burstein, Jerry Hobbs, Ora Lassila, Drew McDermott, Sheila McIlraith, Sridhar Narayanan, Massimo Paolucci, Bijan Parsia, Terry Payne, Evren Sirin, Naveen Srinivasan et Katia Sycara. «Owl-s: Semantic markup for web services». Technical report, W3C. 2004.
- [15]: Enrico Motta, John Domingue, Liliana Cabral, et Mauro Gaspari. «Irs-II: A framework and infrastructure for semantic web services». International Semantic Web Conference, pages 306–318. 2003.
- [16]: D. Fensel et C. Bussler. «The web service modeling framework wsmf». *Electronic Commerce: Research and Applications*, 1(2) : 113–137. 2002.
- [17] : Daoud Zahèra et Maalache Nassima vers une approche de Composition des Services Web Sémantiques à base d'Algorithmes Génétiques, Université Abou Bakr Belkaid–Tlemcen, 2012/2013.
- [18] : Mamoun Mohamad Jamous et Safaai Bin Deris vers une approche de Web Services Non-Functional Classification to Enhance Discovery Speed, Université Teknologi Malaysia, 2011.
- [19]: Hongbing, W., et al. Web Service Classification Using Support Vector Machine. In *Tools with Artificial Intelligence (ICTAI)*, 2010 22nd IEEE International conference on. 2010.
- [20]: Crasso, M., A. Zunino, and M. Campo, Combining Document Classification and Ontology Alignment for Semantically Enriching Web Services. *New Generation Computing*, 2010. 28(4): p. 371-403.
- [21]: Kehagias, D.D., et al., A WSDL Structure Based Approach for Semantic Categorization of Web Service Elements, in *Artificial Intelligence: Theories, Models and Applications, Proceedings*, S. Konstantopoulos, et al., Editors. 2010, Springer-Verlag Berlin: Berlin. p. 333-338.
- [22]: Miller, G.A., WordNet: a lexical database for English. *Commun. ACM*, 1995. 38(11): p. 39-41.
- [23]: Porter, M. Porter Stemming Algorithm. 2006 2006 [cited 2010; Available from: <http://tartarus.org/~martin/PorterStemmer/>].

- [24]: AbuJarour, M., F. Naumann, and M. Craculeac, Collecting, annotating, and classifying public web services, in Proceedings of the 2010 international conference on On the move to meaningful internet systems - Volume Part I. 2010, Springer-Verlag: Hersonissos, Crete, Greece. p. 256-272.
- [25]: Charikar, M.S., Similarity estimation techniques from rounding algorithms, in Proceedings of the thirty-fourth annual ACM symposium on Theory of computing. 2002, ACM: Montreal, Quebec, Canada. p. 380-388.
- [26]: Lee, Y., Quality-context based SOA registry classification for quality of services, in Proceedings of the 11th international conference on Advanced Communication Technology - Volume 3. 2009, IEEE Press: Gangwon-Do, South Korea. p. 2251-2255.
- [27]: Feng Deng vers Une approche Web Service Matching based on Semantic Classification, School of Health and Society Department of Computer Science. Spring 2012.
- [28]: Chantal Cherifi vers Une approche Classification et Composition de Services Web : Une Perspective Réseaux Complexes université de corse-pascal paoli 2011.
- [29]: bouyacoub soumia et kaouadji Sarra vers une approche d'enrichissement de la représentation conceptuelle dans la catégorisation du texte en utilisant les mesures de similarité sémantique Université Abou Bakr Belkaid – Tlemcen, 2012/2013.
- [30]: <http://tartarus.org/~martin/PorterStemmer/>.
- [31]: Elias Iosif, Alexandros Potamianos, SemSim: Resources for Normalized Semantic Similarity Computation Using Lexical Networks.
- [32]: Eyhab Al-Masri and Qusay H. Mahmoud, vers Une approche “Investigating web services on the world wide web,” in Proc. 17th International Conference on the World Wide Web, ACM Press, pp. 795–804, April 2008.
- [33]: Junjie tong, Haihong E, Junde song, Meina song vers Une approche Clustering Web Services via Constructing Web Services Similarity Network\*, PCN & CAD Center Lab, Beijing University of Posts and Telecommunications, Beijing 100876, China.
- [34]: A. Clauset, M. E. J. Newman, Cristopher Moore. vers Une approche Finding Community Structure in Very Large Networks. Phys. Rev. E, 2004, 70, pp. 1-6.

[35] : Chafik Berdjouh vers Une approche basée agent pour la découverte de services Web, université kasdi merbah ouargla, 2009.

[36] : <https://www.projet-plume.org/fiche/netbeans>.