

Université Abou Bekr Belkaid

Tlemcen Algérie



جامعة أبي بكر بلقايد

تلمسان الجزائر

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
Ministère de l'enseignement supérieur et de la recherche scientifique  
Université ABOU BEKR BELKAID

Faculté de Sciences  
d'Informatique



Département

MEMOIRE

Projet de Fin d'Etude

Pour l'obtention du diplôme de Master en Informatique

*Thème*

**LA DECOUVERTE ENSEMBLISTE DES SERVICES  
WEB À BASE DE DOMINANCE FLOUE**

Présenté par

*M<sup>r</sup> TABET AOUL Walid Houcine & MEZIANE Rachad*

Soutenu le 21/06/2015 devant le jury composée de MM :

M<sup>R</sup> .....

Examineur

M<sup>R</sup> .....

Examineur

M<sup>R</sup> .....

Examineur

M<sup>R</sup> Hadjilla Fethallah .....

Encadreur

*Promotion 2014 - 2015*

# *Résumé*

Au cours de ces dernières années les entreprises ont vu un intérêt croissant dans l'utilisation des services Web comme un moyen fiable pour la publication de données et le partage de ressources. L'objectif de travail est d'étudier la problématique de découverte de Services Web, pour cela nous avons proposé une approche hybride qui se base sur la fusion de 05 algorithmes de matching à travers la logique floue.

Mots clés : SOA, découverte de Service, Algorithmes de matching, Logique Floue

# *Abstract*

In recent years companies have seen a growing interest in the use of Web services as a reliable means for publishing and sharing data. In this work we address the Web Service discovery problem, more specifically we propose a hybrid Solution that fuses five matching algorithms by using a Fuzzy dominance relationship.

Keywords :Web Service discovery , Hybrid ,matching algorithm, fuzzy dominance

# ملخص

شهدت الشركات في السنوات الاخيرة اهتماما متزايدا في استخدام خدمات الويب باعتبارها وسيلة يمكن الاعتماد عليها لنشر البيانات و مشاركتها ، الهدف هو دراسة إشكالية اكتشاف خدمات ويب ، لهذا اقترحنا المنهجية الهجينة التي تقوم على دمج 05 خوارزميات المطابقة من خلال المنطق الضبابي.

# *Remerciements*

*Avec l'aide du Dieu ALLAH le tout puissant et grand  
qui nous a donné la force et la vie pour pouvoir achever  
ce travail.*

*Nous tenons à remercier tout particulièrement  
monsieur F.Hadjila enseignant docteur en informatique  
à l'université Abou Bekr Belkaid notre encadreur de  
mémoire pour son aide, son soutien, ses conseils, sa  
patience, sa générosité, Son ouverture d'esprit, sa  
disponibilité et son temps consacré, afin de rendre cette  
étude agréable et intéressante.*

*Et nous tenant à remercier chaleureusement nos  
examineurs d'avoir accepté de participer au jury de  
ce mémoire.*

*Nous adressons nos sincères remerciements a nos  
merveilleuses familles de nous avoir données le courage  
d'accomplir cette modeste recherche et aussi à tous ceux  
qui ont contribué à sa réalisation.*

*Walid & Rachad*

# *Dédicaces*

*Je dédie ce modeste travail et ma profonde gratitude à celle qui m'a transmis la vie, l'amour, le courage, à toi chère maman Amina toutes mes joies, et ma reconnaissance, à mon très cher père Ghouti à qui m'adresse au ciel les vœux les plus ardents pour la conservation de sa santé et de sa vie, pour l'éducation qu'ils m'ont prodigué ; avec tous les moyens et au prix de toutes les sacrifices au sens du devoir qu'ils m'ont enseigné depuis mon enfance, et bien sur à mon cher frère Yacine et mes chères proches.*

*Mes camarades de la promo de MID*

*De 2014/2015.*

*Walid*

# *Dédicaces*

*A ma chère famille :*

*Mon père, ma Mère, Boumedienne, Malîha, Werda et*

*Fayçal.*

*Et mes amis que j'estime beaucoup.*

*Je vous aime !*

*Rachad*

---

## LISTE DES FIGURES

<b>Figure I.1</b>	Structure d'un message SOAP .....	8
<b>Figure I.2</b>	Structure de l'interface de description des services .....	11
<b>Figure I.3</b>	Recherche des services par l'annuaire UDDI .....	14
<b>Figure I.4</b>	Structure de l'Annuaire UDDI .....	15
<b>Figure I.5</b>	Les principaux composants descriptifs d'enregistrement .....	16
<b>Figure I.6</b>	L'architecture et les standards des services WEB .....	17
<b>Figure II.1</b>	Organigramme de Fuzzy Dominated Algorithm .....	27
<b>Figure II.2</b>	Courbe illustrative de la Fonction Dominé Floue .....	29
<b>Figure II.3</b>	Fenêtre principale de l'application .....	30
<b>Figure II.4</b>	Résultats du chargement et d'Indexation des Services Web .....	31
<b>Figure II.5</b>	Résultats de Tri de Rappel et de précision du FDDA .....	32
<b>Figure II.6</b>	L'Histogramme de Rappel selon sept TOP K multiples .....	35
<b>Figure II.7</b>	L'Histogramme de Précision selon sept TOP K multiples .....	35
<b>Figure II.8</b>	L'Histogramme de Rappel des Moyennes des deux approches .....	36
<b>Figure II.9</b>	L'Histogramme de Précision des Moyennes des deux approches .....	36

# TABLE DES MATIERES

Remerciements	
Dédicaces	
Résumé	
Liste des Figures	
Introduction Générale .....	1
<b>Chapitre I : GENERALITES SUR LES SERVICES WEB</b>	
I.1 Introduction.....	4
I.2 L'architecture des Services web « SOA ».....	4
I.2.1. Définition .....	4
I.3 Services Web et Définitions.....	5
I.4 La Famille Des Standards Pour Les Services Web.....	6
I.4.1. SOAP « Simple Object Access Protocol ».....	6
a. Définition .....	6
b. Fonctionnement Et Structure .....	7
c. Avantages ET Inconvénients.....	9
I.4.2. WSDL « Web Services Description Language » .....	10
a. Définition .....	10
b. Fonctionnement ET Structure de WSDL.....	10
c. Les Types de Description .....	11
I.4.3. UDDI « Universal Description Discovery and Integration ».....	13
a. Définition .....	13
b. Intérêts et Fonctionnement d'UDDI.....	13
I.5 Enregistrement et Publication d'un Service Web.....	15
I.6 Recherche et Découverte des Services Web.....	17
I.6.1 Définition.....	17
I.6.2 Etat De L'Art .....	18
a. Les Approches de Découverte des Services Web.....	18
I.7 Conclusion .....	23
<b>Chapitre II : CONCEPTION ET IMPLEMENTATION</b>	
II.1 Introduction.....	25
II.2 La collection de test.....	25
II.3 La Conception .....	26
II.3.1. Organigramme.....	28
II.3.2. Présentation de l'Approche « Aggrégation du Dominance Floue » .....	26
II.4 Prototype.....	29
II.5 Expérimentation .....	32
II.5.1. Java .....	32

II.5.2.	Netbeans .....	32
a.	Définition .....	32
b.	Version Utilisée.....	32
II.5.3.	Microsoft Excel .....	32
II.5.4.	Machine du Développement .....	33
II.5.5.	Discussion .....	33
II.6	Les Avantages de l'Approche FDDA.....	36
II.7	Conclusion .....	36
	Conclusion Générale.....	38
	Références.....	40



INTRODUCTION  
GENERALE

## Introduction Générale

### I) Contexte

Le travail des entreprises aujourd'hui s'insère dans le cadre de découvrir et améliorer toujours de nouvelles connaissances qui aident à obtenir une satisfaction meilleur du client dans le domaine d'internet et du web. Donc dans notre temps présent les ultimes soucis que rencontrent les développeurs et les entreprises d'internet se présentent dans l'inhérence des changements et de développements connus dans l'environnement, en considérations des besoins existants du client, ainsi le principale travail à faire, pour eux, est d'assurer l'intégration et l'interopérabilité de leurs applications qui sont appelés dans ce cas processus métiers.

L'architecture orientée service « SOA » est née pour répondre aux inconvénients des technologies orientées composants, telles que CORBA [OMG, 2008] (Common Object Request Broker Architecture), Java RMI [Dowding, 1998] (Remote Method Invocation), DCOM (Distributed Component Object Model). [Horstmann et Kirtland, 1997]. Mais ces technologies qui partagent et réutilisent des codes existants ont aussi des inconvénients qui les ont pas laissés passer à l'échelle voulue, à la portée du domaine extérieur.

Ces inconvénients étaient le couplage fort des composants, la complexité d'utilisation et de mise en œuvre de ces middlewares, le non support de certains protocoles du web (http, ftp...), et dans certains cas la non standardisation de ces technologies [Emmerich, 2002].

D'une autre façon le problème existant Avec l'interconnexion des ordinateurs en réseau et en particulier à travers internet, c'est qu'il devient possible de faire fonctionner des applications sur des machines distantes .Pour toutes ces raisons, une interaction entre des programmes distants peut être utile. Les Services web apportent une solution à ce problème en définissant une manière standard d'invoquer une application distante et de récupérer les résultats à travers le web.

Pour répondre aux besoins d'un tiers client. Plusieurs travaux de recherches et applications réelles ont été réalisés et modifiés telles que des algorithmes multiples contenant des formules mais chacun de ces algorithmes ou approches négligés sera probablement soumis à une amélioration au futur.

### I) Objectifs du mémoire

En effet ,nous nous sommes intéressés tout d'abord à éclairer des différents paradigmes qui constituent le cycle de fonctionnement des services web et citer les mécanismes derrière les

trois phases fondamentales qui sont la publication, la découverte et la sélection, l'objectif de cette thèse est d'automatiser ces trois phases grâce à l'approche intelligente ou l'algorithme choisi.

Enfin notre mémoire établit d'autres buts ainsi et qu'une application à présenter et des concepts à illustrer dans les deux chapitres suivants :

### **Chapitre 1 :**

- Montrer le cycle de vie des services web depuis leurs créations jusqu'à son utilisation et leur recherche, et définir tous les standards et les données exogènes.
- Décrire le modèle d'informatique répartie qui représente les technologies des services web permettant l'adaptation dans une image plus grande de l'intégration et du déploiement dans la plateforme.
- Décrire plusieurs approches pour la découverte des services web et critiquer ces derniers.

### **Chapitre 2 :**

On fait l'implémentation de notre application en définissant les matériaux, les plateformes, et les langages utilisés, tout par illustrations à l'aide de captures d'écran.

Le serveur ou le fournisseur fournit ces services sur le web et l'administrateur (nous représentons l'administrateur dans ce cas) à la possibilité de mettre en œuvre un système d'application [programmé en Java, Netbeans] en se basant sur un algorithme, qui procédera par la suite à une analyse, discussion de résultats, et standardisation la fiabilité de la méthode utilisée.

À la fin nous terminons par une conclusion générale



CHAPITRE I :  
GENERALITES SUR LES  
SERVICES WEB

**Chapitre I :****GENERALITES SUR LES SERVICES WEB****I.1 Introduction**

Avec l'interconnexion des ordinateurs en réseau et en particulier à travers l'internet, il devient possible de faire fonctionner des applications sur des machines distantes. L'intérêt d'une application fonctionnant à distance peut à première vue sembler inutile dans la mesure où les applications fonctionnent fort bien en local (sur le poste de l'utilisateur), néanmoins une application distante peut répondre aux problématiques dont Les données peuvent être présentes uniquement sur le serveur distant ,pour ces raisons, une interaction entre des programmes distants peut être utile. Grace à l'effort de plusieurs organisations qui ont partagé un intérêt commun en développant et en maintenant "un marché électronique», les services web sont nés. Ces derniers apportent une solution à ce problème en définissant une manière standard d'invoquer une application distante et d'en récupérer les résultats à travers le web. Mais comment avoir un choix actuel du service le plus approprié classé meilleur parmi le reste et est ce que ce dernier est le même que le choix ultérieur.

Pour cela on va parler dans ce chapitre des services web et toute les technologies reliées avec et aussi sur les algorithmes, nous essayons pour cela de résoudre le problème de découverte et de tri des services web a base de nombreuses méthodes proposées.

**I.2 L'architecture des Services web « SOA »****I.2.1. Définition**

En anglais « Service Oriented Architecture » C'est une forme d'architecture de médiation qui est un modèle d'interaction applicative qui met en œuvre des services (composants logiciels) :

- I. avec une forte cohérence interne (par l'utilisation d'un format d'échange pivot, le plus souvent XML).
- II. des couplages externes « lâches » (par l'utilisation d'une couche d'interface interopérable, le plus souvent un service web WS-\*<sup>1</sup>).

---

<sup>1</sup> Service web ws-\* appelée comme ça puisque une spécification « ws » associée au services web et ces derniers implementent ces spécifications

Donc c'est une architecture reposée sur un ensemble de services simples appelés dans ce cas composant qui fonctionnent d'une manière autonome. [1]

Les SOA sont souvent assimilés à des technologies mais ce sont en réalité des principes d'architectures. En effet, ce concept se repose sur une nouvelle manière d'intégrer et de manipuler les différents composants applicatifs d'un système informatique tel que la comptabilité, gestion de la relation client, production, et autres et en même temps doit gérer les liens qu'ils entretiennent. Les SOA cherchent et tentent toujours à rendre les applications plus souples et réutilisables, en s'appuyant sur des standards, ces architectures peuvent fonctionner dans des environnements hétérogènes. Leur but fondamental est d'améliorer l'interopérabilité entre les multiples systèmes sans être mené à des contraintes et de développer des applications à long terme. SOA nous assure une meilleure conception lors de l'ajout de nouveaux services sans l'endommagement de l'application existante. [Hadjilla et al,2013]

### I.3 Services Web et Définitions

Tout d'abord on peut poser la question : C'est quoi un service web ?

Et en répondant on explique ce que c'est ensuite on donne des définitions exactes et formelles.

Et bien c'est un morceau de logique d'affaires, situé quelque part sur l'Internet, qui est accessible par des protocoles standard-basés d'Internet tels que le HTTP ou SMTP. Utilisant un service web pourrait être aussi simple que notant dans un emplacement ou aussi complexe que facilitant une négociation d'affaires d'multi-organisation.

Donc il existe pas mal de définitions et plusieurs vues de cette notion qui sont présents comme suit :

#### Définition 1

Un service Web est une « unité logique applicative » accessible en utilisant les protocoles standards d'Internet. [3]

#### Définition 2

Les services Web sont des entités logicielles qui ont une interface bien définie et qui effectuent une tâche spécifique.

Un service Web est formellement décrit dans un langage standardisé. [4]

**Définition 3**

Un service web (ou service de la toile<sup>2</sup>) est un programme informatique de la famille des technologies web permettant la communication et l'échange de données entre applications et systèmes hétérogènes dans des environnements distribués. Il s'agit donc d'un ensemble de fonctionnalités exposées sur internet ou sur un intranet, par et pour des applications ou machines, sans intervention humaine, de manière synchrone ou asynchrone. [5]

**Définition 4**

La technologie des services Web est un moyen rapide de distribution de l'information entre clients, fournisseurs, partenaires commerciaux et leurs différentes plates-formes. [6]

**Définition 5**

Selon « **Dico du Net** »<sup>3</sup> c'est une technologie permettant à des applications de dialoguer à distance via Internet indépendamment des plates-formes et des langages sur lesquels elles reposent.

**Définition 6**

Selon « **W3C** »<sup>4</sup> c'est un composant logiciel identifié par une URI, dont les interfaces publiques sont définies et appelées en XML. Sa définition peut être découverte par d'autres systèmes logiciels.

**I.4 La Famille Des Standards Pour Les Services Web****I.4.1. SOAP « Simple Object Access Protocol »****a. Définition**

C'est un protocole d'échange de message indépendant des plateformes, c'est un produit de Microsoft et IBM<sup>5</sup>. Sa première version originale première est Version 1.1 apparait par le W3C (Word Wide Web Consortium) en mai 2000. Selon [Gudin et al, 2003]

<sup>2</sup> La toile : C'est un système hypertexte public fonctionnant sur Internet

<sup>3</sup> Dico du net : est un dictionnaire collaboratif en ligne

<sup>4</sup> W3C « World Wide Web Consortium » : un organisme de normalisation, fondé en octobre 1994

<sup>5</sup> IBM : est une société multi nationale américaine présente dans les domaines du matériel informatique, du logiciel et des services informatiques

C'est un protocole léger pour accomplir la transmission des informations structurées dans un environnement distribué et décentralisé. [W3C]

Par une autre définition c'est un cadre générale permettant l'échange de données structuré en format XML et le transport de ces données basé sur HTTP. [7]

Il se compose de deux parties fondamentales une partie que l'on peut appeler enveloppe XML et une partie de protocole de transport. SOAP est une brique essentiel dans la communication entre les clients et les fournisseurs dans l'architecture SOA de base puisque chaque message formulé est de ce type dont on parle. Sans oublier de citer qu'il est particulièrement utile pour exécuter des dialogues requête-réponse RPC (Remote Procedure Call<sup>6</sup>).

SOAP fait parti de l'idée d'un standard simple et neutre lorsqu'il n'exige pas d'utiliser une API<sup>7</sup> spécifique et n'impose aucun modèle de programmation.

## **b. Fonctionnement Et Structure**

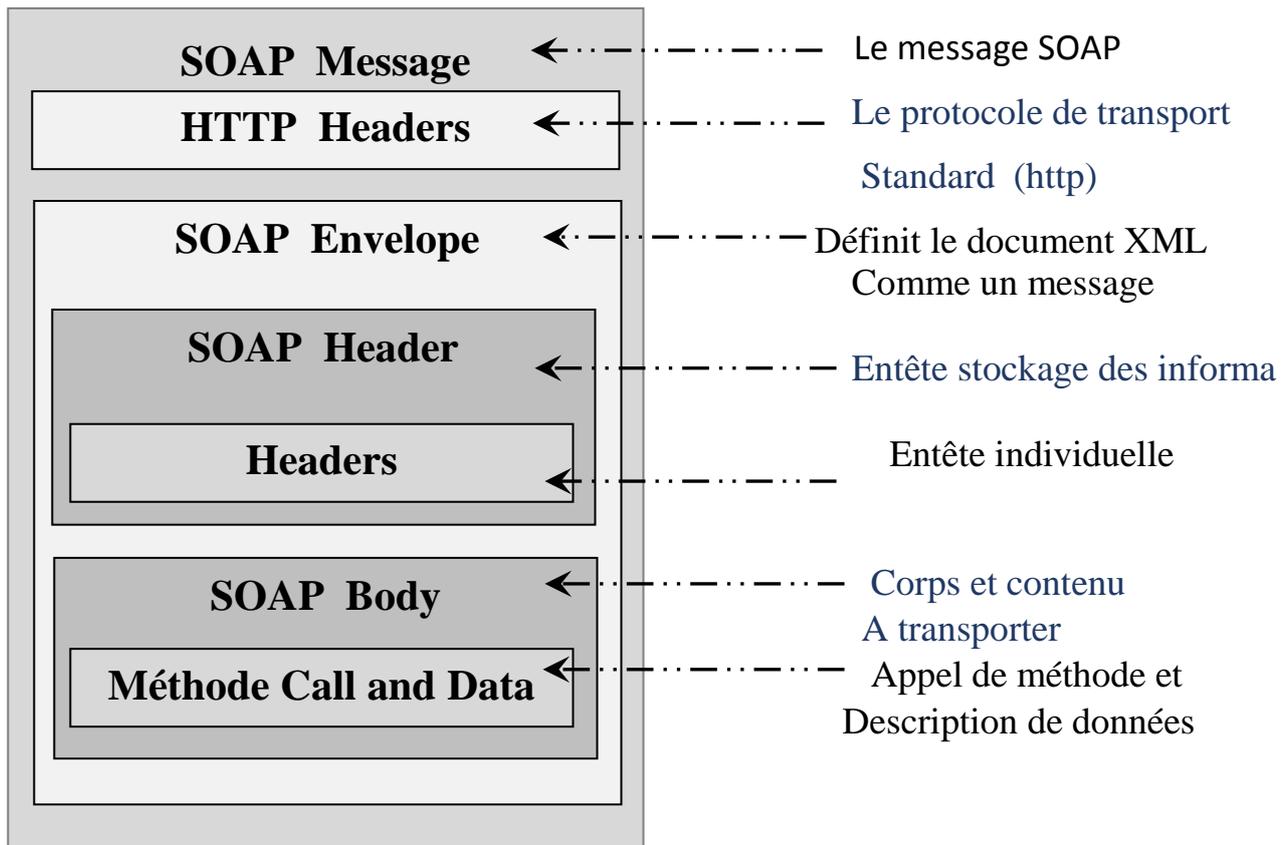
Le message de type SOAP est un document XML qui a une forme qui contient les éléments suivants :

- Une enveloppe qu'elle est a son tour une structure générale représentant le contenu, le destinataire du message, et la nature du message.
- Un ensemble de règles de codage pour l'échange des objets.
- Un fonctionnement en modèle Client/serveur (RPC).

---

<sup>6</sup> Remote Procedure Call : Est un protocole réseau permettant de faire des appels de procédures sur un ordinateur distant à l'aide d'un serveur d'applications

API : Application Programming Interface est un ensemble normalisé de classes, de méthodes ou de fonctions



**Fig I.1** Structure d'un message SOAP

Toutes ces parties constituantes du message SOAP ont un rôle précis et une tâche à accomplir et c'est expliqué comme suit.

#### a) HTTP Headers

Une liaison de SOAP sur un protocole de transport est une description de la manière dont un message SOAP doit être envoyé en utilisant ce protocole de transport, Le type de liaison pour SOAP est HTTP.

#### b) SOAP enveloppe

C'est une partie du message qui permet d'expliquer comment la requête doit être traitée et elle présente aussi les éléments contenus dans le message.

#### c) SOAP Headers

Les têtes SOAP ont été conçus en prévision de la participation d'autres nœuds de traitement appelés intermédiaires SOAP, le long d'un chemin de message à partir d'un émetteur SOAP initial à un récepteur SOAP final.

#### d) SOAP Body

L'élément Body ou corps ne reflète pas de structure XML explicite. On trouve à l'intérieure du corps :

- L'enveloppe du message
- L'espace de nommage pour l'enveloppe SOAP.
- La définition de type d'encodage du message.
- L'espace de nommage pour les types de variables.
- L'espace de nommage de l'appel des méthodes qui contient a son tour :
  - corps.
  - L'espace de nommage pour la fonction getStateName.
- On spécifie un argument à passer à getStateName
  - Gestion d'erreurs.

On appelle service SOAP une application dont les méthodes sont accessibles via SOAP. Ce service peut être développé dans n'importe quel langage de programmation. On a la requête supposée coté client et la réponse supposé coté serveur , si on suppose que la requête porte une opération et une méthode pour la résoudre .Les deux sont formulés sous forme d'un message SOAP, donc ce message est alors envoyé au service SOAP qui va a son tour exécuter la méthode précisée dans la requête et retourne ensuite un message SOAP dont le corps « body » contient le résultat de l'opération.

#### c. Avantages ET Inconvénients

Les Avantages du standard SOAP sont :

- La simplicité.
- La portabilité
- La convivialité avec le pare-feu.
- L'utilisation de normes ouvertes.
- L'interopérabilité.
- L'acceptation universelle.

Les inconvénients sont

- Sérialisation en valeur et non par référence
- Trop de dépendance sur le http.

## I.4.2. WSDL « Web Services Description Language »

Lorsqu'un service est créé, cette création doit être d'une façon typique parce qu'on veut que d'autres personnes l'utilise.

Pour eux de l'exploiter, ils ont besoin de savoir quelles sont les informations à envoyer au service, quelles informations le service vont renvoyer, et où trouver le service dans la première place. Il est plus utile d'avoir un standard, de préférence d'origine humaine et le format de cette information lisible par la machine.

### a. Définition

Après d'énormes recherches dans le domaine des services web on distingue deux types de descriptions, les descriptions syntaxiques et les descriptions sémantiques. La description des services web WSDL fait parti de la première et ce concept veut dire décrire le contrat entre le client et le serveur en décrivant un service web et permettant de l'invoquer. WSDL permet la représentation d'un service web d'une manière plus abstraite afin qu'il aura la possibilité de la réutilisation. [Miller et al, 2004]

Comme SOAP, WSDL est écrit en XML. Cela rend la plate-forme de documents WSDL indépendante. De nos jours la, la plupart des langages de programmation et les plateformes disposent d'outils d'analyse XML, donc peu importe la langue ou de la plateforme que vous utilisez, vous devriez être capable d'analyser les fichiers WSDL. [9]

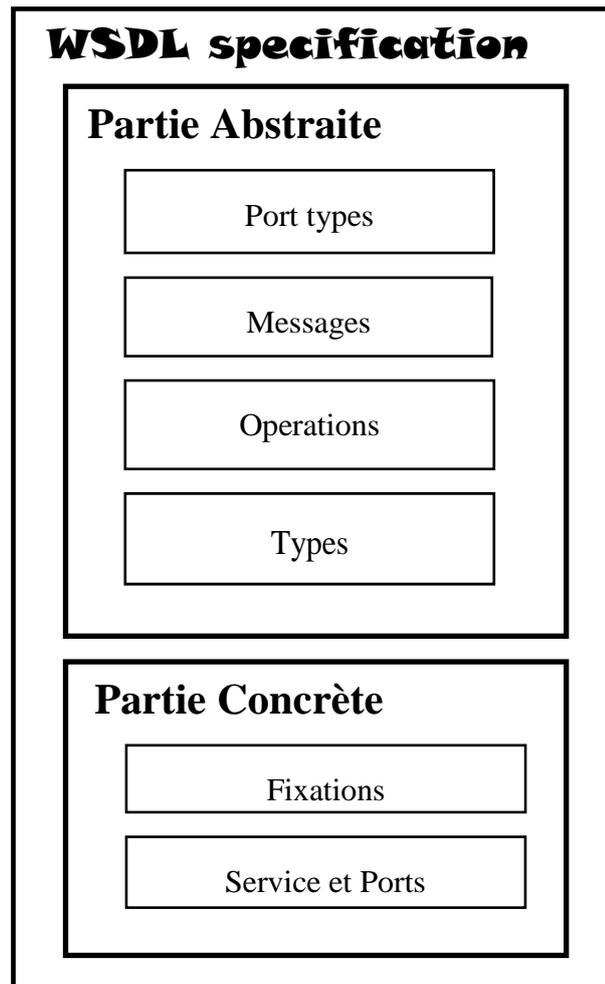
### b. Fonctionnement ET Structure de WSDL

Il à une forme d'un fichier XML donc sa structure est comme suis :

```
<?xml version=" 1.0 " encoding=" utf-8 " ?>
<definitions>
  <types>!--abstract data types</types>
  <message>!--message structure</message>
  <portType>!--Web Service Interface </portType>
  <binding>!--how the service is accessed</binding>
  <service>!--who provides the service</service>
</definitions>
```

WSDL ne présume pas que les échanges auront lieu en utilisant une forme particulière de communication , l'interface qui nous montre la structure de description est comme suit :

- La partie Abstraite : elle décrit
  - a) Les messages dont elle envoie  
Et reçoit.
  - b) l'opération associe un modèle  
d'échange de message à un ou  
plusieurs messages.
- La partie Concrète : elle spécifie
  - a) Le transport des détails de  
Format pour les interfaces
  - b) Un port associe une adresse de  
réseau à une attache.



**Fig I.2** Structure de l'interface de description des services

Avec WSDL, on peut automatiser la génération de procurations pour les services Web dans un véritable Langage et de manière indépendante de la plateforme. Comme le fichier IDL pour DCOM et CORBA, un fichier WSDL est un contrat entre le client et le serveur. Une analogie avec CORBA peut être faite, en effet WSDL est la représentation XML du langage IDL (description d'interfaces). WSDL a deux rôles prépondérants:

- Il sert de référence à la génération de Proxys.
- Il assure le couplage entre le client et le serveur par le biais des interfaces.

### c. Les Types de Description

La description des services web au début est résolu par l'entreprise Microsoft qui a adopté le standard SDL (Service Definition Language) après elle la modifiée pour le rendre un standard nommé SCL (SOAP Contract Language), il y a aussi NASSL (Network Accessible Service

Spécification Language) du IBM. Deux types de description des interfaces des services web sont connus :

### ❖ La Description par Comportement

Puisque l'interface WSDL seule ne peut pas réussir la tâche de décrire les services complexes et la description de collaborations entre plusieurs services web reflète la requête de certains processus complexes donc l'interface responsable de son rôle est de décrire le comportement d'un service relativement à un partenaire spécifique.

Dans ce type on extrait deux paradigmes ou disant deux approches l'Orchestration et la Chorégraphie, la première qui définit le processus automatique d'organisation, de coordination, et de gestion de systèmes informatiques complexes, de middleware<sup>8</sup> et de services, la deuxième qui définit la description des interactions entre plusieurs partenaires, et ce d'un point de vue global.

### ❖ La Description par sémantique

Le fait de décrire les services web à partir de leurs entrées et sorties, pré conditions, post conditions pose des problèmes au niveau des opérations des interfaces WSDL puisque il est possible d'avoir des opérations de même données mais de différentes fonctions. La recherche nous a fait donc connaître quatre spécifications :

- **WSDL-S :**

IBM et l'Université de la Géorgie ont élaboré conjointement des services Web Sémantique (WSDL-S) un document de la note technique qui prescrit un mécanisme pour annoter des services Web avec la sémantique. [10]

- **SAWSDL**

Il définit un ensemble d'attributs d'extension pour le Web Services Description Language et XML qui permet la description de la sémantique supplémentaire de composants WSDL. Les annotations sémantiques pour WSDL et le schéma XML (SAWSDL) ne spécifient pas une langue pour représenter les modèles sémantiques. Au lieu de cela, il fournit des mécanismes par lesquels les concepts des modèles sémantiques, généralement définies en dehors du document WSDL, peuvent être référencés dans les composants WSDL et XML à l'aide d'annotations. [11]

---

<sup>8</sup> Middleware : C'est un logiciel qui permet à différentes applications d'échanger et d'inter opérer.

- **OWLS**

Le « Web Ontology Language pour les services (OWL-S) » est un ensemble de base de constructions de langage de balisage pour décrire les propriétés et les capacités des services Web sous forme non équivoque, interprétable par ordinateur. OWL-S est basée sur les ontologies d'objets et de concepts définis en utilisant le Web Ontology Language « OWL ».

- **WSMO**

Le « Web Service Modeling Ontology » apparu par « Roman et al » en 2005, il constitue le fondement conceptuel et un langage formel pour décrire sémantiquement tous les aspects pertinents de services Web afin de faciliter l'automatisation de la découverte, la combinaison et l'invocation des services électroniques sur le Web.

### **I.4.3. UDDI « Universal Description Discovery and Integration »**

#### **a. Définition**

Lorsque des problèmes classiques qui gênent les systèmes sont présents tel que la manière qu'on peut indiquer qu'un service est disponible sur un réseau, comment le découvrir et le publier, alors la solution principale est l'UDDI créé par IBM, Ariba et Microsoft en première version septembre 2000 puis sa deuxième version en juin 2001 et la troisième en juillet 2002, qui est un annuaire orienté Business. Il est construit à partir de l'infrastructure des services web, ses données sont décrites en XML et il est accessible par messages SOAP en côté architectural et par « java API » en côté java. Afin de mettre à disposition un nouveau service, un fichier créé décrit en XML appelé « Business Registry » est concerné de décrire ce service. [12]

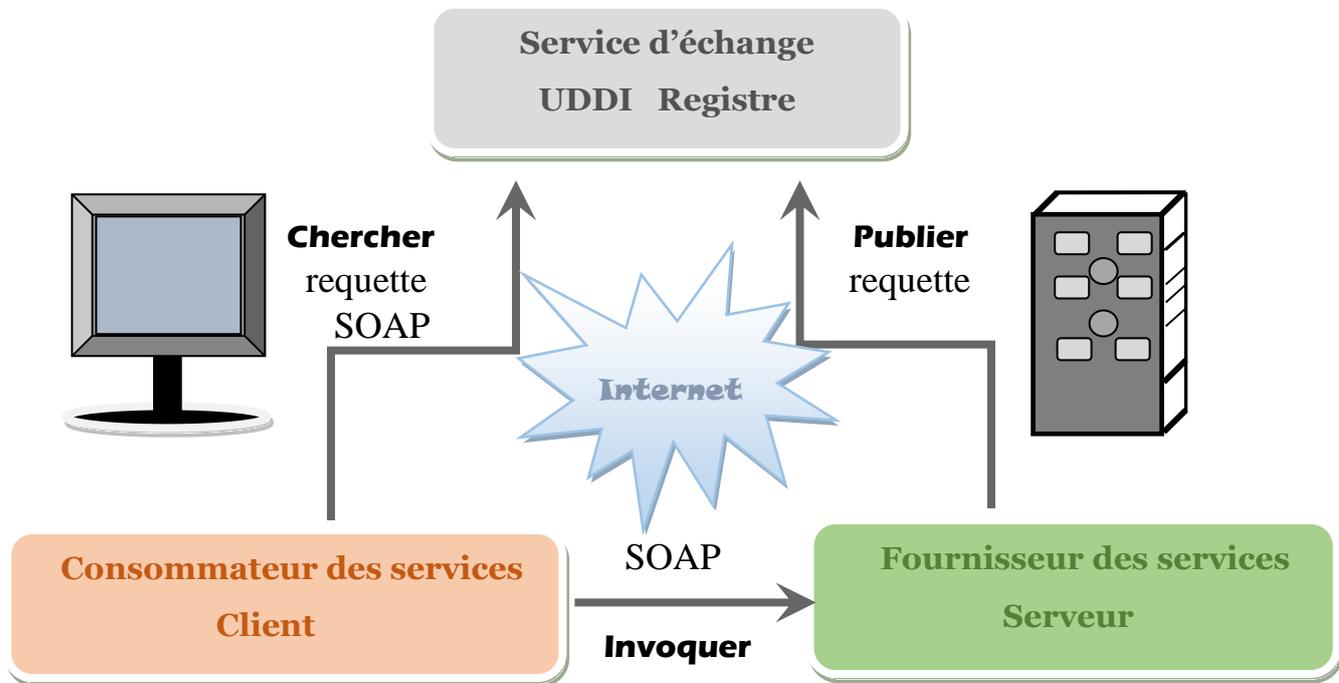
#### **b. Intérêts et Fonctionnement d'UDDI**

Un registre UDDI est défini comme un méta service qui utilise les protocoles SOAP et HTTP et qui suit un modèle d'échange de données basé sur le langage XML ce dernier permet de :

- Publier la description d'un service web.
- Découvrir les services web disponibles.
- L'enregistrement des services.
- La découverte des informations sur les services.

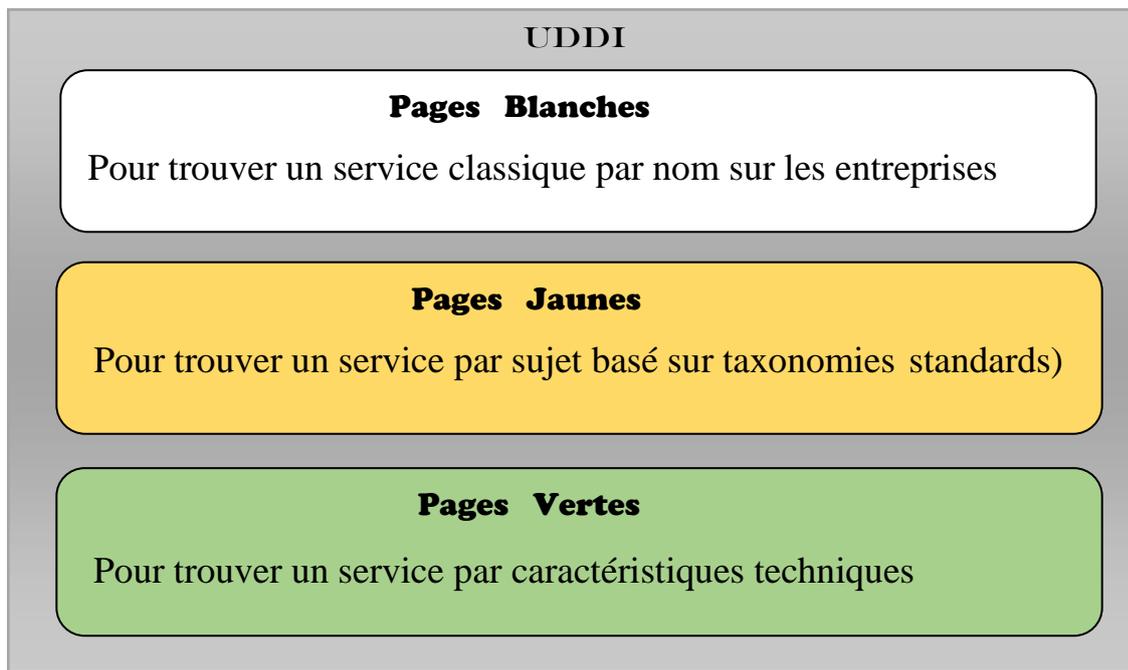
L'annuaire UDDI ne stocke pas les descriptions WSDL, et n'oblige pas à utiliser WSDL. Il fait décrire quatre types d'entités différentes qui sont :

- Les Businesses(en gros les entreprises).
- Les Business services(les services fournis par les entreprises au sens non technique du terme).
- Les Modèles techniques (la description abstraite des services proposés).
- Les fixations des services (l'implémentation des services).



**Fig I.3 Recherche des services par l'annuaire UDDI**

Plusieurs structures d'UDDI sont proposées par W3C afin de les publier par des fournisseurs des services, et puis ces structures sont formalisées en XML, elles proposent les trois types d'informations citées déjà en haut et qui sont les pages blanches concernées par les informations de contacts sur les entreprises et les pages jaunes concernées par les informations de classification des services. Il est organisé selon trois méthodes comme c'est montré dans la figure suivante.



**Fig I.4 Structure de l'Annuaire UDDI**

## I.5 Enregistrement et Publication d'un Service Web

Un enregistrement de service appelé SRV « Service Registration Verify » est une catégorie de données du DNS<sup>9</sup> d'Internet qui vise à indiquer quels sont les services disponibles. [13]

Autrement défini Le « SRV » est un enregistrement de ressource DNS (Domain Name System) utilisé pour identifier les ordinateurs qui hébergent des services spécifiques.

Tout est décrit en XML et de même un « service Binding<sup>10</sup> » peut faire référence à la partie concrète. Chaque élément est identifié par un numéro au sein d'un annuaire appelé UUID (Universally Unique ID). Ces derniers sont utilisés pour lier les éléments entre eux par exemple un « tModel » avec un « BusinessService ».

<sup>9</sup> DNS : En anglais Domain Name System qui est un service permettant de traduire un nom de domaine en informations de plusieurs types qui y sont associées, notamment en adresses IP de la machine portant ce nom.

<sup>10</sup> Service Binding : Le service de liaison vous permet de configurer les ports centralisés

### A. Business Entity

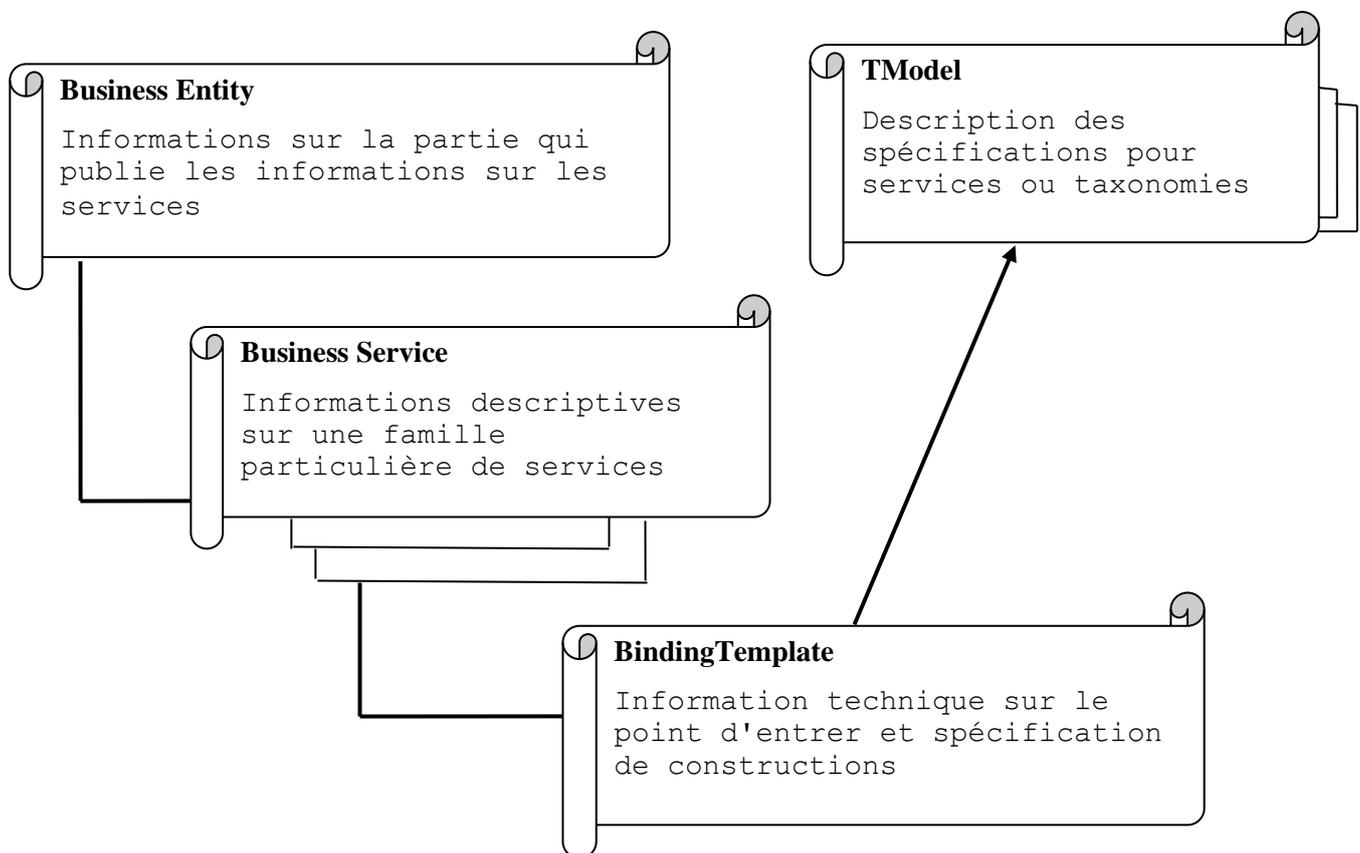
Elle représente la description d'une entreprise et contient des informations de haut niveau, destinées essentiellement à des humains, elle peut contenir des informations de classification.

### B. Business Service

Elle représente la description d'un service et réalise le lien entre une interface de service (un tModel) et son implémentation par l'intermédiaire de sous-éléments bindingTemplates<sup>11</sup>.

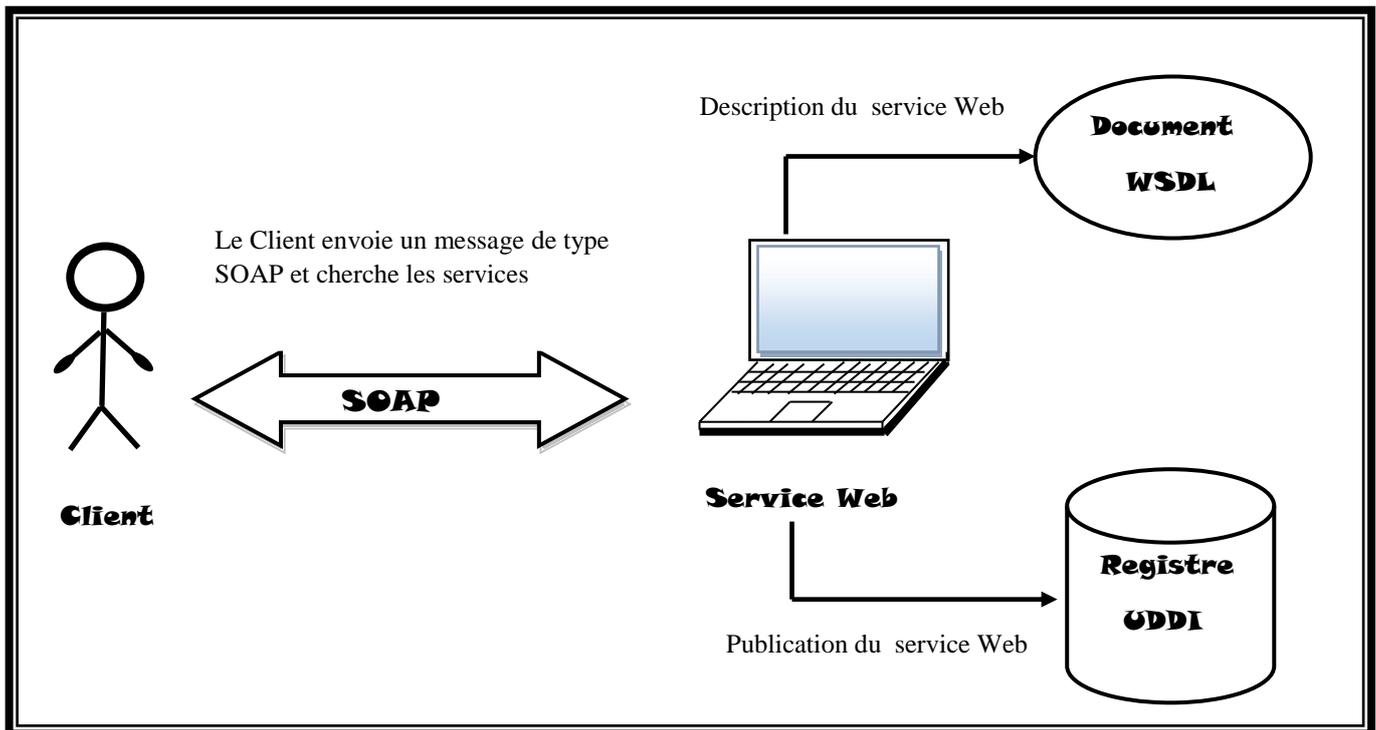
### C. TModel

Il décrit l'interface d'un service et se contente de pointer vers une description externe.



**Fig I.5 Les principaux composants descriptifs d'enregistrement**

<sup>11</sup> bindingTemplate : c'est se contente de pointer vers une description externe.



**Fig I.6 L'Architecture et les standards des services web**

## I.6 Recherche et Découverte des Services Web

### I.6.1 Définition

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. Cette sémantique est apportée grâce aux ontologies une des technologies importantes du Web sémantique. Ainsi, des agents logiciels peuvent être développés afin de raisonner sur ces ontologies rendant la découverte des services Web dynamique et automatique. [14]

[Toma et al, 2005] Définissent la découverte comme étant le processus qui prend en entrée une requête utilisateur et retourne une liste de ressources ou services, pouvant combler éventuellement le besoin décrit.

La découverte sémantique des services fait :

- Déterminer les services les plus pertinents qui correspondent à une requête donnée.
- Rendre Les implémentations courantes basées sur une représentation ontologique des opérations, inputs et outputs.
- Retourner une liste de services classés.

## I.6.2 Etat De L'Art

L'utilisateur cherche les services web et on recherche toujours à fournir un mécanisme fondé sur une recherche syntaxique. Toutefois, cette solution présente des limites dont l'importance croît avec l'évolution pléthorique des services Web. Le but est d'en assurer une meilleure interprétation et par la suite garantir une découverte efficace en améliorant la qualité des résultats obtenus.

Ces travaux s'inscrivent dans trois catégories d'approches : l'approche algébrique, l'approche déductive et l'approche hybride qui vise à bénéficier des deux précédentes. Cette dernière comprend entre autres quelques travaux adoptant un raisonnement à partir de cas i.e. (Case Based Reasoning, CBR).

L'objectif de ces travaux est de capitaliser sur l'expérience en permettant de réutiliser les solutions des cas similaires à un cas cible représentant une requête. Néanmoins, la plupart des travaux en question se limitent d'une part, à la découverte des services Web décrits selon l'ontologie OWL-S, et d'autre part, aux propriétés fonctionnelles des services. Nous pensons alors qu'une approche de découverte de services Web alignée avec les standards W3C et couvrant différents aspects liés à cette problématique, mais aussi visant à optimiser le temps de découverte, demeure nécessaire pour améliorer la performance du processus de découverte.

### a. Les Approches de Découverte des Services Web

Plusieurs approches sont connues aujourd'hui mais selon des critères qui gèrent et contrôlent cette phase de découverte on distingue trois critères dont chacun nous offre ses propres approches :

- Critère d'architecture (Centralisation /Distribution) :

La majorité des applications web ou si on peut dire des services web fournis autour du monde sont des architectures centralisées mais accessibles de partout. Par contre d'une manière générale sur un réseau et en cas de nécessité spécifiquement d'accéder à une application ou généralement de rechercher un service web l'architecture

distribuée semble un choix à étudier sérieusement. L'approche [Patil et al, 2004] est de nature distribué alors que l'approche [Clement et al, 2004] est de nature centralisé.

- Critère d'automatisation : qui mesure le pourcentage d'implication de l'utilisateur lors de la phase de découverte, et on distingue deux approches principales une manuelle et l'autre automatique [Garofalakis et al, 2004].
- Critère de Matchmaking : qui s'intéresse à l'algorithme d'appariement de la requête du client avec la description d'un service Web. Dans ce passage on parle des approches fonctionnelles et des approches non fonctionnelles. Nous nous intéressons essentiellement dans notre travail aux premières approches et exactement aux approches sémantiques et plus précisément les sémantiques hybrides qu'on va citer.

### ➤ Types d'Approches

Les Services sont vus comme une fonction ayant des paramètres d'entrée et de sortie sémantiquement annotés par des concepts d'une ontologie de domaine et des conditions préalable, et bien les différents types de méthodes exposées sont :

#### 1. Syntaxiques

Le principe général des approches basées sur la syntaxe des descriptions des services est la comparaison syntaxique entre la requête de l'utilisateur et les descriptions syntaxiques (WSDL) des services Web. L'origine de cette approche est issue du domaine de recherche d'information.

#### 2. Sémantiques

Les ontologies sont le modèle utilisé pour la représentation sémantique des services Web(SWSs), elle permet d'établir des relations sémantiques entre les différents concepts d'un domaine. La découverte sémantique par des approches vise à faire une sélection et rangement en tenant compte des préférences de l'utilisateur.

Il existe trois sous approches sémantiques qui sont :

- Sémantiques Logiques

Cette catégorie utilise des concepts d'ontologies et des règles logiques. Les degrés d'appariement sont déterminés de différentes façons et en fonction de la sémantique des éléments des descriptions à appairer. Il existe principalement trois approches d'appariement :

- **IO-matching** : appelé aussi « IO-matching de profil de service » ce type d'appariement est déterminé à partir des données sémantiques des paramètres de service : les entrées : inputs (I) et les sorties outputs (O).
- **PE-matching** : Ce type d'appariement est déterminé à partir d'appariements sur des prés conditions (P) et des effets (E) des services et des requêtes.
- **IOPE-matching** : Ce type d'appariement est déterminé à partir d'appariements sur les données sémantiques des inputs (I), des outputs (O), des prés conditions (P) et des effets (E) des services et des requêtes.

Parmi les approches existantes et développées au paravent on cite

:

- **WSC (une approche basée sur DAML<sup>12</sup>)**

Dans l'appariement sémantique de WSC « Web Services Capabilities) Un service publié correspond à une requête lorsque respectivement tous les outputs et les inputs de la requête correspondent à des outputs et inputs du service publié. L'approche se base sur un IO-matching donc un processus d'appariement qui considère juste les inputs et les outputs.

- **OWLS-M (OWL-S Matchmaker)**

Une approche proposée basée sur l'IOPE-matching, Le résultat est soit un échec s'il y a échec d'appariement soit une valeur numérique dite (RANK).Quartres degrés d'appariement pour l'évaluer sont (fail,unknown,subsumes,equivalent).

- **ALS (Automatic Location of Services)**

Pour une découverte sémantique (dite localisation sémantique) des services pour répondre a une requête cela se fait en deux phases une de la découverte et l'autre de la contractualisation, cette approche propose six degrés d'appariement appelés « degrés d'appariement intentionnel ».

- Sémantiques Non Logiques

Certaines approches de découverte se fondent sur un appariement non logique. Ce dernier utilise des mécanismes syntaxiques, structurels et numériques tels que le concept de distance numérique,

---

<sup>12</sup> DAML : est développé comme une extension de XML et RDF

l'appariement de graphes structurés et autres. L'idée principale est d'exploiter la sémantique implicite plutôt que la sémantique explicite, On va citer :

- **iMatcher1**

Le « iMatcher1 » offre un système de découverte non logique grâce à un matchmaker<sup>13</sup> (apparieur) syntaxique basé sur les profils de services. Il prend en entrée un ensemble de profils de service spécifiés en OWL-S qui sont stockés sous forme de graphes RDF sérialisés dans une base de données RDF avec une extension appelée iRDQL.

- **DSD-matchmaker**

Le « DSD-matchmaker » permet une découverte basée sur l'appariement de graphes entre deux descriptions de service spécifiées dans le langage de description de service orienté objet.

- Sémantiques Hybrides (Logiques et Non Logiques)

Les approches hybrides utilisent une combinaison de mécanismes logiques et non logiques, Il y a un matchmaker appelé « **LOG4SWS.KOM** » et aussi « **FUSION** », et le célèbre système « **OWLSMX (Version 1)** » qui contient cinq matchmakers :

- ❖ **Extended jaccard**

Le coefficient de Jaccard binaire mesure le degré de chevauchement entre les deux ensembles. Par exemple étant donné des vecteurs binaires d'indicateurs de deux ensembles  $\mathbf{X}_a=(\mathbf{0},\mathbf{1},\mathbf{1},\mathbf{0})$  et  $\mathbf{X}_b=(\mathbf{1},\mathbf{1},\mathbf{0},\mathbf{0})$ .

Le cardinal de leur intersection est 1 et le cardinal de leur union est 3 ce qui rend leur coefficient de Jaccard 1/3.

La formule est comme suis :

$$g^{(J)}(x_a, x_b) = \frac{x_a^T x_b}{\|x_a\|_2^2 + \|x_b\|_2^2 - x_a^T x_b}$$

Une autre mesure de similarité très liée à Jaccard est le coefficient **Dice** ( ) qui peut être obtenu à partir du coefficient de Jaccard en ajoutant à la fois le numérateur et le dénominateur.

<sup>13</sup> Matchmaker : C'est la mise en relation de deux individus par affinités

$$g^{(D)}(x_a, x_b) = \frac{2x_a^T x_b}{\|x_a\|_2 + \|x_b\|_2} x_a^T x_b$$

### ❖ Jonson Shannon

C'est une méthode populaire pour mesurer la similarité entre deux distributions de probabilité, basé selon [Kullback-Leibler] sur la divergence, la similitude est donnée comme suit :

$$JSD_{\pi_1, \dots, \pi_n}(P_1, P_2, \dots, P_n) = H(\sum_{i=1}^n \pi_i p_i) - \sum_{i=1}^n \pi_i H(p_i)$$

Où  $\pi_1, \dots, \pi_n$  les poids qui sont sélectionnés pour les distributions des probabilités  $P_1, P_2, \dots, P_n$ , et  $H(p_i)$  l'entropie de Shannon pour la distribution  $P$ .

### ❖ Cosine

C'est une mesure de similarité entre deux vecteurs d'un espace, la distance Cosinus est un terme souvent utilisé pour le complément dans l'espace positif, Le cosinus de deux vecteurs peut être obtenu en utilisant la formule de produit scalaire euclidien:

$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \|\mathbf{b}\| \cos \theta$$

Nous avons deux vecteurs A et B donc la formule de Cosinus est :

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=0}^n A_i \times B_i}{\sqrt{\sum_{i=0}^n (A_i)^2} \times \sqrt{\sum_{i=0}^n (B_i)^2}}$$

La similitude résultant varie de (-1) qui signifie le contraire (1) qui signifie le même et a (0) indiquant orthogonalité et entre les valeurs c'est la dissemblance.

### ❖ Information Loss

La perte intensionnel d'informations basée sur la métrique de similarité on la donne le signe LOI « loss of information » est calculée par la formule suivante :

Soit  $\left\{ \begin{array}{l} \mathbf{OUT} : \text{les données de sortie (output)} \\ \mathbf{IN} : \text{les données d'entrée (input)} \\ \mathbf{R} : \text{la requête} \\ \mathbf{S} : \text{le service} \end{array} \right.$

$$\text{Similitude}(S, R) = 1 - \frac{\text{Loi}_{IN}(R,S) + \text{Loi}_{OUT}(R,S)}{2}$$

$$\text{LOI}_x(R, S) = \frac{|PC_{R,x} \cup PC_{S,x}| - |PC_{R,x} \cap PC_{S,x}|}{|PC_{R,x}| + |PC_{S,x}|}$$

Avec  $x \in \{IN, OUT\}$ ,  $PC_{R,x}$  et  $PC_{S,x}$  ensemble de composants primitifs dans l'expression logique (inputs/outputs) = (entrés/sorties) de concept de la requête R et du service S.

## I.7 Conclusion

Nous Avons présenté dans ce chapitre tout ce qui concerne Les Services Web qui s'annoncent plus que jamais comme la solution optimale aux problèmes d'échanges de données et d'intégration d'applications. Malgré les efforts qui ont été mis en œuvre par l'architecture « SOA » il reste beaucoup à faire. En effet, l'aspect sécurité est presque inexistant et la performance est faible en le comparant avec celle des anciennes telles que RMI et CORBA.

Les approches présentées en ce chapitre ont toujours leur poids dans le domaine de la découverte et malgré quelques bons résultats de quelques unes il reste toujours l'envi de rechercher et de trouver autres approches qui donnent de meilleures performances y compris notre approche et aussi d'autres que nous allons en parler dans le prochain chapitre.



CHAPITRE II :  
CONCEPTION ET  
IMPLEMENTATION

## Chapitre II :

# CONCEPTION ET IMPLEMENTATION

### II.1 Introduction

Dans ce chapitre nous décrivons la phase de réalisation de notre application en implémentant l'environnement de recherche.

En premier lieu on traite quelques aspects relatifs à la modélisation de la conception de l'application. En deuxième lieu l'implémentation y compris les outils et langages utilisés en développement, après on va comparer et discuter nos résultats avec d'autres approches déjà vus et même les récemment développées.

### II.2 La collection de test

Nous décrivons dans ce niveau la base employée pour diriger notre approche, nous avons utilisé un extrait de corpus qui est développé par le centre allemand pour la recherche en intelligence artificielle DFKI.

Alors nous avons un ensemble de services web décrit à travers des documents owls, ces documents comprennent « **1007 services web** » qui sont à leur tour regroupés en 07 classes :

- Le domaine d'éducation
- Le domaine médical
- Le domaine de nourriture
- Le domaine militaire
- Le domaine de voyages (tourisme)
- Le domaine de communication, et le domaine d'économie.

L'annuaire UDDI d'IBM fait référence à la plupart de ces services web qui sont traduits du format WSDL en format OWLS de façon semi-automatique.

La base propose aussi un ensemble de requêtes environ **29 requêtes** réparties sur les 07 classes et aussi modélisées sous forme de document OWLS. Pour chaque document OWLS qui représente un service ou qui représente une requête a une partie appelée « profil » contenant des éléments profil dites '**has input**' et '**has output**' ces derniers sont employés comme entrées pour le module de découverte des services web.

Chaque document OWLS) qui représente un service web a sa propre étiquette accréditée par des experts humains pour qu'il soit considéré relevant ou bien non relevant par rapport à une requête donnée.

Autrement dit, l'étiquette proposée est de nature binaire (relevant ou non) , et elle nous permet de faire le calcul des rappels et des précisions de notre nouvelle approche. La base offre aussi un ensemble d'ontologies pour décrire les services et les requêtes, chaque classe de services possède une ou plusieurs ontologies.

Comme exemple, nous pouvons donner dans ce qui suit toutes informations afin de décrire un service web et une requête.

### **Requête**

**Cette requête cherche les noms des auteurs d'un roman donné.**

**Concepts d'entrée : novel**

**Concepts de sortie: author**

### **Service**

**Ce service donne les pays où il y a une activité précise donnée.**

**Nom de service : activity\_town**

**Concepts d'entrée : activity**

**Concepts de sortie : town**

## **II.3 La Conception**

Notre objectif est de mettre en œuvre une application qui fait charger des services web déjà triés dans des documents XML et puis les trier selon notre algorithme proposé par notre approche, pour cela on donne a la suite toutes les étapes de cette conception.

### **II.3.1. Présentation de l'Approche « Aggrégation du Dominance Floue »**

Cette méthode dite aussi **FDDA** « **Fuzzy Dominated Aggregation Approach** » ou par son algorithme dite **FDDAA** « **Fuzzy Dominated Aggregating Algorithm** » a son propre algorithme et sa propre formule a fin de calculer le score finale dont nous avons besoin ensuite pour trier les services.

On parle ici du mot **dominated** qui veut dire que nous cherchons en première classe un service dont score est le plus faible possible parce que chaque score calculé par l'algorithme représente le degré avec lequel il est dominé par d'autres services.

Étant donné deux vecteurs U et V de dimension d, la fonction élémentaire dominé floue ou « EFDD Elementary Fuzzy Dominated function » et le score « FDD Fuzzy Dominated » calcule la mesure dans laquelle U est dominé par V:

$$FDD(u,v) = \frac{1}{d} \sum_{i=1}^d EFDD(u_i, v_i)$$

Là où la fonction dominée Floue « EFDD(x, y) » est définie comme suit :

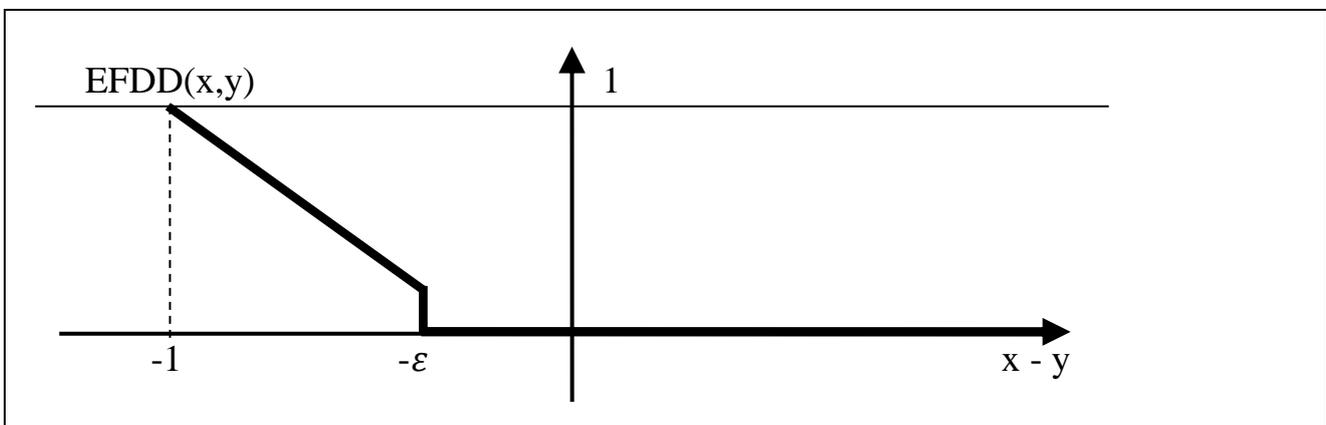
$$EFDD(x, y) = \begin{cases} 0 & \text{Si } (x-y) \geq -\varepsilon \\ \text{Si non } |x - y| - \varepsilon & \end{cases}$$

Et la Formule totale est :

$$Score(S_1) = \frac{1}{(|Base| - 1)} \cdot \sum_s FDDA(S_1, S)$$

Avec le symbole « Base » représente l'ensemble des services, V est un service appartenant à la base et  $S_1 \neq V$ .

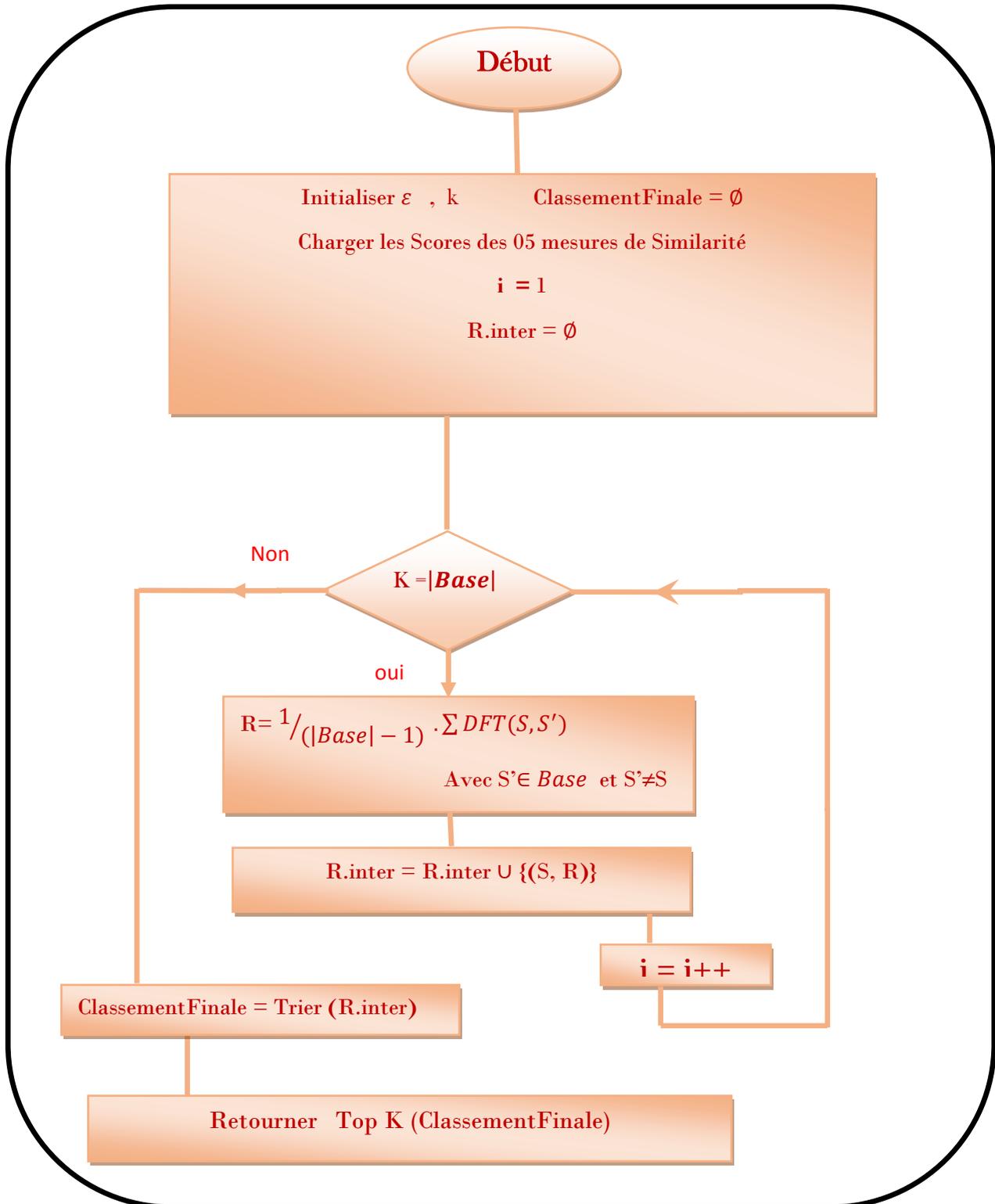
En schématisant tous cela nous avons donc le paramètre  $\varepsilon \in [0,1]$  choisi par un expert ou en faisant plusieurs études expérimentales, ce dernier permet la régulation du score dominée entre x et y.



**Fig II.2** Courbe illustrative de la Fonction EFDD

### II.3.2. Organigramme

L'organigramme suivant (figure II.1) représente les étapes menées à des résultats des précisions et rappels pour des topK multiples.





Ensuite nous cliquons sur le bouton charger et ça donne les résultats du chargement.

```

Fichier Editer Affichage Naviguer Source Refactorer Exécuter Débogage Profile Equipe Outils Fenêtre Aide
<configuration par d...
Rechercher (Ctrl+F)

Sortie - mlwsdiscovery (run)
asccon: [#WeatherProcess, SUMO.owl]
ascprop: []
chemins: [#WeatherProcess]
indexation de: #WeatherProcess onto: SUMO.owl
affichage des niveaux d'asc
asccon: [#Motion, SUMO.owl], [#Process, SUMO.owl], [#Physical, SUMO.owl], [#Entity, SUMO.owl], [#TimeDuration, SUMO.owl], [#TimeMeasure, SUMO.owl], [#ConstantQuantity, SUMO.owl], [#
ascprop: []
chemins: [#Motion, #Process, #Physical, #Entity, #TimeDuration, #TimeMeasure, #ConstantQuantity, #PhysicalQuantity, #Quantity, #Abstract]
affichage des niveaux d'asc
asccon: [#Legal-Agent, portal.owl], [#Generic-Agent, portal.owl], [#Temporal-Thing, support.owl]
ascprop: [#organization-part-of, #has-size, #headed-by, #has-size, #affiliated-person, #has-sub-unit]
chemins: [#Legal-Agent, #Generic-Agent, #Temporal-Thing]
servN*=1006
indexation de: #Organization onto: portal.owl
affichage des niveaux d'asc
asccon: [#Sports, travel.owl], [#Activity, travel.owl]
ascprop: []
chemins: [#Sports, #Activity]
servN*=1006
indexation de: #Organization onto: portal.owl
indexation de: #Surfing onto: travel.owl
indexation de: #Destination onto: travel.owl
affichage des niveaux d'asc
asccon: []
ascprop: []
chemins: []
indexation de: #LuxuryHotel onto: travel.owl
affichage des niveaux d'asc
asccon: [#Hotel, travel.owl], [#Accommodation, travel.owl]
ascprop: [null]
chemins: [#Hotel, #Accommodation]
fin indexation

base des prop des concepts[#4WheeledCar, my_ontology.owl], [#Rigid, #Wheel, #Shape, #Person, #Color, #LifeTime, #Engine, #Power, #Speed, #madeBy, #belongsTo, #Model, #hasValue, #Pro
base des chemins[#4WheeledCar, my_ontology.owl], [#Car, #Auto, #Machine, #Object]], [#1PersonBicycle, my_ontology.owl], [#Bicycle, #Cycle, #Object]], [#Price, concept.owl], [#Unt

```

**Fig II.4 Résultats du chargement et Indexation des Services web**

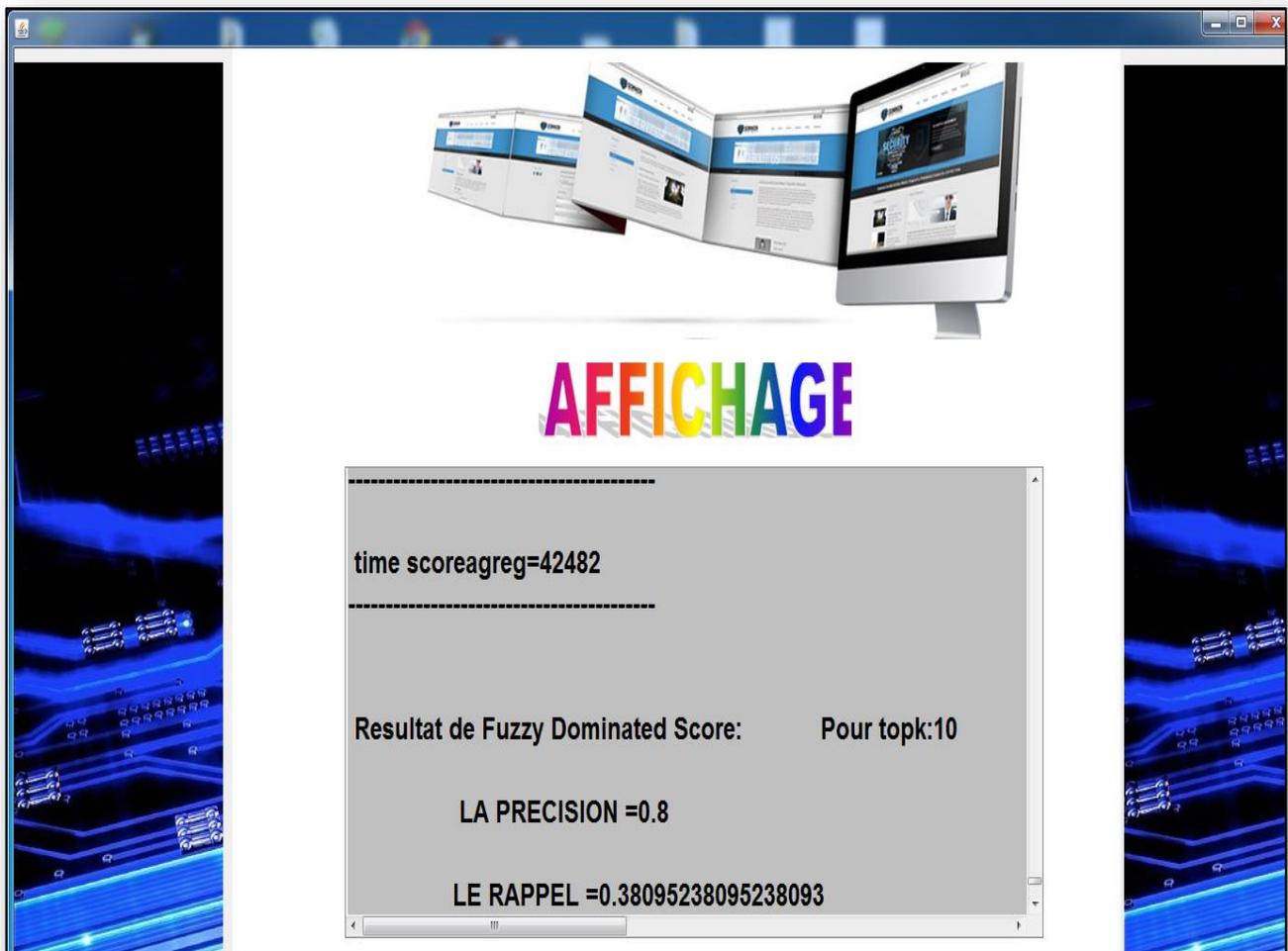


Fig II.5 Résultats de Trie de Rappel et de Précision du FDDA

## **II.5 Expérimentation**

Dans ce qui suit nous citons l'environnement et les différents outils utilisés lors de la réalisation du projet .Nous commençons par une présentation du langage JAVA, suivi par la présentation de NetBeans qui est un environnement de développement intégré (EDI) aussi l'Excel un logiciel de tabulations et calcules sans oublier la machine sur laquelle nous avons développé notre application. A la fin nous illustrons le projet par les captures écrans montrant les étapes de l'application.

### **II.5.1. Java**

C'est le nom d'une technologie mise au point par Sun Microsystems qui permet de produire des logiciels indépendants de toute architecture matérielle. Java est à la fois un langage de programmation et une plateforme d'exécution. Le langage Java a la particularité principale d'être portable sur plusieurs systèmes d'exploitation tels que Windows, Mac OS ou Linux.

### **II.5.2. Netbeans**

#### **a. Définition**

NetBeans est un environnement de développement intégré (EDI) placé en open source par Sun en juin 2000 sous licence CDDL et GPLv2. En plus de Java, NetBeans permet également de supporter différents autres langages, comme Python, C, C++, JavaScript, XML, Ruby, PHP et HTML. [6]

#### **b. Version Utilisée**

Nous avons travaillé avec la version du « NetBeans IDE 6.9.1 » sorti en Juin 2010, ajouté le support pour OSGi, le « Spring Framework 3.0 », l'injection de dépendance Java EE (JSR-299), le « Zend Framework » pour PHP, et plus facile pour la navigation du code tels que le formatage les astuces et le refactoring dans plusieurs langues.

### **II.5.3. Microsoft Excel**

Microsoft Excel est un logiciel tableur de la suite bureautique Microsoft Office, développée et distribuée par l'éditeur Microsoft. Nous avons utilisé la version Excel 2007.

Pour faire les résultats obtenus des rappels et précisions dans un Histogramme en 3D nous avons besoin de ce logiciel Excel.

## II.5.4. Machine du Développement

Nous avons travaillé sur un micro « HP Pavilion g6 Notebook PC » Pentium 4 processus Intel ® avec « Windows 7 édition Familiale Service Pack 1 » intégré et mémoire « RAM 4,00 Go » système d'exploitation « 64 Bits ».

## II.5.5. Discussion

Tout d'abord nous définissons les termes utilisés comme suit :

**Top k :** Représente le nombre des services choisis pour comparer les services pertinents triés avec les services relevants prédéfinis par les experts, nous avons choisi jusqu'à la valeur 70 puisque le nombre max des services relevant est de 75. Nous l'utilisons pour calculer le Rappel et la précision.

**Le Rappel R :** C'est le Rapport entre le nombre des services dites correctes trouvés par le système d'application et le nombre total des services dites correctes, et plus formellement :  $R = \frac{vp}{(vp+fn)}$  avec :

Vp : les vrais positifs (un résultat correct, et considéré comme étant valide par le système).

Fp : les faux positifs (un résultat erroné, mais considéré comme étant valide par le système).

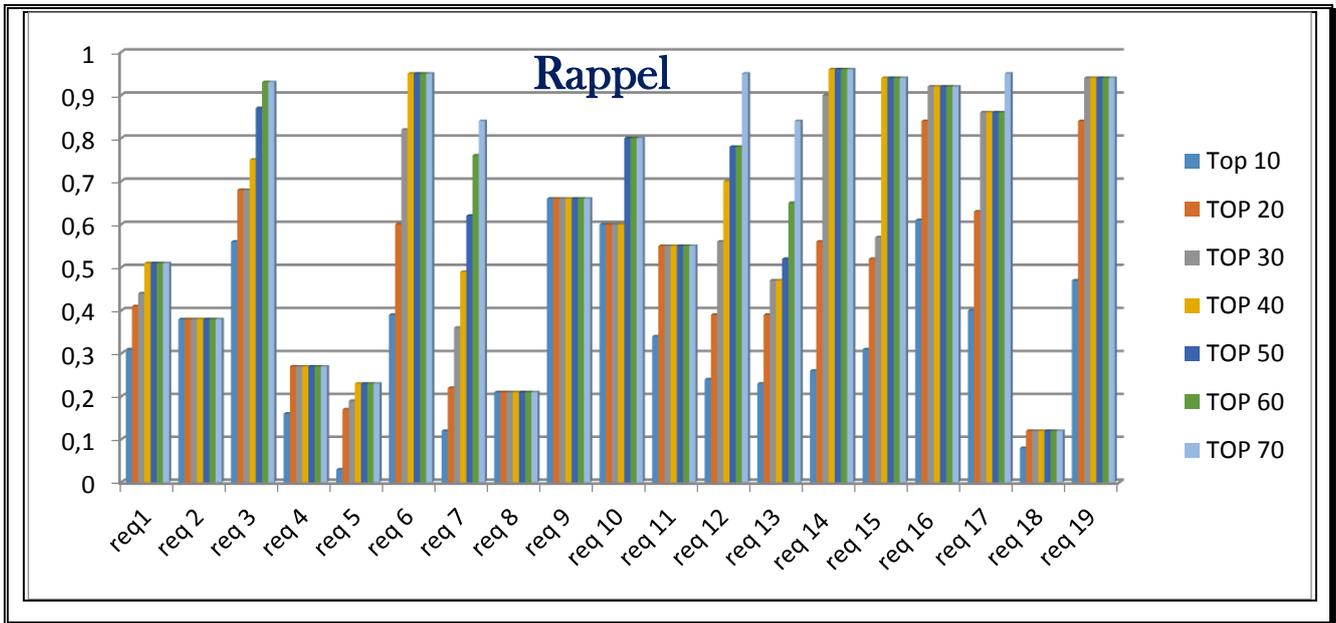
Fn : les faux négatifs (un résultat correcte et considéré comme faux par le système).

**La précision P :** C'est le Rapport entre le nombre des services dites correctes trouvés par le système d'application et le nombre total des services sélectionnés, et plus formellement :

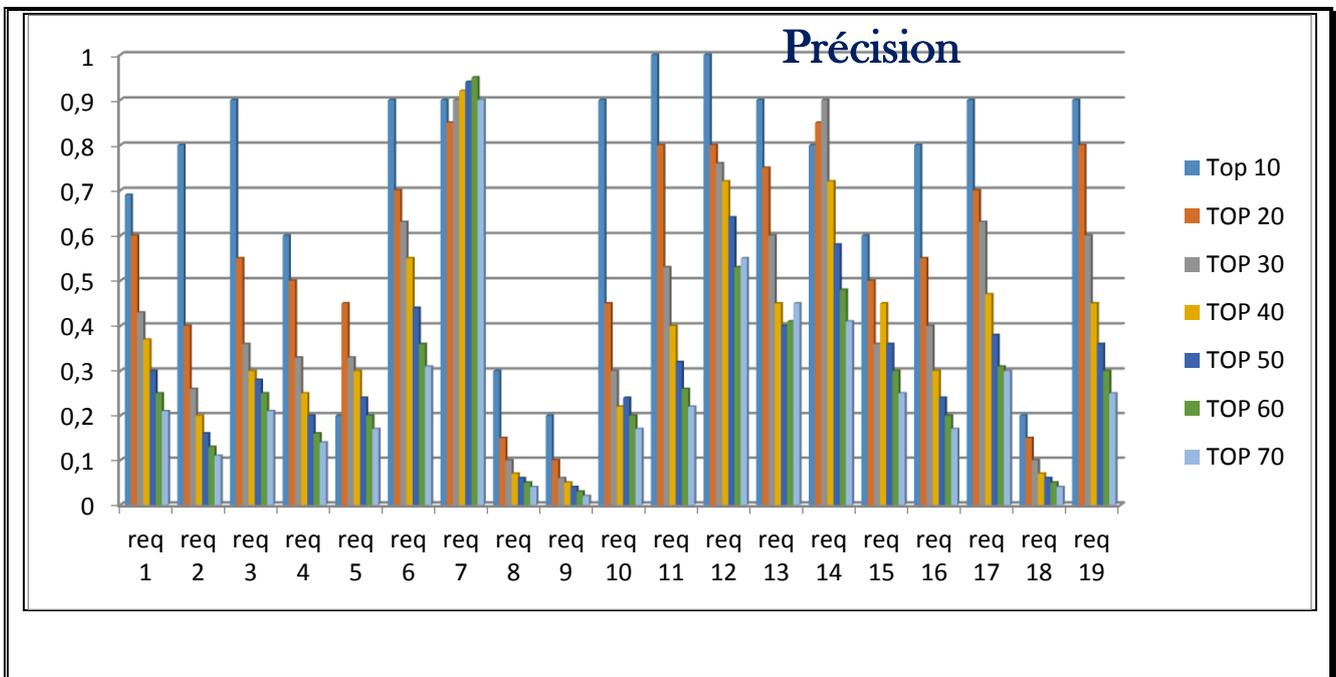
$$P = \frac{vp}{(vp+fp)}$$

Notre approche Dominance Floue a donnée des résultats de rappel et précision satisfaisantes. Nous comparons nos résultats avec celles de l'approche de nos camarades.

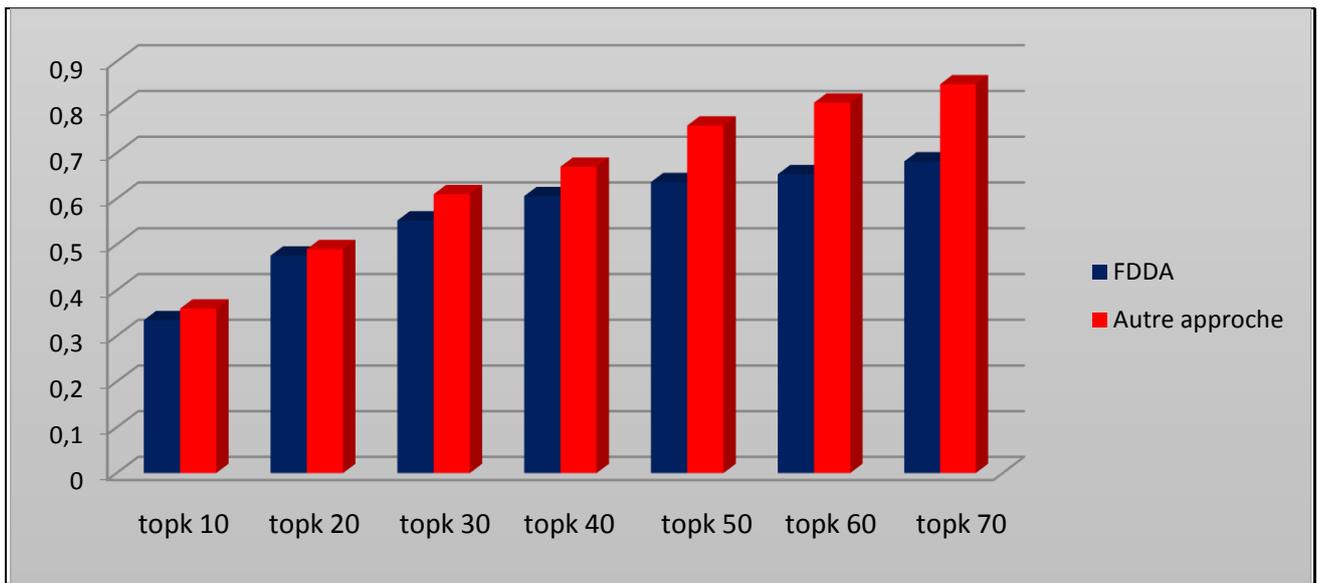
On voit dans la **Figure II.4** l'Histogramme du rappel et précision représentant les 19 requêtes :



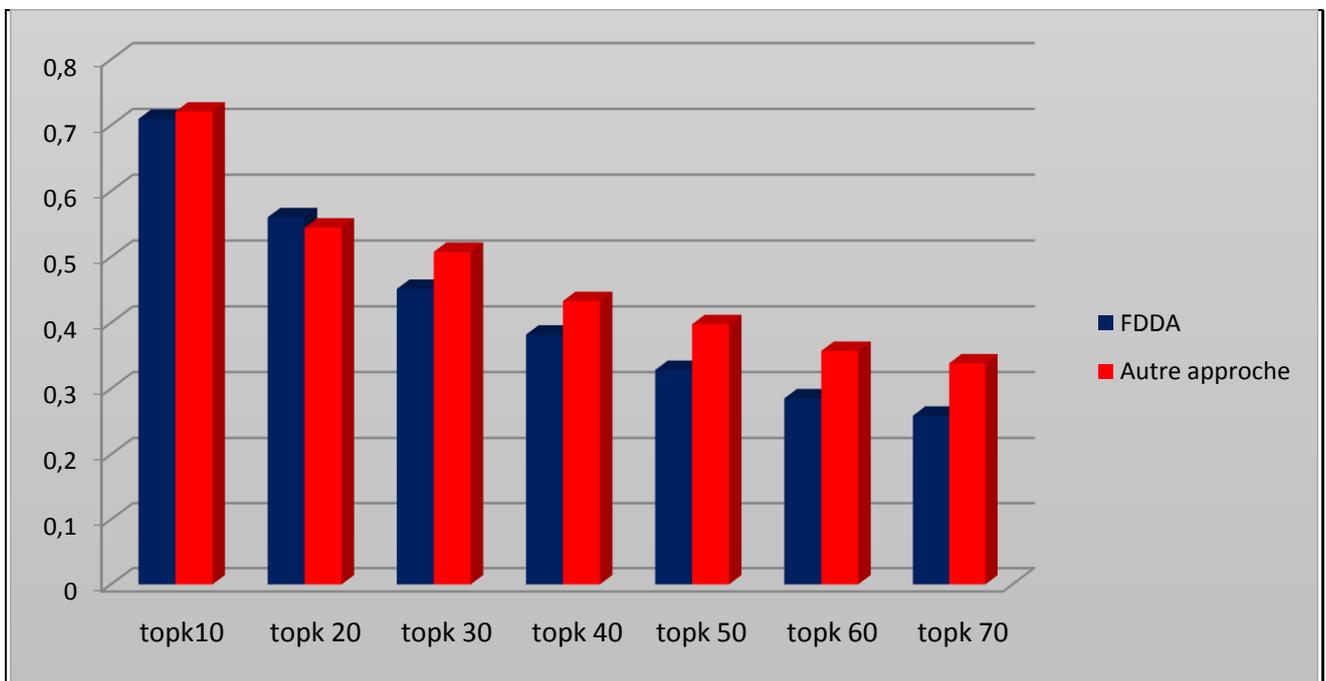
**Fig II.6** L'Histogrammes de Rappel selon sept Top K multiples



**Fig II.7** L'Histogrammes de Précisions selon sept Top K multiples



**Fig II.8** Histogramme de Rappel des moyennes des deux Approches



**Fig II.9** Histogramme de Précision des Moyennes des deux Approches

Si nous méditons dans le premier histogramme du Rappel Nous remarquons que le taux de rappel pour l'autre approche développée par Mr Ziani et Mr Bouchekara (nos camarades) est égal à **0.81** pour **top k 60** en outre son taux dans FDDA notre approche est égal à **0.65** qui veut dire moins faible ,cette remarque est la même pour tous les autres top k .

En méditant l'Histogramme de précision nous remarquons seulement dans le **top k 20** que FDDA a le taux de précision élevé par rapport au taux de l'autre approche par contre pour les autres top k c'est le contraire.

## **II.6 Les Avantages de l'Approche FDDA**

Lorsque notre approche est récemment développée, elle a apportée de nouveaux aspects et des avantages, le meilleur de ces derniers c'est :

- L'indépendance de certains paramètres tels que la taille de la liste des services triés en entrée.
- La localisation des Services offrant des compromis.

## **II.7 Conclusion**

Nous avons présenté dans ce chapitre toute l'application de découverte floue des services web.

Les expérimentations effectuées montrent que l'approche FDDA a des performances satisfaisantes pour  $K = 10$  et  $K = 20$ .



Conclusion  
GENERALE

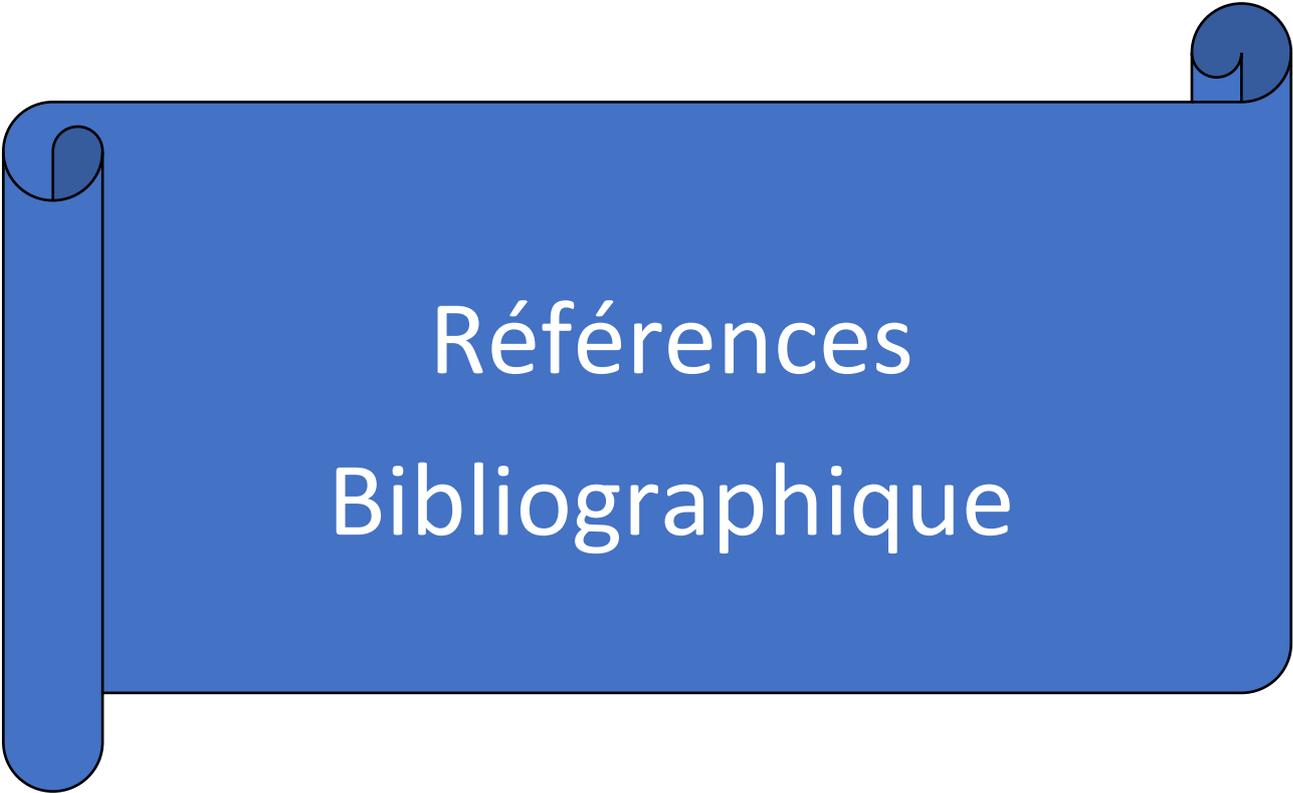
---

## Conclusion Générale

Nous avons réalisé et présenté dans ce mémoire le concept de services Web ainsi ses techniques et ses protocoles nécessaires tels que les standards célèbres et l'architecture SOA. Dans ce paradigme nous avons pensé à proposer une nouvelle approche située précisément dans la phase de découverte des services et elle est basée sur la notion de dominance floue.

Nous avons travaillé sur cette approche appelé FDDA fuzzy dominated aggregation dans notre prototype afin d'arriver à une solution claire et satisfaisante au problème de tri et de découverte où le client pose ses besoins. Les résultats obtenus sont acceptables et méritent d'être étendus afin de booster les performances.

Les travaux futurs doivent traiter le problème des égalités des scores, dans ce sens nous pouvons proposer un deuxième algorithme qui reclasse les services équivalents et nous pouvons prendre par exemple le vote à base de Condorcet ou la dominance probabiliste.



Références  
Bibliographique

---

## Références

- [Emmerich, 2002] W. Emmerich. Distributed component technologies and their software engineering implications. In ICSE '02 : Proceedings of the 24th International Conference on Software Engineering, pages 537–546, New York, NY, USA, 2002. ACM.
- [Booth et al, 2004] D.Booth,H.haas,F.McCabe,E.Newcomer,M.Champion, C.Ferris, et D. Orchard.W architecture,W3C Working group note 11.In <http://www.w3.org/TR/ws-arch/2004>
- [Hadjila et al, 2013] F.Hadjila , M.A Chikh, Composition et interopération des Services web Sémantiques
- [Gudin et al, 2003] M.Gudin , M .Hadley, N .Mendelsohn, J.J. Moreau, and H.F.Nilelsen.Simple Object access protocol(soap) 1.2.World Wide Web consortium, <http://www.w3.org/TR/soap,2003>.
- [1] D.K. Barry Web Services, les architectures orientés services,et Cloud Computing :Guide du gestionnaire Savvy(Seconde édition),2000.

- 
- [4] Dimitrios Skoutas, Dimitris Sacharidis, Alkis Simitsis, and Timos Sellis, Senior, Ranking and Clustering Web Services using Multi- Criteria Dominance relationship,
- [3] S. Hammoudi et D.Lopes , Introduction aux Services Web,2003
- [5] O.Pperrin,Université de Lorraine ,LORIA ,version 2012.
- [6] M.Gharzouli ,B.Mohamed Composition des Web Services Sémantiques ,2011.
- [7] A.Nadalin, C.kaler M.Goodner, M.Gudgin, A.Barbir, NH.Granqvist.Web Service Security : Soap Message Scurity 1.1 In <http://www.oasis-open.org/committees/wss/>,2004.
- [Miller et al, 2004] J.Miller ,K.Verma, P.Rajasekaran, A.Sheth, R.Aggaroil,and K.Sivashanmugam.wsd1-s :Adding semantics to wsdl,2004.
- [8] M.Klusch , B.Fries, K.Sycara . Automated Semantic Web Service Discovery,2005
- [9] <http://WWW.W3.ORG/2001/SW/wiki/SAWSDL>
- [10] UDDI Project, UDDI Technical White Paper, September 2000.

- [11] Alex Ferrata and Matthew MacDonald, Programming . NET Web Services, O'Reilly, September 2002. ISBN: 0-596-00250-5
- [12] M.Chelbabi , Découverte de Services Web Sémantiques : une Approche basée sur le Contexte ,18 novembre 2006.
- [Toma et al, 2005] I.Toma, D.Fensel, M.Moran, K.Iqbal, T.Strang, and D.Roman. An evolution of Discovery approaches in Gird and Web services environments., September 2015.
- [Galofalakis et al, 2004] J.Garofalakis,Y.Panagis, E.Sakkopoulos and A.Tsakalidis. Web Services Discovery Mechanisms : looking for a needle in a haystack in International Workshop on web Enginnering, 2004.