



République Algérienne Démocratique et Populaire  
Université Abou Bakr Belkaid– Tlemcen  
Faculté des Sciences  
Département d'Informatique

### Mémoire de fin d'études

Pour l'obtention du diplôme de Master en Informatique

Option : *Modèle intelligent et décision (MID)*

## Thème

*La Découverte de services web à base  
De Dominance Probabiliste*

### Réaliser par :

- Bouchekara Med Kacem
- Ziani Soheyb

Présenter le 23 Juin 2015 devant le jury composé de MM :

- MrMahfoud.H (Président)
- Mr Hadjila Fethalah (Encadreur)
- Mr Merzoug.M (Examineur)
- Mr Etchiali (Examineur)

Année Universitaire : 2014-2015

# Remerciements

Nous remercions Dieu le tout puissant qui nous a bénis et permis d'arriver là où nous sommes.

Nous tenons à exprimer nos remerciements envers toutes les personnes qui ont contribué au bon déroulement et à l'aboutissement de cette thèse.

Nous tenons particulièrement à remercier notre encadreur Mr. Fethallah Hadjila pour nos avoir encadré et orienté durant notre travail.

Nos remerciements également au :

- Président de jury : Mr Mahfoud.H
- Examineurs : Mr Merzoug.M , Mr Etchiali

D'avoir accepté d'examiner et d'évaluer notre travail.

Enfin, nous tenons à remercier nos proches et nos ami(e)s pour les moments partagés, ainsi que notre famille.

# *Dédicaces*

*Louange à Dieu, Seigneur des mondes ;*

*C'est toi que nous adorons et de toi que nous implorons secours ;*

*Que la paix et la bénédiction soit sur son dernier envoyé ;*

*A nos chers parents qui ont été présents, à chaque instant, Leur*

*irremplaçable et inconditionnel soutien nous a permis d'écarter les*

*doutes, de soigner les blessures et de partager nos joies ;*

*A notre petite famille, nos frères et sœurs;*

*A tout la famille Ziani et Bouchekara ;*

*A tous nos ami(e)s sans exception ;*

Soheyb & Kacem

## Résumé

Au cours de ces dernières années les entreprises ont largement utilisé la technologie des services Web pour faire l'interopérabilité et l'intégration de leurs applications.

Pour réaliser efficacement ces objectifs, les entreprises doivent localiser en premier lieu, les services qui répondent à leur besoins, et pour ce faire il est impératif de développer des modèles et des algorithmes assez adéquats pour cette tâche.

Les approches développées actuellement pour la découverte de services Web sont peu efficaces. Dans ce travail nous proposons une solution qui agrège plusieurs méthodes de recherche. La fusion de ces approches est faite grâce à la notion de dominance probabiliste. Cette dernière estime le pourcentage d'instances (de services) dominées par un candidat donné. Les résultats expérimentaux sont satisfaisants et démontrent l'utilité de l'approche.

**Mots clés:** architecture orientée services, découverte de services, fusion de données, agrégation probabiliste.

## **Abstract**

The web service discovery is one of the most challenging issues in service oriented computing. In this work we propose an approach that aggregates multiple retrieval methods. The fusion of these algorithms is done through the concept of probabilistic dominance. This later estimates the percentage of service instances that are dominated by a particular candidate. The experimental results are very encouraging and demonstrate the utility of the approach.

**Key words:** Web Service ranking, Web Service Discovery, Service Matching, probabilistic Dominance, Rank Aggregation, similarity measures.

## ملخص

اكتشاف خدمات الإنترنت هي إحدى أكبر التحديات في مجال الحوسبة، لكن المناهج المطورة حالياً لاكتشافها ضعيفة. ولتحقيق هذه الأهداف على نحو فعال، يجب على الشركات أولاً تحديد موقع الخدمات التي تلبي احتياجاتهم، ولذلك لا بد من تطوير نماذج وخوارزميات كافية تماماً للقيام بهذه المهمة. في هذا العمل نقترح نهجاً يجمع أساليب بحث متعددة. حيث يتم دمج هذه الخوارزميات من خلال مفهوم الهيمنة الاحتمالية. هذا يقدر في وقت لاحق النسبة المئوية لحالات الخدمة التي يهيمن عليها مرشح معين. النتائج التجريبية مشجعة جداً وتوضح فائدة هذا النهج.

**الكلمات الرئيسية:** ترتيب خدمة ويب، اكتشاف خدمة ويب، مطابقة الخدمة، الهيمنة الاحتمالية، الرتبة التجميع، تدابير التشابه.

# Table de Matière

|  |    |
|--|----|
| Introduction Générale .....  | 1  |
| Contexte .....   | 1  |
| Problématique.....   | 1  |
| Contribution .....   | 1  |
| <b>Chapitre: L'architecture SOA et les services Web</b>  |    |
| I. Introduction .....  | 4  |
| II. Définition de l'architecture SOA.....  | 4  |
| II.1 Définition Métier .....   | 4  |
| II.3 Définition Technique .....  | 4  |
| II.4 Historique de l'architecture SOA.....   | 5  |
| II.5 Caractéristiques de l'architecture SOA .....  | 6  |
| II.5.1 Couplage faible.....  | 6  |
| II.5.2 Frontières explicites .....   | 7  |
| II.5.3 indépendance par rapport à La plateforme .....  | 7  |
| II.5.4 Le langage de programmation.....  | 7  |
| II.5.5 Le modèle d'un service.....   | 7  |
| III. Concept de service Web.....   | 8  |
| III.1 Définition 1 .....   | 8  |
| III.2 Définition 2 .....   | 8  |
| III.3 Exemples d'utilisation .....   | 8  |
| III.4 Caractéristiques des services Web.....   | 9  |
| III.4.1 Un service Web est une application logicielle qui est reconnue par un URI.....   | 9  |
| III.4.2 Capacité des interfaces et liaisons d'être publiées, localisées et invoquées via le langage XML .....                                  | 9  |
| III.4.3 Capacité d'interagir avec les composantes des logiciels via des éléments XML avec l'utilisation des protocoles Internet standards..... | 9  |
| III.4.4 Composante logicielle légèrement couplée à interaction dynamique.....  | 10 |
| III.5 Avantages des services Web .....   | 10 |
| III.6 Architecture des services Web.....   | 11 |
| Architecture de référence .....  | 11 |
| III.7 Les principales technologies standards autour de service Web .....   | 12 |
| III.7.1 Le protocole SOAP .....  | 12 |
| III.7.2 WSDL .....   | 14 |

|  |    |
|--|----|
| III.7.3 L'annuaire UDDI .....  | 15 |
| IV. Recherche et découverte des services web .....                   | 17 |
| IV.1 Introduction .....  | 17 |
| IV.2 Définition.....   | 17 |
| Définition 1.....  | 17 |
| IV.3 Critères de découvert.....                                      | 18 |
| IV.3.1 Le critère d'automatisation.....                              | 18 |
| IV.3.2 Le critère d'architecture (Centralisation/distribution) ..... | 18 |
| IV.3.3 Le critère de Matchmaking .....                               | 18 |
| IV.4 Les approches.....  | 18 |
| IV.4.1 Approches syntaxique .....                                    | 18 |
| IV.4.2 Approche sémantique .....                                     | 19 |

## **Chapitre II: Conception et implémentation du prototype**

|   |    |
|---|----|
| I. Introduction.....  | 25 |
| II. Présentation du Corpus.....                                     | 25 |
| III. La Conception .....  | 26 |
| III.1 Définition de l'Approche « Dominance Probabiliste « DP »..... | 26 |
| III.2 Exemple d'application .....                                   | 27 |
| III.3 Organigramme .....  | 28 |
| III.4 Déroulement de l'algorithme .....                             | 29 |
| IV. Présentation du prototype.....                                  | 29 |
| V. Expérimentation .....  | 34 |
| Environnement de développement et outils .....                      | 34 |
| Conclusion .....  | 37 |
| Conclusion générale et Perspectives.....                            | 39 |
| Annexe.....   | 44 |

# Liste des figures

|                      |  |    |
|----------------------|--|----|
| <b>Figure II. 1</b>  | Organigramme de Dominance Probabiliste .....     | 28 |
| <b>Figure II. 2</b>  | Interface d'accueil.....                         | 30 |
| <b>Figure II. 3</b>  | Chargement de la base .....                      | 30 |
| <b>Figure II. 4</b>  | Affichage des services web utilisés.....         | 31 |
| <b>Figure II. 5</b>  | Affichage des requêtes .....                     | 31 |
| <b>Figure II. 6</b>  | le Lancement de l'algorithme .....               | 32 |
| <b>Figure II. 7</b>  | interface principale d'exécution.....            | 32 |
| <b>Figure II. 8</b>  | affichage numérique des résultats .....          | 33 |
| <b>Figure II. 9</b>  | Affichage Graphique des résultats .....          | 33 |
| <b>Figure II. 10</b> | Histogramme du score de Précision.....           | 35 |
| <b>Figure II. 11</b> | Tableau du score de Précision.....               | 35 |
| <b>Figure II. 12</b> | Histogramme du score de Rappel .....             | 36 |
| <b>Figure II. 13</b> | Tableau de Score du Rappel.....                  | 36 |
| <b>Figure II. 14</b> | table de comparaison de score de précision ..... | 37 |
| <b>Figure II. 15</b> | table de comparaison de score de Rappel .....    | 37 |

**Introduction**

**Générale**

## Introduction Générale

### Contexte

Plusieurs technologies innovantes ont été introduites ces dernières années, pour renforcer la communication et le partage des ressources. Ces tâches ont menés les experts du domaine à inventer et découvrir des nouvelles méthodes pour résoudre plusieurs problèmes informatiques.

Parmi les technologies qui ont apparus au début du 21<sup>ème</sup> siècle, et qui ont été adoptées par les entreprises, c'était la technologie des services web.

Les services web sont des briques de bases logicielles s'affranchissant de toute contrainte de compatibilité logicielle ou matérielles. Les services Web comportent de nombreux avantages, ils sont utilisables à distance via n'importe quel type de plate-forme, ils peuvent servir au développement d'applications distribuées et sont accessibles depuis n'importe quel type de clients. Les services Web appartiennent à des applications capables de collaborer entre elles de manière transparente pour l'utilisateur, Leur mise en œuvre repose sur une architecture décentralisée. Cette architecture orientée services (Service Oriented Architecture-SOA) est un style architectural fondé sur la description des services et de leurs interaction. Les services sont publiés dans des annuaires par des fournisseurs qui les hébergent. Ils sont accessibles via un réseau pour les clients qui les découvrent, les sélectionnent, les invoquent et les utilisent.

### Problématique

Un des majeurs problèmes qui peuvent faire face au client c'est la découverte des services web, consistant à satisfaire les besoins en utilisant des annuaires prédéfinis. Le problème consiste à rechercher et classer les services selon degré de mérite. En outre cette découverte doit avoir une bonne performance en termes de rappel et de précision.

### Contribution

Dans le cadre de ce mémoire, nous proposons une approche qui se base sur les probabilités. Ces dernières ont répondu des difficultés dans divers domaines pour résoudre des problèmes de décision. Nous nous inspirons de ces techniques pour décrire une approche de découverte des meilleurs services Web en se basant sur une des techniques de probabilité

qui aide le client à trouver le meilleur service disponible. Les approches probabilistes avec un bon réglage de leurs paramètres, constituent une approche intéressante pour la résolution des problèmes de découverte. De plus, ce domaine est très dynamique et ne cesse de se développer de jour en jour.

### **Plan de Travail**

**Chapitre I** : Introduit les services web et les standards associés

**Chapitre II** : Présente le prototype développé et les résultats obtenus

# **Chapitre :1**

## **L'architecture SOA et les services Web**

## I. Introduction

Le développement rapide des systèmes d'informations a flamboyé ces dernières décennies à travers la diffusion de l'accès à internet. De nouveaux modèles d'interaction ont été également marqués citons-en : la SOA (Architecture Orienté Service).

L'architecture SOA est un modèle abstrait, qui définit un système par un ensemble d'agents logiciels distribués, qui fonctionnent de concert afin de réaliser une fonctionnalité globale préalablement établie. SOA se présente comme un style architectural. Elle fournit un ensemble de méthodes pour le développement et l'intégration de systèmes dont les fonctionnalités sont développées sous forme de services.

L'approche ultime de cette vision consiste, donc, à créer des applications constituées uniquement de services qui interagissent entre eux. Dans ce cas, peu importe où est déployé le service, ce qui importe est que le service remplisse un rôle bien précis [Schreiner 2005].

## II. Définition de l'architecture SOA

L'architecture SOA peut être définie et expliquée par plusieurs définitions, ces définitions se basent sur les caractéristiques techniques de l'architecture, en outre d'autres portent attention aux aspects métiers. Les définitions suivantes clarifient les différentes vues de la SOA. Cependant, elles renvoient toutes vers un seul sens :

### II.1 Définition Métier

« L'architecture orientée service est un ensemble de méthodes techniques, métiers, procédurales, organisationnelles et gouvernementales pour réduire ou éliminer les frustrations avec les technologies d'information, et pour mesurer quantitativement la valeur métier des technologies d'information, pendant la création d'un environnement métier agile pour un intérêt concurrentiel. » [Margolisand 2007].

### II.3 Définition Technique

« L'architecture SOA est un paradigme permettant d'organiser et d'utiliser des savoir-faires distribués pouvant être de domaines variés. Cela fournit un moyen uniforme d'offrir, de découvrir, d'interagir et d'utiliser des savoir-faires pour produire le résultat désiré avec des pré-conditions et des buts mesurables. » [Josuttis 2007].

Du point de vue des applications, l'architecture orientée services permet le développement d'une nouvelle génération d'applications dynamiques ou composites. Ces applications permettent aux utilisateurs d'accéder à des informations et à des processus hétérogènes, et de les utiliser de différentes manières, notamment via le Web. Du point de vue de l'infrastructure, l'architecture orientée services permet au service Web de simplifier l'intégration des applications et des systèmes, de recombinaison et de réutiliser les fonctionnalités des applications, et d'organiser les différentes phases du processus de développement, dans un cadre cohérent et unifié. En réalité, la philosophie des SOA décompose une application monolithique en une suite de services assurant la modularité dans leurs fonctionnalités [Devaux 2008].

## II.4 Historique de l'architecture SOA

Une nouvelles philosophie de développement nommé l'architecture orientes est apparu dès les années quatre-vingts.

Le problème de l'approche objet est le fait de rendre difficile l'intégration d'applications résidentes dans plusieurs plateformes. Elle est fragile, car un service de ce système est offert comme une méthode d'une classe implémentée par un objet [Bieberstein 2008].

Pour pallier les défauts de l'approche objet, l'approche composant est apparue. Dans le monde Java, le composant CORBA est le plus utilisé. Il rend possible la distribution des tâches de développement à travers plusieurs programmeurs, et fait que, le système soit plus robuste et maintenable [Bieberstein 2008].

L'interopérabilité des plateformes n'était plus facile avec les modèles de composants utilisés ces dernières années. Cette interopérabilité s'avère encore plus difficile si le composant à invoquer est situé au-delà d'un pare-feu [Erl 2008].

La mise en œuvre de ces deux architectures soulève des difficultés dans le cadre d'une infrastructure ouverte telle qu'Internet. En effet, ces architectures proposent chacune sa propre infrastructure. Ce qui impose une forte liaison (un fort couplage) entre les services offerts par les composants et leurs clients. Ainsi, on ne peut assembler que des objets CORBA entre eux. Le résultat est que les systèmes construits à base de ces architectures sont homogènes.

Dans les années 2000, sont apparus les services Web pour pallier à tous ces problèmes, et en ne s'intéressant qu'à la manière d'interagir avec ce service Web sans connaître sa structure ou

sa technologie. Une application orientée services Web est simplement l'agrégation de services en une logique simple et en une application unifiée. Cependant, la clé de l'interopérabilité entre les services Web est l'utilisation des protocoles standards, des messages, et des contrats.

En outre, après l'avènement du B2C, où les entreprises mettaient en ligne leurs services pour leurs consommateurs à travers des applications Web, celles-ci souhaitaient accroître leurs productivités à l'aide du paradigme B2B. Le B2B repose sur l'échange de produits, d'informations et de services entre entreprises. Ceci implique l'utilisation de services et la collaboration avec des systèmes proposés par d'autres concepteurs, et par conséquent, une maîtrise de l'hétérogénéité [Papazoglou 2003].

L'interopérabilité est ainsi devenue une nécessité pour l'entreprise dans le monde du B2B. C'est justement ce que les services Web apportent par rapport aux solutions dites homogènes. D'une certaine façon, le modèle des services Web est une évolution du modèle des composants distribués. Cette évolution est rendue nécessaire par l'utilisation intensive d'Internet [Papazoglou 2003].

## **II.5 Caractéristiques de l'architecture SOA**

L'approche SOA confère aux entreprises plus de flexibilité dans leur activité en améliorant les applications et l'infrastructure informatique. Elle fournit un accès rapide à des informations plus précises et permet à l'entreprise d'identifier et de résoudre plus efficacement les problèmes de flux. Parmi ces caractéristiques, on peut citer les plus primordiales suivantes [Bejaoui 2008] [Kaabi 2008] [Albertella 2009] :

### **II.5.1 Couplage faible**

Le service est indépendant de son environnement tout en gardant des relations de collaborations. Cette notion d'indépendance se traduit, aujourd'hui, par le fait qu'un service doté d'une sécurité autonome, et il doit protéger ses fonctions et ses messages envoyés sans avoir besoin de connaître le degré de sécurité des clients, Le couplage faible signifie aussi que le client est indépendant de toute la technique adoptée par le service web, et cela inclut le system exploitation, Le langage de programme, et les bibliothèques utilisées.

### **II.5.2 Frontières explicites**

Les messages d'interaction avec un service web sont sous forme d'accord en XML pour garantir l'interopérabilité. Les messages permettent de produire des systèmes faiblement couplés qui touchent plusieurs systèmes d'exploitation.

### **II.5.3 indépendance par rapport à La plateforme**

La plateforme utilisée pour supporter une implémentation d'un service web ne doit pas être adéquate aux consommateurs. Ceci inclut les couches intermédiaires du système d'exploitation, du protocole de communication et même les couches de l'application.

### **II.5.4 Le langage de programmation**

Une SOA doit être implémentée indépendamment des spécifications des langages de programmation. Cependant, la pratique a révélé quelques problèmes d'interopérabilité entre les demandeurs et les fournisseurs de services en matière de représentation de types de données très complexes (tableaux, pointeurs nuls, . . . Etc.). Ils sont implémentés différemment dans différents systèmes et qui ont différents comportements en performances.

### **II.5.5 Le modèle d'un service**

On peut décrire les différents services et leurs relations chacun avec l'autre en termes d'un modèle, qui peut être sauvegardé, échangé et utilisé pour automatiser les interactions et faciliter la création d'une orchestration des services. Ceci nous permet d'utiliser dans plusieurs cas les mêmes services, pour créer des modèles d'interaction différents, et pour la construction de nouveaux domaines ou de nouvelles solutions.

Le paradigme orienté services représente une nouvelle tendance d'ingénierie logicielle. Il assure un développement d'applications plus rapide et à moindre coûts. SOA est une architecture fournissant une infrastructure nécessaire, pour intégrer les applications isolées afin de les utiliser an tant que services dans un réseau. La SOA est entrée depuis peu de temps dans le domaine du réel, grâce à un ensemble de normes appelées collectivement servicesWeb, ces derniers sont des réalisations concrète des architectures SOA, sont la déclinaison du paradigme des architectures orientées service, sur le Web.

Un service Web est toujours accompagné d'une description fournissant aux applications les informations nécessaires à son utilisation. En réalité, la philosophie des SOA

décompose une application homogène en une suite de services assurant la modularité dans leurs fonctionnalités [Brown 2008].

### III. Concept de service Web

Les services web sont alors apparus pour faciliter tout d'abord la tâche de développeur. Avant toute chose, Microsoft, contrairement aux idées reçues, n'a pas créé les services web mais Microsoft a participé avec des grands entreprises telles que IBM, SUN...à la standardisation des services web.

Ceci montre bien que la technologie des services web est une technologie très jeune, on peut dire qu'un service Web est souvent vu comme une application accessible à d'autres applications sur le Web, mais il existe plusieurs définitions pour les services Web :

#### III.1 Définition 1

Le consortium W3C définit un serviceWeb comme étant : « une application, ou un composant logiciel qui vérifie les propriétés suivantes » [Cerami 2002] :

- 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.

#### III.2 Définition 2

«Les services Web sont la nouvelle vague des applications Web. Ce sont des applications modulaires, auto-contenues et auto-descriptives qui peuvent être publiées, localisées et invoquées depuis le Web. Les services Web effectuent des actions allant de simples requêtes à des processus métiers complexes. Une fois qu'un service Web est déployé, d'autres applications (y compris des services Web) peuvent le découvrir et l'invoquer»

[Ponge 2008].

#### III.3 Exemple d'utilisation

Nous allons voir un cas d'utilisation de service web sans rentrer dans les détails. Dans cet exemple, prenons le cas simple d'une entreprise qui met en place un système de messagerie

au sein de son Intranet. Elle pourra donc opter pour la technologie des services web en développant le serveur de messagerie en service web .NET. L'avantage de cette solution est que l'application cliente pourra être développée sous n'importe quel langage. L'entreprise pourra même développer plusieurs clients certains marchent sous Unix, d'autres sous Windows mais utilisant le même serveur de messagerie, c'est-à-dire le WebService qui se charge de la transmission et de la réception des messages, nous pourrions donc très bien avoir un client Windows qui communique avec un client Unix [Nicolescu Matthieu 2003].

### **III.4 Caractéristiques des services Web**

Plusieurs acteurs définissent les services Web par des caractéristiques technologiques distinctives, qui sont [Cerami 2002] :

#### **III.4.1 Un service Web est une application logicielle qui est reconnue par un URI**

URI est la façon d'identifier un point de contenu sur le Web, que ce soit une page de texte, une vidéo, une image ou un programme. La forme la plus commune des URI est appelé Uniform Resource Locator (URL).

Le service web est donc accessible en spécifiant son URI, Par exemple cette URI: `http://www.w3.org/Icons/WWW/w3c_main.gif` identifie un fichier qui peut être consulté à l'aide de l'application du protocole Web, ("`http: //`") qui est logé sur un ordinateur nommé `www.w3.org`, Le fichier se trouve dans le chemin d'accès `"/Icons/WWW/w3c_main.gif"`.

#### **III.4.2 Capacité des interfaces et liaisons d'être publiées, localisées et invoquées via le langage XML**

Les principales tâches d'un service Web sont : La publication dans un registre, la localisation en interrogeant le registre qui l'héberge et l'invocation par un ou plusieurs services Web après sa localisation. Ces tâches sont réalisées via l'utilisation d'XML.

#### **III.4.3 Capacité d'interagir avec les composantes des logiciels via des éléments XML avec l'utilisation des protocoles Internet standards**

Un service Web est créé pour être utilisé et interagir avec d'autres logiciels contrairement à une page Web, ou à une autre application qui n'utilise pas les

services Web. C'est l'interopérabilité basée sur l'utilisation de l'XML et les protocoles Internet standards, par exemple, HTTP, SMTP, FTP, . . . etc.

#### **III.4.4 Composante logicielle légèrement couplée à interaction dynamique**

Un service Web ayant un programme qui permet de l'invoquer est appelé consommateur de service Web. Le service Web et son consommateur sont indépendants l'un de l'autre. Si une modification est à faire sur le consommateur, on n'a pas besoin de connaître la machine, le langage de programmation, le système d'exploitation ou autre paramètre, afin d'établir à nouveau une communication entre le service Web et son consommateur. Le consommateur possède une fonctionnalité qui correspond à faire une localisation et une invocation sur le service Web, au moment de l'exécution du programme de service Web de manière automatique.

### **III.5 Avantages des services Web**

L'idée essentielle derrière les services Web est de partager les applications et les programmes en un ensemble d'éléments réutilisables appelés service, de sorte que, chacun de ces éléments effectuent une tâche principale et efficace, afin de faciliter l'interopérabilité entre tous ces services Web [Chappell 2002].

Nous débutons par trois avantages décrit dans O'Reilley Network<sup>1</sup>

1. Les Web Services permettant à des portions de logiciels écrite en différents langages, ou évoluant sur différents systèmes d'exploitation, de communiquer entre eux et à peu de frais.
2. Les Web Services permettant à des applications supportant différentes processus d'organisation ou de différentes organisations, de communiquer, entre elles et/ou d'échanger des données facilement et à peu de frais.
3. Les web services utilisent des protocoles de données non propriétaires et universel afin que l'intégration entre deux nouveaux logiciels et les systèmes existant soit simple.[Clay Shirky,2012]

---

<sup>1</sup> Adaptation de <http://webservices.xml.com/pub/a/ws/2002/04/12/excreport.html>

### III.6 Architecture des services Web

Techniquement, un service Web peut donc être perçu comme étant une interface décrivant une collection d'opérations accessibles via le réseau, à travers des messages XML standardisés. D'un point de vue technique, la description d'un service Web inclut tous les détails nécessaires à l'interaction avec le service constituant, ce qu'on appelle l'architecture des services Web, par exemples, le format des messages, les signatures des opérations, le protocole de transport et la localisation du service Web [Kreger 2001].

L'interopérabilité est l'objectif premier des services Web. Pour permettre cet échange d'information entre des applications distantes, les services Web sont composés des couches standards.

L'architecture de services web appelé architecture de référence contient trois couches principale [Kreger 2001].

#### Architecture de référence

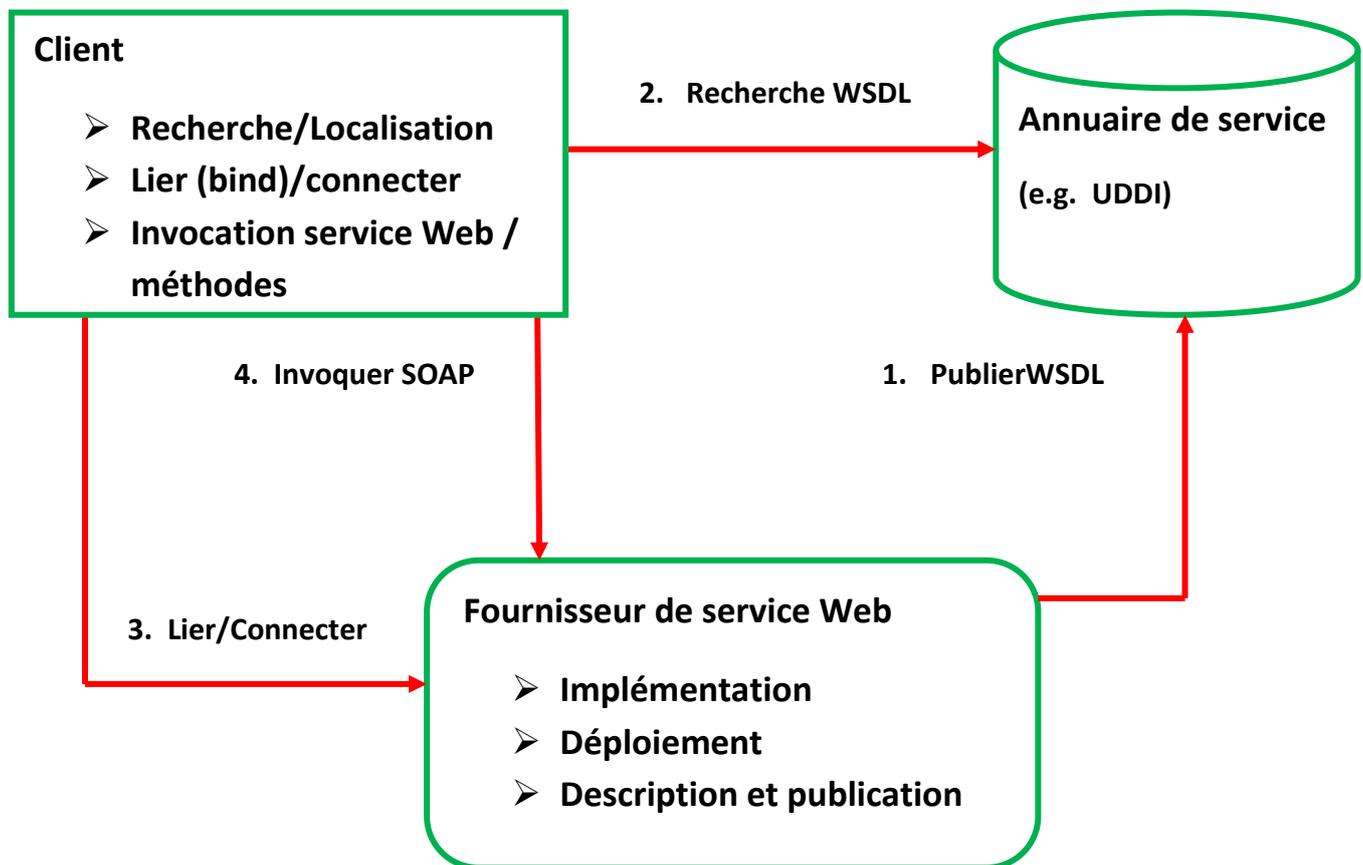
Cette architecture vise trois objectifs importants [Kreger 2001] :

- Identification des composants fonctionnels.
- Définition des relations entre ces composants.
- Etablissement d'un ensemble de contraintes sur chaque composant de manière à garantir les propriétés globales de l'architecture.

L'architecture de référence s'articule autour des trois rôles suivants (figure I. 1) :

- ✚ **Le fournisseur de service** : c'est le propriétaire du service Web. D'un point de vue technique, il est formé par la plateforme d'accueil du service Web, son rôle et la publication du service web à l'aide de formats standardisés (XML, WSDL-web service description language).
- ✚ **Le client** : c'est lui qui va demander un service Web. D'un point de vue technique, il est composé par l'application qui va rechercher et sélectionner un service Web. L'application cliente peut être elle-même un service Web.
- ✚ **L'annuaire des services** : Correspond à un registre de descriptions des services Web offrant des facilités de publication des services Web à l'intention des fournisseurs, ainsi que des facilités de recherche des services Web à l'intention des clients. Les interactions de base entre ces

trois rôles incluent les opérations de publication, de recherche et de liens d'opérations[Kreger 2001].



**Figure I. 1 Architecture de référence des services Web.**

### III.7 Les principales technologies standards autour de service Web

L'infrastructure des services web s'est concrétisée autour de trois spécifications considérées comme des standards, à savoir SOAP, WSDL (W3C) et UDDI (OASIS). Nous les détaillons dans la section suivante :

#### III.7.1 Le protocole SOAP

SOAP (W3C), veut dire « Simple Object Access Protocol » et la traduction de cette définition en français donnerait « Protocole Simple d'Accès aux Objets ». En effet, le protocole SOAP

consiste à faire circuler du XML via du HTTP sur le port 80. Cela facilite grandement les communications, car le XML est un langage standard et le port utilisé est le port 80 qui ne pose pas de problèmes pour les firewalls. SOAP peut donc être utilisé dans tous les styles

de communication: synchrone ou asynchrone, point à point ou multipoint, Intranet ou Internet [Kulchenko 2001].

Le SOAP fait partie de la couche de communication des services Web. La force de ce protocole réside dans son universalité et sa flexibilité. Il définit la structure des messages XML utilisés par les applications pour dialoguer entre elles. Par exemple, un client SOAP Java s'exécutant sur Linux, ou un client SOAP PERL s'exécutant sur Solaris peut se connecter à un serveur SOAP Microsoft s'exécutant sur Windows 2000. L'invocation d'une méthode d'un service Web depuis un client suppose la coopération de plusieurs couches logicielles.

La figure I.3 montre un exemple de message SOAP contient une requête et une réponse. Considérons la méthode d'un service simple qui double la valeur d'un entier donné.

#### Requête

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<SOAP-ENV:Envelope
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:doubleAnInteger
      xmlns:ns1="urn:MySoapServices">
      <param1 xsi:type="xsd:int">123</param1>
    </ns1:doubleAnInteger>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

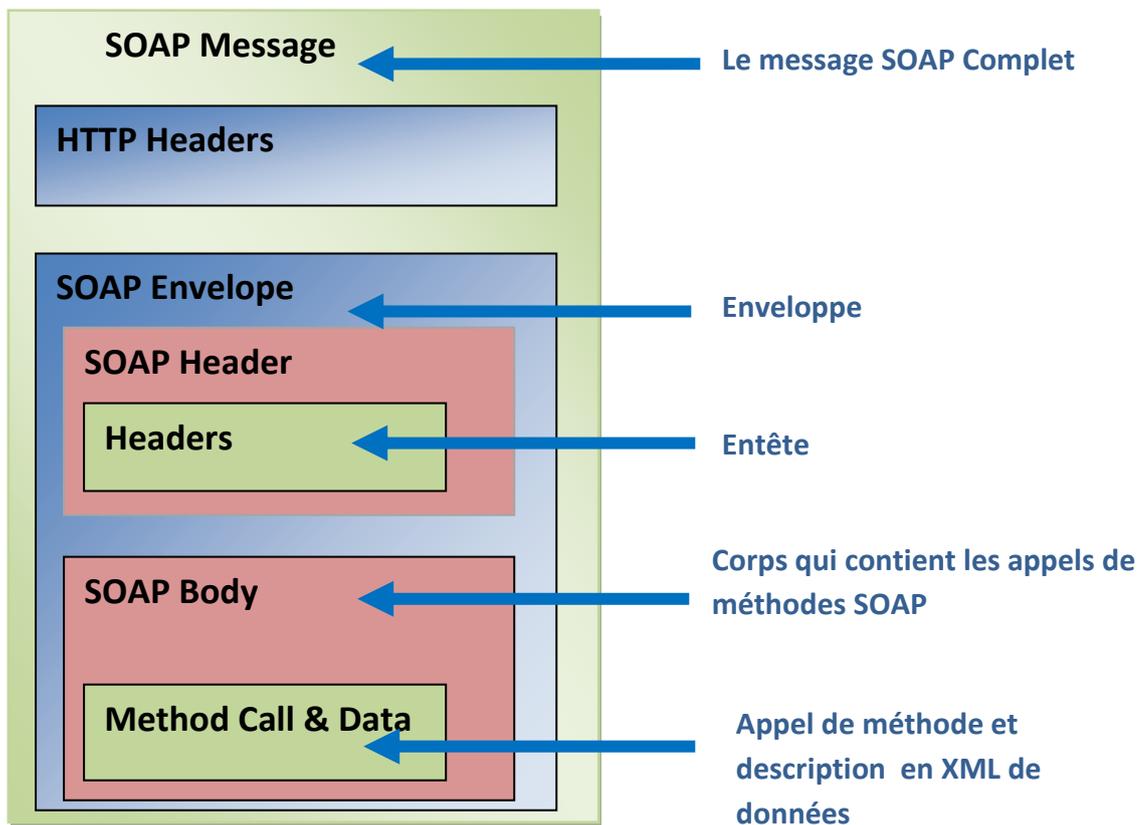
#### Réponses

```
<?xml version="1.0" encoding="UTF-8" ?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:doubleAnIntegerResponse
      xmlns:ns1="urn:MySoapServices"
      SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <return xsi:type="xsd:int">246</return>
    </ns1:doubleAnIntegerResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

**Figure I. 2: Le formatage visuel d'un message SOAP**

Le message est englobé dans une enveloppe et divisés en 2 parties : l'entête et le corps.

- L'entête (Header) : offre des mécanismes flexibles pour étendre un message SOAP sans aucune préalable connaissance des parties communicantes. Les extensions peuvent contenir des informations concernant l'authentification, la gestion des transactions, le paiement, etc.
- Le corps (Body) : offre un mécanisme simple d'échange des informations mandataires destinées au receveur du message SOAP. Cette partie contient les paramètres fonctionnels tels que le nom de l'opération à invoquer, les paramètres d'entrés et de sortis ou des rapports d'erreur [Ben Halima 2009].



**Figure I. 3 les composants d'un message SOAP**

### III.7.2 WSDL

WSDL a été développé conjointement par IBM, Microsoft et Ariba et a été présenté pour analyse au W3C qui l'a accepté comme une notice et publié sur leur site. L'utilité de WSDL est de décrire et publier le format et les protocoles d'un Web service

de manière homogène par l'utilisation du format XML. Cela permettra au requérant et à l'émetteur d'un service de comprendre les données qui seront échangées. WSDL est simplement un IDL (Interface Définition Langage) tout comme Corba IDL ou les interfaces de Java ou de COM par exemple. WSDL est différent de ceux-ci en ce sens qu'il est tout aussi neutre du point de vue du protocole que du point de vue de l'implémentation. Cependant, malgré que WSDL soit neutre du point de vue du protocole, la majorité des implantations WSDL<sup>2</sup> se font à partir de SOAP puisque les promoteurs de SOAP<sup>3</sup> sont aussi les promoteurs de WSDL. Toutefois, des liaisons pour http, MIME et SMTP sont aussi déjà disponibles. De plus, WSL pour être lié à différents autres protocoles tel que BEEP (Block Extensible Exchange Protocol de l'IETF (Internet Engineering Task Force)), DCOM ou IIOP (Internet Inter-Orb Protocol)<sup>4</sup>.

### III.7.3 L'annuaire UDDI

UDDI (OASIS), a été conçu en 2000 à l'initiative d'un ensemble d'industriels (Ariba, IBM, Microsoft), en vue de devenir le registre standard de la technologie des services Web. Pour convenir à la technologie des services Web, les services référencés dans UDDI sont accessibles par l'intermédiaire du protocole de communication SOAP, et la publication des informations concernant les fournisseurs et les services doit être spécifiée en XML afin que la recherche et l'utilisation soient faites de manière dynamique et automatique. UDDI constituent méta-service possédant des fonctions de publication et de recherche [Lopez-Velasco 2008].

En clair, l'UDDI gère l'information relative à la publication, la découverte et l'utilisation d'un service Web. Ce standard définit donc, un registre des services Web sous un format XML. Les organisations publient les informations décrivant leurs services Web dans l'annuaire, et l'application client ayant besoin d'un certain service, consulte cet annuaire pour la recherche des informations concernant le service Web qui fournit le service désiré, pour une éventuelle interaction, ainsi l'UDDI a été créé pour faciliter la découverte de services Web en plus de leurs publications. Par une API SOAP, on peut interagir avec l'UDDI au moment de la conception et l'exécution des applications afin de découvrir des données techniques, et administratives sur les entreprises et leurs services Web. L'annuaire

<sup>2</sup> Web Services, [http://www.omg.org/news/meetings/workshop/webservices\\_2002.htm](http://www.omg.org/news/meetings/workshop/webservices_2002.htm) Mark Perreira, Chief Scientist, Talking Blocks Contracts for Services: Needs and Nonsense!

<sup>3</sup> William L. Oellermann, Jr., Architect of web Services, éd. Apress, 2001, p. 606

<sup>4</sup> Understanding web Services : XML, WSDL, SOAP and UDDI , éd. Addison-Wesley, 2002, p. 27

UDDI repose sur le protocole SOAP, les requêtes et les réponses sont des messages SOAP [Scott 2002] [Gardien 2002].

L'UDDI est subdivisé en deux parties principales : partie publication ou inscription, et partie découverte.

La partie publication regroupe l'ensemble des informations relatives aux entreprises et à leurs services. Ces informations sont introduites via une API d'enregistrement. La partie découverte facilite la recherche d'information contenue dans UDDI grâce à l'API SOAP [Newcomere 2004].

La structure des données UDDI est exprimée en utilisant les schémas XML. Ils sont divisés en 4 types de structures de données :

- **Business Entity** : C'est l'équivalent des pages blanches de l'annuaire. Cette structure inclut donc les informations concernant l'entreprise ou l'entité qui publie son Web service. Cette structure peut inclure un BusinessKey (clé d'affaire) qui peut prendre la forme d'une clé UUID<sup>5</sup> (Universally unique identifiers) assurant la protection de l'information contenu dans le Web service.
- **BusinessService** : C'est l'équivalent des pages jaunes de l'annuaire. On y retrouve les informations concernant le nom et la description du Web service Offert.
- **BindingTemplate** : C'est l'équivalent des pages verts citées plus haut. Il contient les informations concernant les points d'accès au Web service et les aspects techniques permettant la liaison entre le Web service et l'API du requérant de celui-ci. Il fait référence à un ou plusieurs tModel.
- **tModel** : Il s'agit du mécanisme permettant l'échange de métadonnée à propos du Web service. Il peut s'agir d'un pointeur à un fichier WSDL, mais il peut aussi pointer d'autres types de fichiers comme sur des fichiers sbXML ou Roset Janet par exemple. Mais chaque tModel ne définira qu'un type d'interface avec laquelle elle peut interagir. Cependant un fichier UDDI peut contenir plusieurs tModel.

---

<sup>5</sup>DCE Glossary-What is a UUID ?.<http://www.dsps.net//uuid.html> et DCE 1.1:Remote Procedure Call – Universal Unique Identifier , <http://www.opengroup.org/onliepubs/9629399/apdxa.htm>

## IV. Recherche et découverte des services web

### IV.1 Introduction

La découverte de services web désigne le processus par lequel les futurs utilisateurs ou consommateurs d'un service recherchent manuellement ou semi automatiquement le service correspondant à leur besoin.

Les services web auront été préalablement créés et publiés dans des annuaires de services web comme document UDDI ou Web Services Inspection Language (WSIL).

L'implémentation de serveurs UDDI et de moteurs WSIL devrait fournir des recherches simples d'API (application programming interface) ou de "web-based GUI" pour aider à trouver ces services web.

### IV.2 Définition

#### Définition 1

« la découverte est la localisation automatique des services répondant à une requête utilisateur ». [Keller et al, 2004].

#### Définition2

le processus de découverte est la localisation d'une description compréhensible par la machine d'un service éventuellement inconnu au préalable et correspondant à certains critères fonctionnels. [Booth et al, 2004]

Plusieurs critères peuvent être utilisés pour catégoriser les approches de découverte nous avons :

- Le critère d'automatisation.
- Le critère de centralisation/distribution des annuaires.
- Le principe de l'algorithme de matching (syntaxique, sémantique (logique, non logique), hybride, comportemental, non fonctionnel...)

### **IV.3 Critères de découvert**

#### **IV.3.1 Le critère d'automatisation**

Ce critère mesure le pourcentage d'intervention de l'utilisateur lors de la phase de découverte, et on peut distinguer deux approches principaux : l'approche manuelle et l'autre automatique.

#### **IV.3.2 Le critère d'architecture (Centralisation/distribution)**

Ce critère désigne comment le stockage et la localisation des descriptions de services dans le réseau. Selon cette idée, l'architecture peut être décomposée en deux type : centralisée et décentralisée.

#### **IV.3.3 Le critère de Matchmaking**

Ce critère s'intéresse à l'algorithme de description de la requête de client avec la description du service, il y a deux approches qu'on peut les distinguées, les approches fonctionnelles et les approches non fonctionnelles, la première approche c'est elle qu'on va s'intéressé dans notre travail.

### **IV.4 Les approches**

Les services considéré comme une fonction peuvent inclure les entrées et les sorties, la catégorie de service, le comportement. On distingue plusieurs types concernant cette approche, les approches syntaxiques et les approches sémantique.

#### **IV.4.1 Approches syntaxique**

Le principe de cette approche basée sur la comparaison de mots clé de la requête des utilisateurs avec les attributs enregistré. Cette approche [Algergawy et al, 2010] utilise deux algorithme: le matching basé sur les niveau, et le matching des schémas. Le premier compare syntaxiquement les éléments concrets de WSDL, et le deuxième compare les éléments abstraits du document WSDL. Parmi les approches qu'on peut les citer :

##### **a) Approche UDDI**

L'approche UDDI est basée sur l'utilisation d'un registre de description de services web. La publication et la découverte des services web basée sur la comparaison des mots clés.

### b) Approche basé sur les qualités de service

Cette approche prend la qualité de service comme contrainte pendant la recherche d'un web service, elle comporte quatre élément:

- Le fournisseur de service : offre le service en l'enregistrant dans l'annuaire UDDI.
- Le consommateur de service : c'est lui qui découvre et invoque les services.
- L'annuaire UDDI : c'est l'annuaire doté d'informations sur la description fonctionnelle du web service et des informations sur les Qds associés a ce service.
- Et le certificateur :son rôle et de vérifier les revendications de qualité de service d'un web service avant son enregistrement.[**ADDOUR Drifa 2013**]

### IV.4.2 Approche sémantique

Ces approches fondées sur des langages particuliers d'ontologies du web. Ceci leur permet de décrire les services web de façons non ambiguë et interprétable par des programmes. Cette famille comprend OWLS et WSMO qui représente les modèles sémantique les plus utilisées dans la littérature [**Ibrahim el Bitar 2014**], Cette catégorie assure une meilleur interprétation et par la suite garantir une découverte efficace en améliorant la qualité des résultats obtenus , donc ces approches sont plus complexes et plus fiable que les technique syntaxique, cette approche peuvent être regroupées en trois classes :l'approche algébrique, l'approche déductive et l'approche hybride. L'approche algébrique exploite généralement la théorie des graphes et procède par calcul de distance, tandis que l'approche de découverte déductive est fondée essentiellement sur la logique. L'approche qualifiée d'hybride, quant à elle, est celle qui combine les mécanismes de ces deux classes.

#### ➤ Approche algébrique

L'approche de découverte algébrique trouve ses origines dans l'algèbre et les mécanismes de recherche d'information. Elle est fondée sur le calcul du degré de similarité textuelle à partir de graphes structurés construits à cet effet, ou encore sur le calcul de distance (du chemin) entre les concepts appariés. Cette approche que nous qualifions d'algébrique utilise des mécanismes de matching structurel, numérique et syntaxique à travers un appariement de graphes structurés et en calculant des distances numériques pour vérifier la similitude syntaxique. Pour exploiter la sémantique, ces mécanismes d'appariement utilisent les fréquences des termes et les sous-graphes. **iMatcher1** [**Schumacher et al. 2008**], Agent Approach for Service Discovery and Utilisation (AASDU)[**Palathingal et Chandra, 2004**] et DSD-Matchmaker

[Klein et König-ries, 2004a] sont des plateformes et travaux qui adoptent cette catégorie d'approches.[Ibrahim el Bitar 2014]

➤ **Approche déductive**

L'approche déductive est fondée sur la logique. Les travaux optant pour cette catégorie d'approches utilisent des descriptions de services et des requêtes spécifiées en langages issus de formalismes logiques, telles que la logique de description et la logique de premier ordre. Ils utilisent également des règles logiques pour la découverte de services Web et exploitent les ontologies pour couvrir leur aspect sémantique. Pour calculer le degré d'appariement, ils recourent à différentes façons et ciblent plusieurs éléments de description de services tout en prenant en considération leur sémantique. Ils optent essentiellement pour trois types d'appariement : IO-matching (Inputs and Outputs matching) [Srinivasan et al., 2004] [Fan et al., 2005] [Paolucci et al., 2002], PE-matching (pré conditions and effects matching) [Schumacher et al., 2008] et IOPE-matching (Inputs, Outputs, pré conditions and effects matching) [Jaeger et al., 2005] [Keller et al., 2005][Stollberg et al., 2007] [Küster et König-Ries, 2008]. Dans l'IO-matching, les éléments objet d'appariement se limitent aux inputs et outputs. PE-matching prend les pré-conditions et effets comme éléments d'appariement. Quant au cas du IOPE-matching, les concepts sémantiques décrivant les inputs, outputs, pré conditions et effets font à la fois l'objet de l'appariement des services et des requêtes. [Ibrahim el Bitar 2014]

➤ **Approche Hybride**

L'approche hybride utilise des mécanismes déductifs intégrant les méthodes de calcul de distances. L'idée est de remédier aux limites de chacun de ces deux mécanismes en les combinant. Plusieurs travaux [Rey, 2002] [Kiefer et al., 2007] [Kiefer et Bernstein, 2008] [Bertoli, 2004] [Kaufer et Klusch, 2006] [Klusch et Kapahnke, 2008] optent pour cette approche, citon-en :

- a. **OWLS-MX** : OWLS-MX [Klusch et al, 2006] est un matchmaker sémantique hybride qui réalise un IO-matching entre des profils de services OWL-S<sup>6</sup>.
- b. **OWLS-iMatcher2** :OWLS-iMatcher2 [Kiefer et al., 2007] [Kiefer, 2009] [Kiefer et Bernstein, 2008] est une approche hybride de découverte qui opte à la

---

<sup>6</sup> OWL-S (Ontology Web Language for Service) est une ontologie qui s'inscrit dans le cadre de l'approche à base de langage sémantique

fois pour un appariement déductif des concepts d'inputs/outputs et un appariement algébrique utilisé pour le calcul de la similarité textuelle des noms et des signatures des services [Bernstein et al., 2005] [Bernstein et Kiefer., 2005].

- c. **WSMO-FX** : L'approche WSMO-MX [Kaufer et Klusch, 2006], quant à elle, est une approche hybride d'appariement de services décrits en WSML-MX [Klusch et al., 2008]. Elle opte pour un appariement sémantique [Klusch et al., 2006] et syntaxique et procède par des techniques d'appariement de graphes orienté objet héritées du DSD-Matchmaker [Klein et König-ries, 2004] et d'appariement intentionnel de services [Keller et al., 2005].
- d. **SAWSDL-MX** : SAWSDL-MX [Klusch et Kapahnke, 2008] est un matchmaker inspiré des matchmakers OWLS-MX [Klusch et al., 2006] et WSMO-MX [Kaufer et Klusch, 2006] qui permet de découvrir des services décrits en SAWSDL [SAWSDL, 2011]. Il opte à la fois pour un appariement logique fondé sur le raisonnement par subsomption et un appariement syntaxique fondé sur des techniques de recherche d'information.

### Les mesures de similarités

Dans tous les domaines de l'informatique dans lesquels on désire analyser de manière automatique un ensemble de données, il est nécessaire de disposer d'un opérateur capable d'évaluer précisément les ressemblances ou les dissemblances qui existent au sein de ces données. Sur cette base, il devient alors possible d'ordonner les éléments d'ensemble, de les hiérarchiser ou encore d'en extraire des invariants. Pour qualifier cet opérateur, nous utiliserons dans ce chapitre le terme de fonction de similarité ou plus simplement celui de mesure de similarité.

Parmi les mesures de similarité citons-en:

#### 1. Extended Jaccard

La mesure de similarité de Jaccard est définie par le nombre des objets communs divisé par le nombre total des objets moins le nombre d'objets communs :

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

## 2. Cosinus

Cette mesure utilise la représentation vectorielle complète, c'est-à-dire la fréquence des objets (mots). Deux objets (documents) sont similaires si leurs vecteurs sont confondus. Si deux objets ne sont pas similaires, leurs vecteurs forment un angle (X, Y) dont le cosinus représente la valeur de la similarité. La formule est définie par le rapport du produit scalaire des vecteurs x et y et le produit de la norme de x et de y l'équation est comme suit :

$$\text{Cos}(\theta) = \frac{\sum_{i=0}^n A_i \times B_i}{\sqrt{\sum_{i=0}^n (A_i)^2} \times \sqrt{\sum_{i=0}^n (B_i)^2}}$$

## 3. Jonson Shannon

C'est une approche basée sur la divergence, la similitude est donnée comme suit :

$$\text{JSD}_{\pi_1, \dots, \pi_n}(P_1, P_2, \dots, P_n) = H\left(\sum_{i=1}^n \pi_i p_i\right) = \sum_{i=1}^n \pi_i H(p_i)$$

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

## 4. Loss Information

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

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

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

**Soit :** {  
    **OUT** : Les données de sortie  
    **IN** : les données de sortie  
    **R** : La requête  
    **S** : Le service

## 5. Logique

elle donne 5 score :

- exact (service = requête).
- plug-in ( service est. un classed folly de la requête ).
- subsume ( service cas particulier de la Requête mais indirect ).
- subsumed by (la requête est direct par rapport au service).
- fail ( aucun lien entre le service et la requête ).

## IV. Conclusion

La technologie des services Web offre de fortes potentialités pour surmonter les problèmes d'interopérabilité des systèmes. Elle constitue un cadre prometteur pour l'intégration des applications, et pour la gestion des interactions entre divers partenaires dans un environnement distribués, hétérogènes, ouvert et versatile qui est le Web. Requête

**Chapitre 2 :**

**Conception et**

**Implémentation du**

**prototype**

## I. Introduction

L'objectif de cette partie se résume dans la présentation d'une approche de découverte de services web en se basant sur la probabilité à fin de fusionner tous les algorithmes de matching, pour cela nous décrivons en premier temps le prototype implémentant l'environnement de développement, Ensuite nous montrons les différentes expérimentations menés, puis discuter les résultats obtenus et les comparer avec d'autres approches récemment traitées,

## II. Présentation du Corpus

Dans cette phase nous décrivons le corpus dont nous avons utilisée pour évaluer notre approche, pour cela on a utilisé un corpus open source OWLS-TC version 2.2.114 qui était développé par le centre allemand pour la recherche en intelligence artificielle (<http://www.dfki.de/scallops>). Ce corpus décrit un ensemble de services web a travers des document OWLS, elle dispose de 1007 services web réparties en 7 classes :

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

La base propose aussi un ensemble de requêtes réparties sur les 07 classes, ces requêtes sont modélisées sous forme de document OWLS. Chaque document OWLS (service ou requête) comporte dans sa partie « profile » des éléments « profile :hasinput » et « profile:hasoutput », ces derniers sont employés comme entrées pour le module de découverte des services web. Chaque service web (i.e. document OWLS) est étiqueté manuellement par des experts humains comme étant relevant ou non par rapport à une requête donnée. En d'autre termes, chaque service web possède une étiquette binaire (relevant ou non) par rapport à une requête donnée. Ceci permet le calcul des rappels et des précisions des approches proposées. La base offre aussi un ensemble d'ontologies pour décrire les services et les requêtes, chaque classe de services possède une ou plusieurs ontologies. Dans ce qui suit, nous donnons un exemple d'une description de service

web et d'une description de requête à l'aide des éléments « profile : hasinput » et « profile : hasoutput ».

- Requête

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

Concepts d'entrée : Hospital

Concepts de sortie: Investigating

- Service

Ce service donne les prix des véhicules ayant trois roues.

Nom de service : 3wheeledcar\_price

Concepts d'entrée : 3wheeledcar

Concepts de sortie : price

### III. La Conception

#### III.1 Définition de l'Approche « Dominance Probabiliste « DP »

Dominance Probabiliste et dite aussi « probabilistic dominance approach » est composé de deux mots la Dominance et la probabilité.

La relation de dominance est une relation binaire, si on considère A et B deux élément on dit que A domine B si est seulement si A est plus grand est plus fort et plus pertinent par rapport à B.

Le calcul de la probabilité désigne l'utilisation de raisonnements probabilistes dans la résolution de problèmes purement déterministes.

On a appliqué cette approche pour la résolution de notre problématique, pour que cette tâche soit accomplie on a considéré chaque service web de notre corpus comme un vecteur qui contient au plus cinq sous vecteurs, chacun de ces derniers présente le score de similarité entre la requête et le service qui est calculé avec un des algorithmes de matching. On donne la fonction de probabilité de dominance entre deux services U et V sous la forme suivante :

$$Domin - Proba (U, V) = \frac{1}{|U|} \sum_{i=1}^m \left( \frac{1}{|V|} \sum_{j=1}^m |u > v| \right)$$

Tel que :

$u > v$ : veut dire que U domine V avec :  $u \in U ; v \in V$

m : le nombre de mesures de similarités

Par conséquent on calcule la Probabilité Global de chaque service par rapport au tous les autres services de la base avec la Formule suivante :

$$Proba - Domin - global(U) = \frac{1}{|base - 1|} \cdot \sum_{S \neq U} Domin. Proba(U, S)$$

Avec :  $S \in base$

### III.2 Exemple d'application

Etant donnés deux services A et B on affecte pour chaque service des scores de similarité

A {(0.8 ; 0.9), (0.75 ; 0.95), (0.74 ; 0.6)}

B {(0.8 ; 0.65), (0 ; 0.92)}

Donc on applique la formule Domin-Proba (A,B) :

- $\frac{1}{|A|} = 1/3, \frac{1}{|B|} = 1/2$
  - Domin-Proba (A,B) =  $1/3 * (1/2 * 1 + 1/2 * 1 + 0) = 1/3 = 0,33$
  - Domin-Proba (B, A) =  $1/2 * (0 + 0) = 0$
  - Proba-Domin-Global(A) = 0,33
  - Proba-Domin-Globa(B) = 0
- Donc on peut dire que A est meilleur que B parce que le score de A est plus grand que le score de B.

## III.3 Organigramme

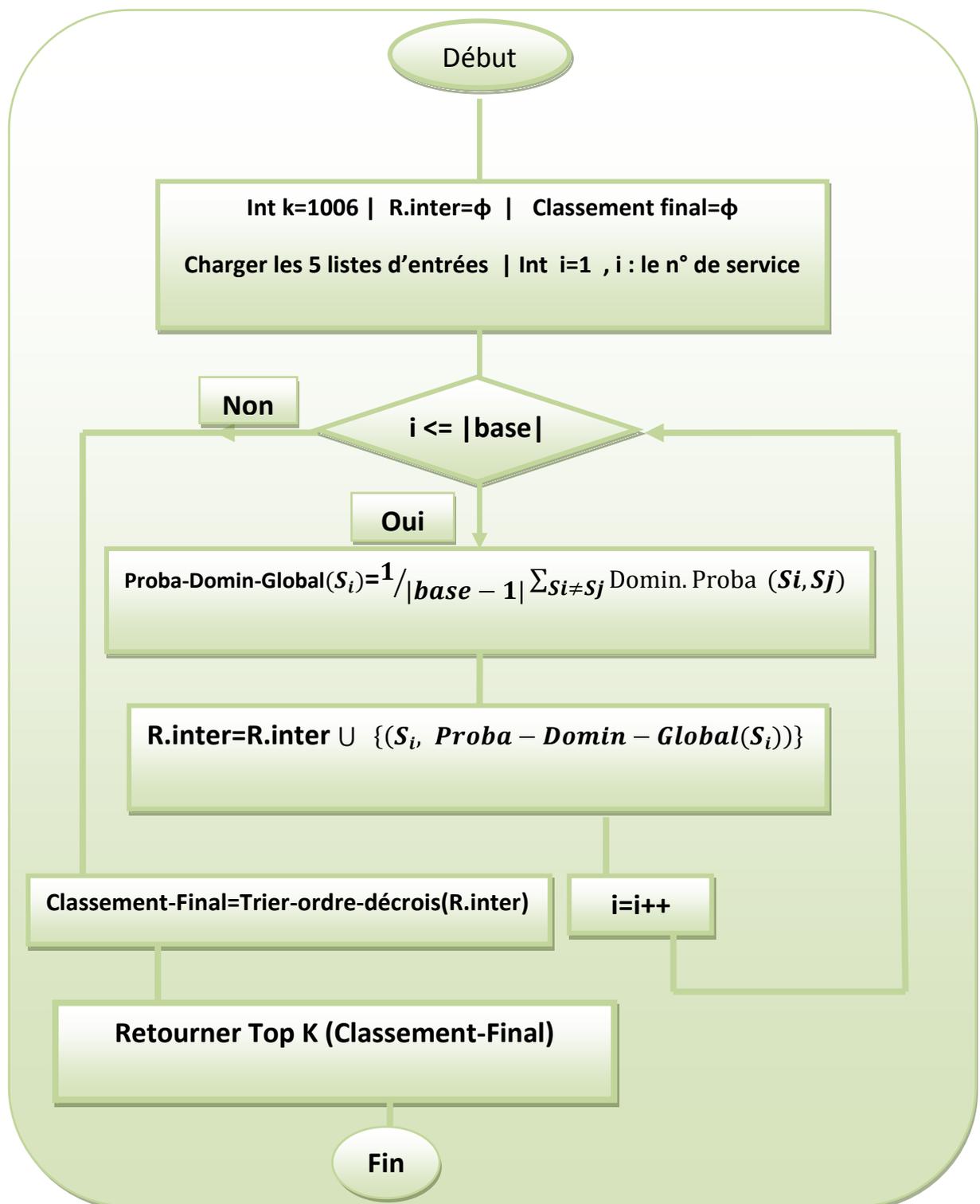


Figure II. 1 : Organigramme de Dominance Probabiliste

### III.4 Déroulement de l'algorithme

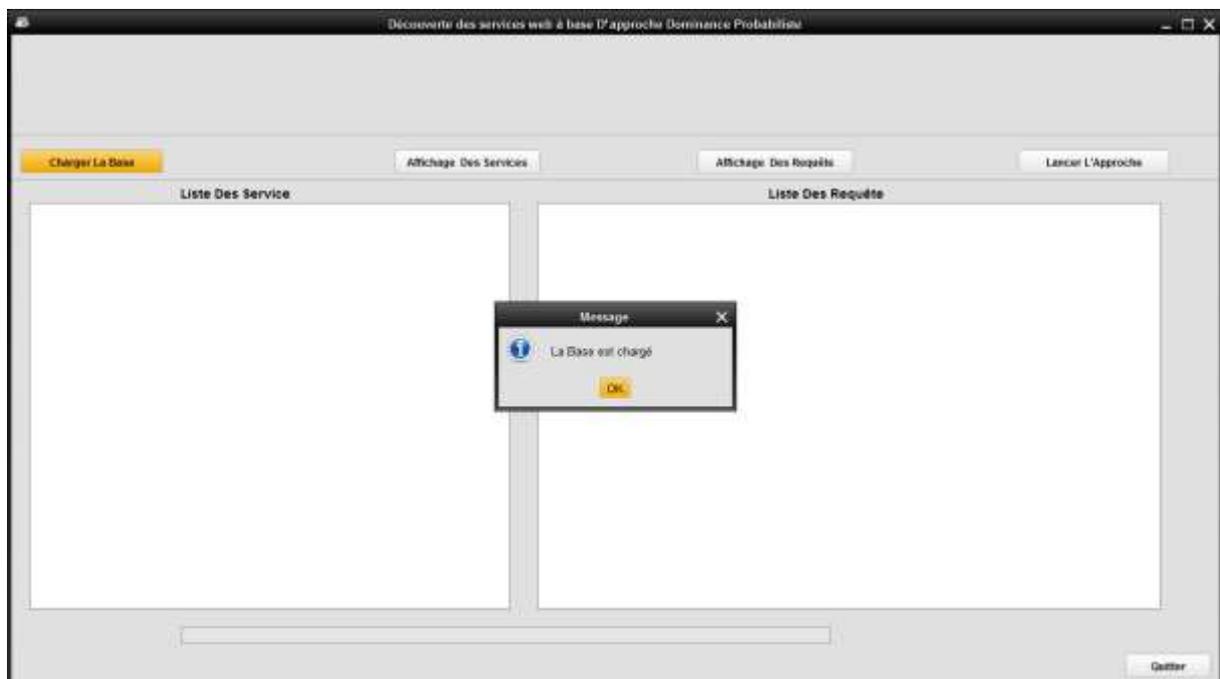
- On initialise les paramètres de l'algorithme et on charge la base des services ainsi que les scores individuels.
- Boucle Principal :
  - ✓ On calcule le score du service  $S_i$  en appliquant la Formule de la Dominance Global :
 
$$\text{« Proba-Domin-Global}(S_i) = \frac{1}{|base - 1| \sum_{S_i \neq S_j} \text{Domin} - \text{Proba}(S_i, S_j) \text{ »}}$$
 Et la « Domin – Proba( $S_i, S_j$ ) =  $\frac{1}{|S_i|} \sum_{i=1}^m (\frac{1}{|S_j|} \sum_{j=1}^m |u > v|)$  »
   
avec :  $u \in S_i$  et  $v \in S_j$ 
  
plus le score est grand plus le rang du service est meilleur
  - ✓ On ajoute le service  $S_i$  avec son score dans un résultat intermédiaire « R.inter »
  - ✓ Incrémentation de l'index du service.
  - ✓ Trier le résultat inter par ordre décroissante dans des scores de probabilité.
  - ✓ Retourne les TopK service dans une liste classée.

## IV. Présentation du prototype

Nous allons présenter les différentes étapes de l'application, bien précisément les interfaces homme machine :



**Figure II. 2 : Interface d'accueil**



**Figure II. 3 Chargement de la base**

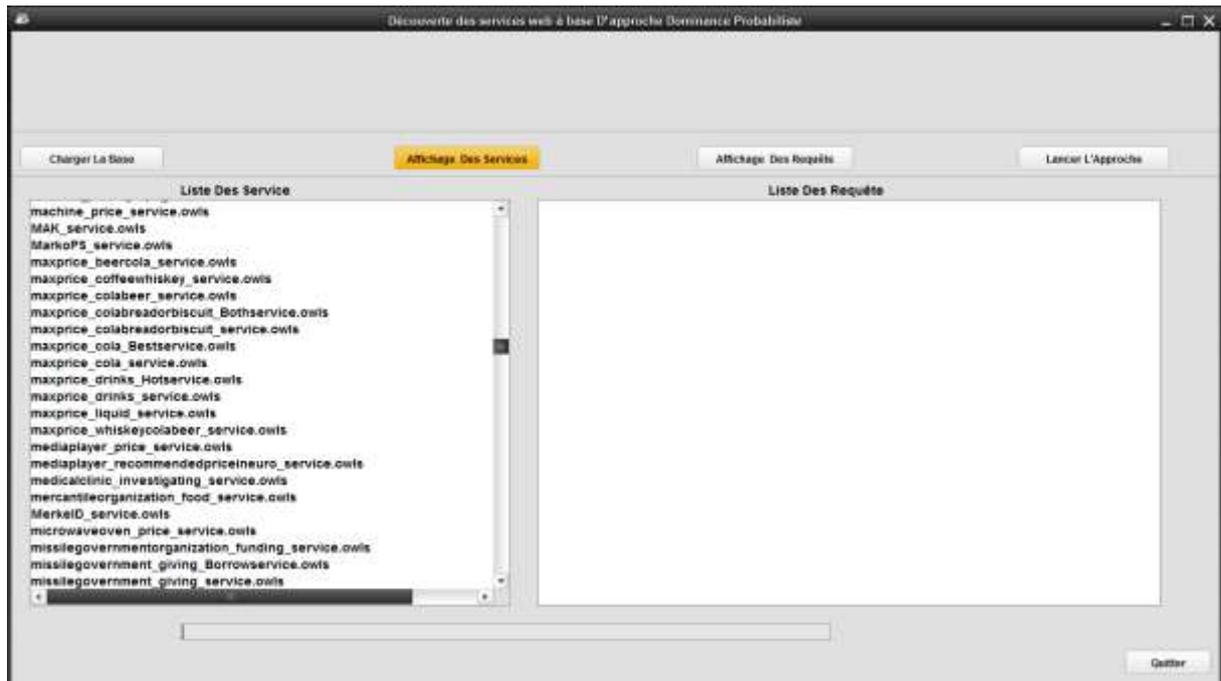


Figure II. 4 Affichage des services web

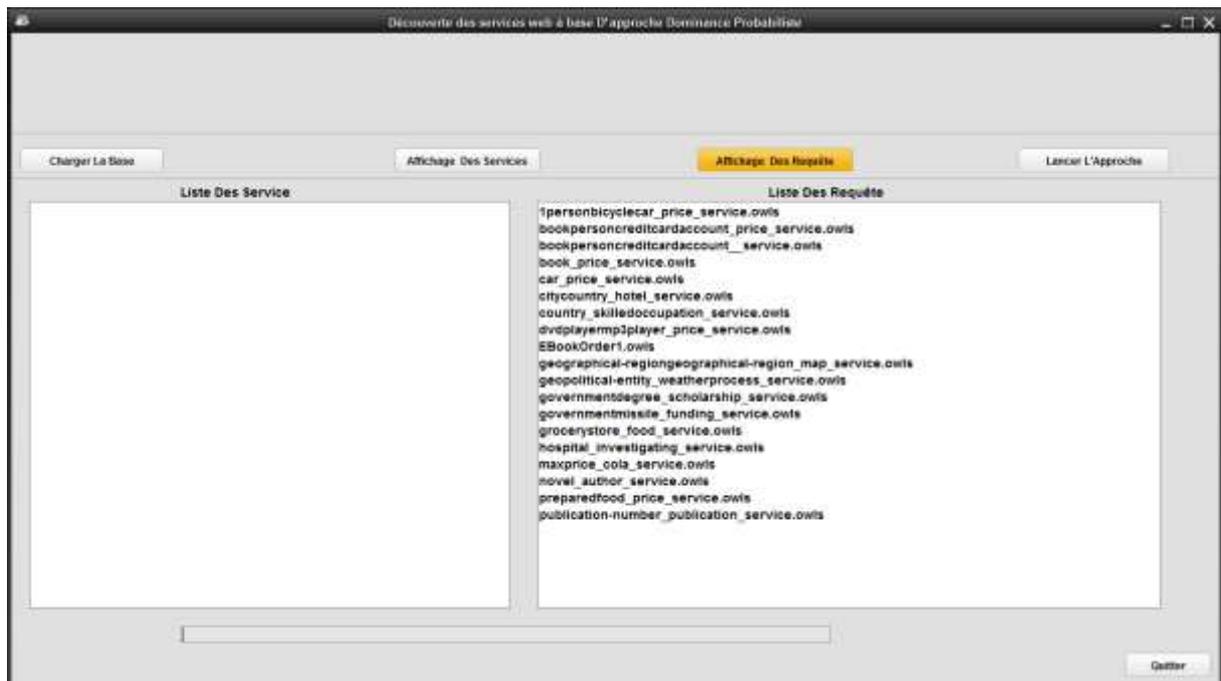


Figure II. 5 Affichage des requêtes

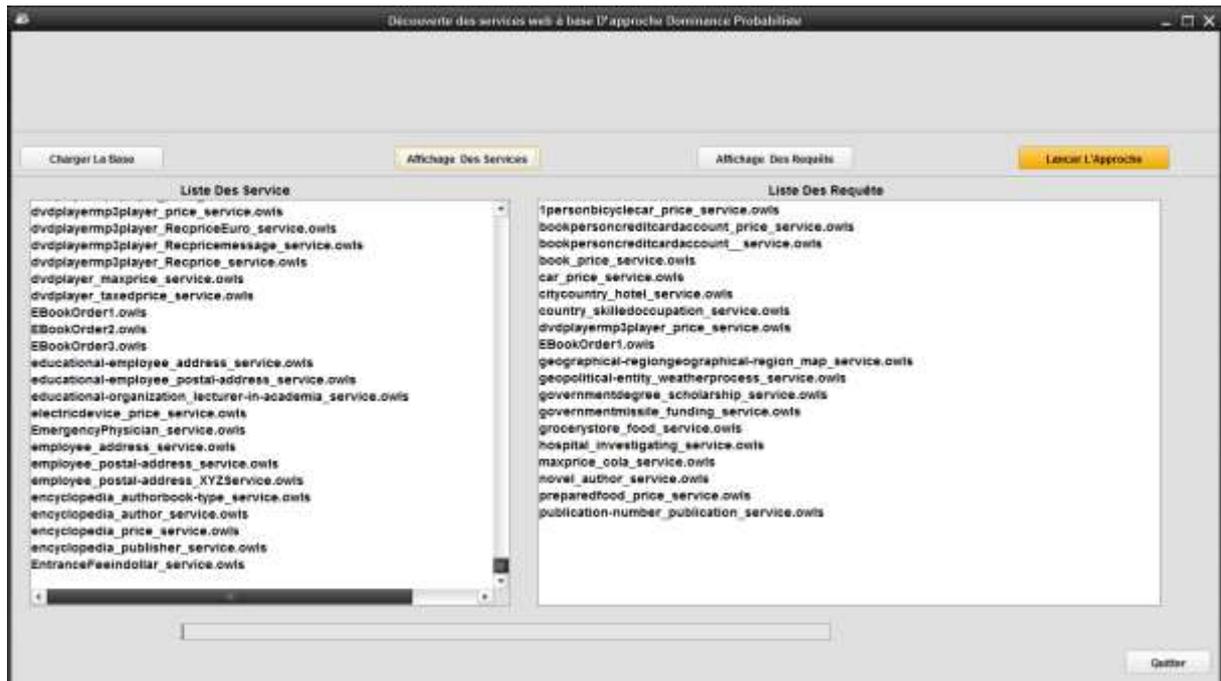


Figure II. 6 le Lancement de l’algorithme



Figure II. 7 interface du choix de la requête et l’exécution

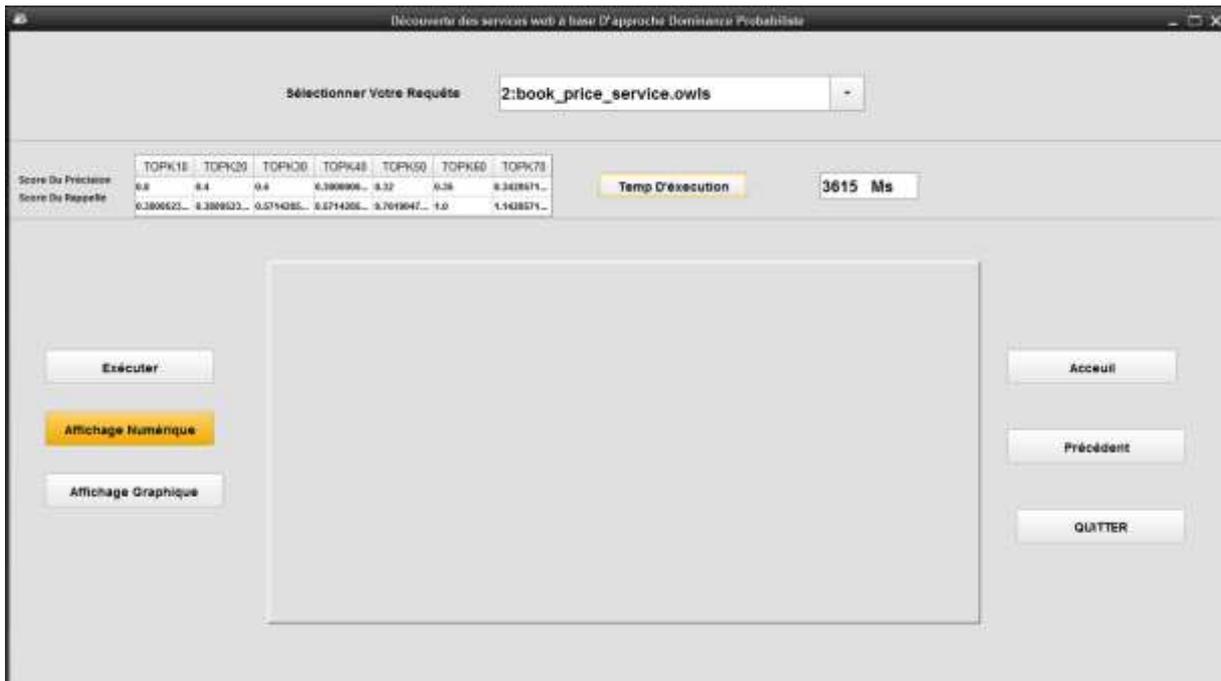


Figure II. 8 affichage numérique des résultats et du temps d'exécution



Figure II. 9 Affichage Graphique des résultats

### Définition des paramètres

La précision P : c'est le rapport entre les réponses correctes fournies par le système et le nombre total des réponses, Et plus formellement :  $Précision = \frac{vp}{(vp+fp)}$ . 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).

Le rappel R : c'est le rapport entre les réponses correctes fournies par le système et le nombre réel des réponses correctes appartenant au corpus, et plus formellement :  $Rappel = \frac{vp}{vp+fn}$ . Avec :

fn : les faux négatifs. (Un résultat correct, et considéré comme faux par le système).

Top k : Représente le nombre des services choisi pour comparer les services pertinents triés avec les services relevant 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.

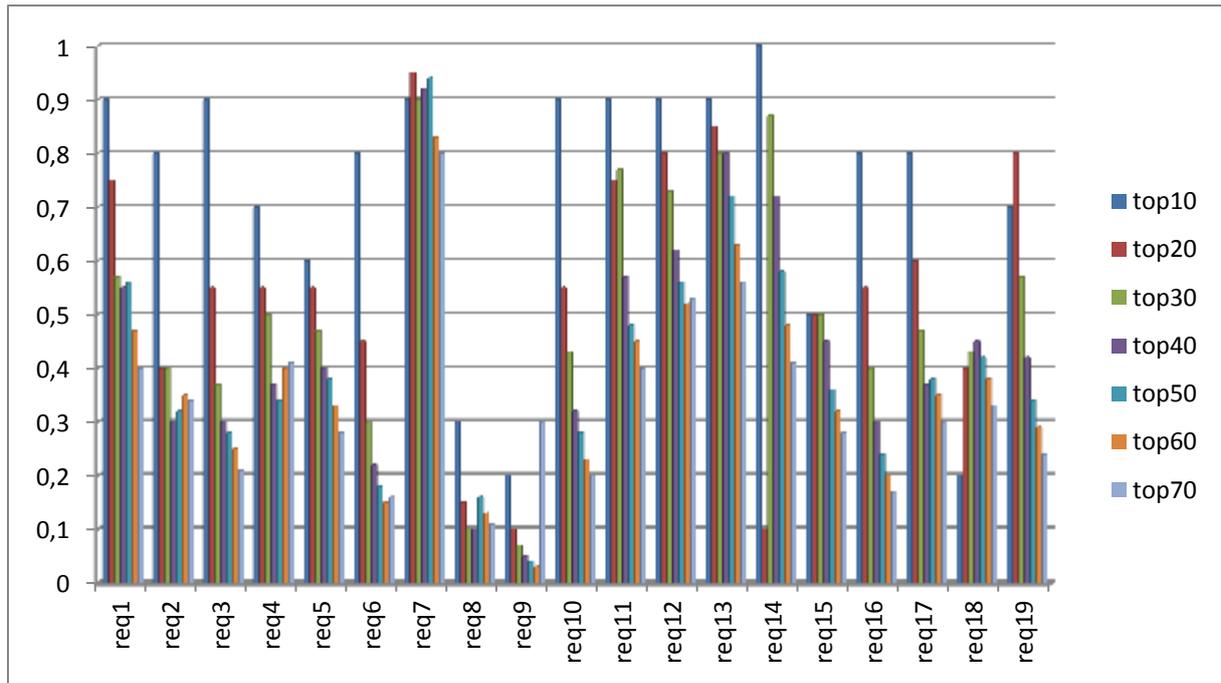
## V. Expérimentation

### Environnement de développement et outils

Dans cette section, nous décrivons les expériences permettant l'analyse des performances des approches proposées.

Ces expérimentations sont menées sur une plateforme, ayant processeur Corei3 avec 04 GO de RAM et sous le système d'exploitation Windows8. Le système est développé avec java (jdk 1.7.0) et sous l'environnement Netbeans 6.9.1 et le Microsoft Excel 2007 pour la représentation des résultats sous forme de tableau et d'Histogramme.

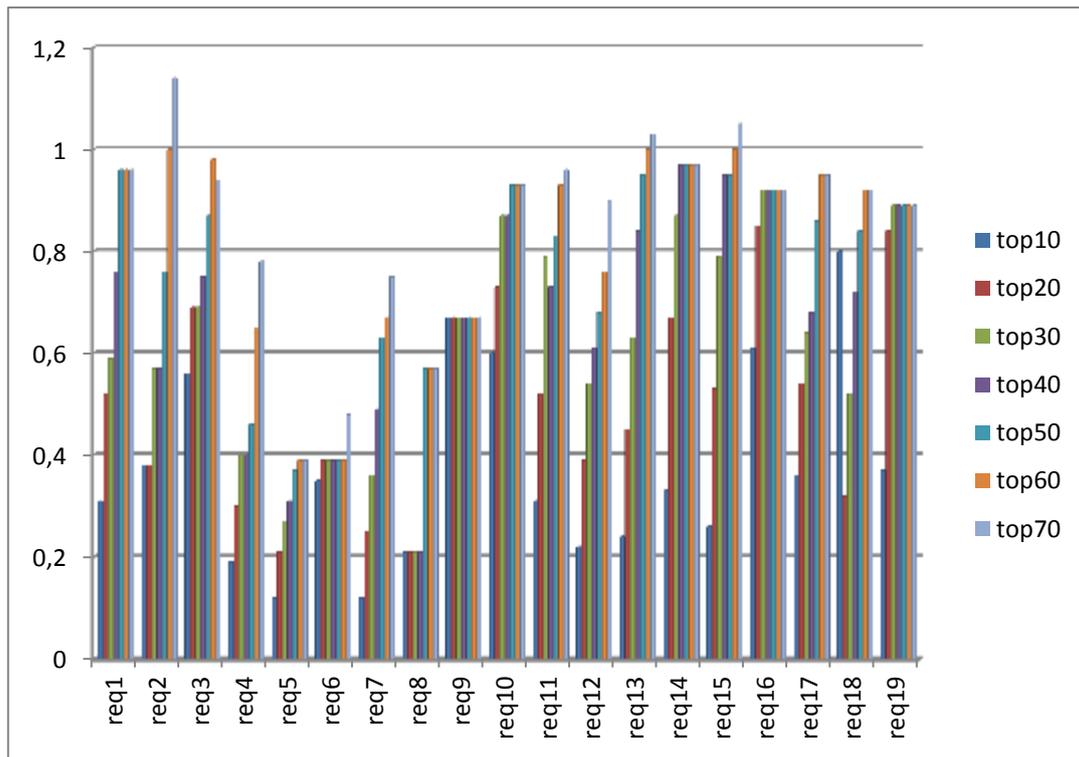
Les figures ci-dessus montrent le score de Rappel et de précision :



**Figure II. 10** Histogramme du score de Précision

| Requête | top10      | top20      | top30      | top40      | top50      | top60      | top70      |
|---------|------------|------------|------------|------------|------------|------------|------------|
| req1    | 0,9        | 0,75       | 0,57       | 0,55       | 0,56       | 0,47       | 0,4        |
| req2    | 0,8        | 0,4        | 0,4        | 0,3        | 0,32       | 0,35       | 0,34       |
| req3    | 0,9        | 0,55       | 0,37       | 0,3        | 0,28       | 0,25       | 0,21       |
| req4    | 0,7        | 0,55       | 0,5        | 0,37       | 0,34       | 0,4        | 0,41       |
| req5    | 0,6        | 0,55       | 0,47       | 0,4        | 0,38       | 0,33       | 0,28       |
| req6    | 0,8        | 0,45       | 0,3        | 0,22       | 0,18       | 0,15       | 0,16       |
| req7    | 0,9        | 0,95       | 0,9        | 0,92       | 0,94       | 0,83       | 0,8        |
| req8    | 0,3        | 0,15       | 0,1        | 0,1        | 0,16       | 0,13       | 0,11       |
| req9    | 0,2        | 0,1        | 0,07       | 0,05       | 0,04       | 0,03       | 0,3        |
| req10   | 0,9        | 0,55       | 0,43       | 0,32       | 0,28       | 0,23       | 0,2        |
| req11   | 0,9        | 0,75       | 0,77       | 0,57       | 0,48       | 0,45       | 0,4        |
| req12   | 0,9        | 0,8        | 0,73       | 0,62       | 0,56       | 0,52       | 0,53       |
| req13   | 0,9        | 0,85       | 0,8        | 0,8        | 0,72       | 0,63       | 0,56       |
| req14   | 1          | 0,1        | 0,87       | 0,72       | 0,58       | 0,48       | 0,41       |
| req15   | 0,5        | 0,5        | 0,5        | 0,45       | 0,36       | 0,32       | 0,28       |
| req16   | 0,8        | 0,55       | 0,4        | 0,3        | 0,24       | 0,2        | 0,17       |
| req17   | 0,8        | 0,6        | 0,47       | 0,37       | 0,38       | 0,35       | 0,3        |
| req18   | 0,2        | 0,4        | 0,43       | 0,45       | 0,42       | 0,38       | 0,33       |
| req19   | 0,7        | 0,8        | 0,57       | 0,42       | 0,34       | 0,29       | 0,24       |
| Moyenne | 0,72105263 | 0,54473684 | 0,50789474 | 0,43315789 | 0,39789474 | 0,35736842 | 0,33842105 |

**Figure II. 11** Tableau des scores de Précision



**Figure II. 12 Histogramme du score de Rappel**

| Requête | top10     | top20     | top30     | top40 | top50      | top60      | top70      |
|---------|-----------|-----------|-----------|-------|------------|------------|------------|
| req1    | 0,31      | 0,52      | 0,59      | 0,76  | 0,96       | 0,96       | 0,96       |
| req2    | 0,38      | 0,38      | 0,57      | 0,57  | 0,76       | 1          | 1,14       |
| req3    | 0,56      | 0,69      | 0,69      | 0,75  | 0,87       | 0,98       | 0,94       |
| req4    | 0,19      | 0,3       | 0,4       | 0,4   | 0,46       | 0,65       | 0,78       |
| req5    | 0,12      | 0,21      | 0,27      | 0,31  | 0,37       | 0,39       | 0,39       |
| req6    | 0,35      | 0,39      | 0,39      | 0,39  | 0,39       | 0,39       | 0,48       |
| req7    | 0,12      | 0,25      | 0,36      | 0,49  | 0,63       | 0,67       | 0,75       |
| req8    | 0,21      | 0,21      | 0,21      | 0,21  | 0,57       | 0,57       | 0,57       |
| req9    | 0,67      | 0,67      | 0,67      | 0,67  | 0,67       | 0,67       | 0,67       |
| req10   | 0,6       | 0,73      | 0,87      | 0,87  | 0,93       | 0,93       | 0,93       |
| req11   | 0,31      | 0,52      | 0,79      | 0,73  | 0,83       | 0,93       | 0,96       |
| req12   | 0,22      | 0,39      | 0,54      | 0,61  | 0,68       | 0,76       | 0,9        |
| req13   | 0,24      | 0,45      | 0,63      | 0,84  | 0,95       | 1          | 1,03       |
| req14   | 0,33      | 0,67      | 0,87      | 0,97  | 0,97       | 0,97       | 0,97       |
| req15   | 0,26      | 0,53      | 0,79      | 0,95  | 0,95       | 1          | 1,05       |
| req16   | 0,61      | 0,85      | 0,92      | 0,92  | 0,92       | 0,92       | 0,92       |
| req17   | 0,36      | 0,54      | 0,64      | 0,68  | 0,86       | 0,95       | 0,95       |
| req18   | 0,8       | 0,32      | 0,52      | 0,72  | 0,84       | 0,92       | 0,92       |
| req19   | 0,37      | 0,84      | 0,89      | 0,89  | 0,89       | 0,89       | 0,89       |
| Moyenne | 0,3689473 | 0,4978947 | 0,6110526 | 0,67  | 0,76315789 | 0,81842105 | 0,85263158 |

**Figure II. 13 Tableau des Score du Rappel**

|             | TOP10 | TOP20 | TOP30 | TOP40 | TOP50 | TOP60 | TOP70 |
|-------------|-------|-------|-------|-------|-------|-------|-------|
| Domin-proba | 0,72  | 0,54  | 0,50  | 0,43  | 0,39  | 0,35  | 0,33  |
| FDDA        | 0,71  | 0,55  | 0,45  | 0,38  | 0,32  | 0,28  | 0,26  |

**Figure II. 14 Table de comparaison de score de précision des deux approches**

|             | TOP10 | TOP20 | TOP30 | TOP40 | TOP50 | TOP60 | TOP70 |
|-------------|-------|-------|-------|-------|-------|-------|-------|
| Domin-proba | 0,36  | 0,49  | 0,61  | 0,67  | 0,76  | 0,81  | 0,85  |
| FDDA        | 0,33  | 0,47  | 0,55  | 0,60  | 0,63  | 0,65  | 0,68  |

**Figure II. 15 Table de comparaison de score de Rappel des deux approches**

### Discussion des résultats

Nous constatons d'après les moyennes du Rappel et de précision que les résultats de notre approche ont donné pour la majorité des taux élevé par rapport à ceux de l'approche FDDA<sup>7</sup>,sauf pour la précision du Top20.

### Conclusion

Nous avons présenté dans ce chapitre l'approche de dominance probabiliste pour la découverte de service web, les résultats confirment l'efficacité de cette approche dans ce genre de problème. Donc on conclut que notre approche est excellente.

<sup>7</sup> FDDA (Fuzzy Dominated aggregation)

# Conclusion générale

## Conclusion générale et Perspectives

Nous avons présenté dans ce mémoire les technologies liées aux services Web, ainsi que les protocoles de transmission...etc. Nous avons proposé aussi une des approches hybrides qui se base sur les probabilités. Notre prototype adopte l'approche de PD (probabilité de dominance) comme une solution du problème de découverte de service web pertinent, l'approche a donné des résultats encourageants, donc on peut dire que cette approche est l'une des solutions prometteuses pour les problèmes de découverte.

Comme perspectives à ce travail, nous proposons d'implémenter d'autres algorithmes de découverte, pour cela, nous pouvons citer :

- ✓ K-Dominance Probabiliste
- ✓ Le vote majoritaire hybridé avec les probabilités
- ✓ L'apprentissage des paramètres de la fonction de dominance



## Références bibliographiques

- [Albertella 2009]** D. Albertella, Les avantages compétitifs de SOA : aspects techniques, organisationnels et financiers est-ce que SOA remplit ses promesses ? , Université de Fribourg, Suisse, 2009.
- [Bejaoui 2008]** R. Bejaoui, SOA : les impacts sur le cycle de vie du projet de développement logiciel. Maîtrise en informatique de gestion, Université de Québec à Montréal, 2008.
- [Bieberstein 2008]** Norbert Bieberstein, Keith Jones robert, et Tilak  
Executing SOA: a Practical Guide for the Service-O  
Architect, édition IBM Press, Mai 2008.
- [Cerami 2002]** Ethan Cerami, Web Services Essentials, édition O'Reilly,  
Février 2002.
- [Chappell 2002]** David Chappell, et Tyler JEWELL, Java Web Service, édition  
O'Reilly, Mars 2002.
- [Devaux 2008]** C. Devaux, Urbanisation et SOA : Quelques bonnes pratiques  
pour leur mise en œuvre, Livre Blanc, Aubay, France, 2008
- [Erl 2008]** Thomas Erl, SOA Principles of Service Design, édition prentice  
hall, juillet 2008.
- [Fan et al., 2005]** J. Fan, B. Ren, and L.-R. Xiong. An Approach to Web Service  
Discovery Based on the Semantics. In FSKD (2), pages 1103–1106, 2005
- [Gardien 2002]** Georges Gardien, XML des bases de données aux Services Web,  
édition Dunod, 2002.
- [Jaeger, 2011]** Michael C. Jaeger. OWL-S Matcher.
- [Josuttis 2007]** Nicolai Josuttis, SOA in Practice, édition O'Reilly, Septembre  
2007.

- [Kaabi 2008]** R. S. Kaabi, Une approche méthodologique pour la modélisation intentionnelle des services et leur opérationnalisation, Thèse de Doctorat de l'Université Paris 1, Sorbonne, 2008.
- [Kreger 2001]** Heather Kreger, Web Service Conceptual Architecture, édition IBM Software Group, Mai 2001.
- [Klein et al., 2005]** M. Klein, B. König-Ries, and M. Mussig. What is needed for semantic service descriptions: A proposal for suitable language constructs. *Int. J. Web Grid Serv.*, 1(3/4) :328–364. 2005.
- [Keller et al., 2005]** U. Keller, R. Lara, H. Lausen, A. Polleres, and D. Fensel. Automatic location of services. In *Proc. of the 2nd European Semantic Web Conference (ESWC)*, pages 1–16, Heraklion, Crete, 2005. LNCS 3532, Springer.
- [Küster et König-Ries, 2008]** U. Küster and B. König-Ries. Evaluating semantic web service matchmaking effectiveness based on graded relevance. In *Proc. of the 2nd International Workshop SMR on Service Matchmaking and Resource Retrieval in the Semantic Web at the 7th International Semantic Web Conference (ISWC08)*, Karlsruhe, Germany, October 2008.
- [Margolisand 2007]** Ben Margolisand, et Joseph Sharpe, SOA for the Business Developer: Concepts, BPEL, and SCA, édition MC Press
- [Newcomere 2004]** Eric Newcomere, Understanding Web Services Xml WSDL SOAP and UDDI, édition O'Reilly, 2004.
- [Palathingal et-Chandra, 2004]** Palathingal P. and Chandra S.; Agent approach for service discovery and utilization. In *HICSS*, 2004.
- [Paolucci et al., 2003]** Paolucci, M., Srinivasan, N. Sycara, K., Nishimura, T. Toward a Semantic Choreography of Web Services: From WSDL to DAML-S. In: *Procs of the International Conference of Web Services (ICWS03)*, June 2003, Las Vegas, Nevada, USA. CSREA Press 2003, pp.22-26.

- [Papazoglou 2003]** M.P. Papazoglou, Web Services and Business Transactions, World Wide Web Internet and Web Information Systems, edition Kluwer Academic, 2003.
- [Ponge 2008]** Julien Ponge. Model Based Analysis of Time-aware Web Services Interactions. Thèse de Doctorat de l'Université Blaise Pascal - Clermont-Ferrand II, dans le cadre de l'Ecole Doctorale des Sciences pour l'Ingénieur, France, 2008.
- [Schumacher et al., 2008]** M. Schumacher, H. Helin, and H. Schuldt. Semantic Web Service Coordination. Chapter 4, CASCOM: Intelligent Service Coordination in the Semantic Web, Birkhäuser Basel, 2008
- [Scott 2002]** Scott Short, Construire des Services Web XML, édition Dunod, 2002.
- [Srinivasan et al., 2004a]** Srinivasan N., Paolucci M., Sycara K.; An Efficient Algorithm for OWL-S Based Semantic Search in UDDI, Semantic Web Services and Web Process Composition, Springer-Verlag Berlin Heidelberg, 2004, pp96-100.
- [Srinivasan et al., 2004b]** N. Srinivasan, M. Paolucci, and K. Sycara. Adding OWL-S to UDDI, implementation and throughput. In In First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004), pages 6–9, 2004.
- [Srinivasan et al., 2006]** Srinivasan N., Paolucci M., Sycara K. Semantic Web service Discovery in the OWL-S IDE, Proceedings of the 39th Annual Hawaii Intel. Conf. On System Sciences (HICSS'06) Track 6, 2006
- [Stollberg et al., 2007]** M. Stollberg, U. Keller, H. Lausen, and S. Heymans. Two-Phase Web Service Discovery Based on Rich Functional Descriptions. In ESWC '07 : Proc. Of the 4th European conference on The Semantic Web, pages 99–113, Berlin, Heidelberg,
- [Schreiner 2005]** Wolfgang Schreiner, et Schahram tdar, A survey on Web services composition, Int. J. Web and Grid Services, 2005.

Annexe

## Annexe

**XML** : est une recommandation du W3C. C'est un langage de balisage définissant un format universel de représentation des données, permettant de décrire la structure hiérarchique d'un document. Ainsi, un document XML contient à la fois les données et les indications sur le rôle que jouent ces données. Ces indications permettent de déterminer la structure du document. XML est défini pour fonctionner indépendamment des plateformes et des systèmes d'exploitation.

**B2B** (Business-to-Business): Qualificatif signifiant une interaction entre deux entités d'affaires. Par exemple, nous pouvons utiliser l'expression marketing B2B, qui signifie le marketing d'entreprise à entreprise.

**B2C** : Le Business to Consumer aussi appelé Business to Customer est le nom donné à l'ensemble d'architectures techniques et logiciels informatiques permettant de mettre en relation des entreprises directement avec les consommateurs : en français, « des entreprises aux particuliers ».

**CORBA** (Common Object Request Broker Architecture) : Architecture et spécifications visant à permettre de créer, publier et gérer des objets, sur un réseau, dans un contexte de programmation distribuée. Cette architecture a déjà été largement utilisée sauf par Microsoft qui avait développé sa propre architecture DCOM.

**W3C** (World Wide Web Consortium) : Consortium industriel visant à favoriser l'interopérabilité des produits informatique et l'évolution du World Wide Web, par le développement de standards et de spécifications techniques.

**OASIS**(Organization for the Advancement of Structured Information Standards): OSBL technologique international visant la promotion et l'adoption de standards indépendants des manufacturiers, dans le domaine des technologies de l'information. OASIS tente de rapprocher les acteurs industriels et les groupes de standardisation afin qu'ils s'entendent sur l'utilisation universelle de XML.

Protocole: Ensemble de règles qui sont partagée entre deux ordinateurs afin de communiquer l'un avec l'autre.

**HTTP** (Hypertext Transfer Protocol) : Protocole de communication permettant l'échange de fichiers sur le World Wide Web. HTTP est un protocole d'application développé par le W3C.

**SMTP** (Simple Mail Transfer Protocol (littéralement) :est un protocole de communication utilisé pour transférer le courrier électronique vers les serveurs de messagerie électronique.

**FTP** File Transfer Protocol :est un protocole de communication destiné à l'échange informatique de fichiers sur un réseau TCP/IP. Il permet, depuis un ordinateur, de copier des fichiers vers un autre ordinateur du réseau, ou encore de supprimer ou de modifier des fichiers sur cet ordinateur. Ce mécanisme de copie est souvent utilisé pour alimenter un site web hébergé chez un tiers.

**HTML** (Hypertext Markup Language) : Ensemble de balises et de symboles insérés dans un fichier informatique visant à être affiché sur un navigateur World Wide Web. Les balises HTML définissent comment un navigateur Web devra afficher les informations contenues dans une page Web. HTML a été développé par le W3C.

**API** (Application Programming Interface) :est une interface fournie par un programme informatique. Elle permet l'interaction des programmes les uns avec les autres, de manière analogue à une interface homme-machine, qui rend possible l'interaction entre un homme et une machine.

## Résumé

Au cours de ces dernières années les entreprises ont largement utilisé la technologie des services Web pour faire l'interopérabilité et l'intégration de leurs applications.

Pour réaliser efficacement ces objectifs, les entreprises doivent localiser en premier lieu, les services qui répondent à leur besoins, et pour ce faire il est impératif de développer des modèles et des algorithmes assez adéquats pour cette tâche.

Les approches développées actuellement pour la découverte de services Web sont peu efficaces. Dans ce travail nous proposons une solution qui agrège plusieurs méthodes de recherche. La fusion de ces approches est faite grâce à la notion de dominance probabiliste. Cette dernière estime le pourcentage d'instances (de services) dominées par un candidat donné. Les résultats expérimentaux sont satisfaisants et démontrent l'utilité de l'approche.

**mots clés** : architecture orientée services, découverte de services, fusion de données, agrégation probabiliste.

## Abstract

The web service discovery is one of the most challenging issues in service oriented computing. In this work we propose an approach that aggregates multiple retrieval methods. The fusion of these algorithms is done through the concept of probabilistic dominance. This later estimates the percentage of service instances that are dominated by a particular candidate. The experimental results are very encouraging and demonstrate the utility of the approach.

**Key words:** Web Service ranking, Web Service Discovery, Service Matching, probabilistic Dominance, , Rank Aggregation, similarity measures.

## ملخص

اكتشاف خدمات الإنترنت هي إحدى أكبر التحديات في مجال الحوسبة، لكن المناهج المطورة حالياً لاكتشافها ضعيفة. ولتحقيق هذه الأهداف على نحو فعال، يجب على الشركات أولاً تحديد موقع الخدمات التي تلبي احتياجاتهم، ولذلك لا بد من تطوير نماذج وخوارزميات كافية تماماً للقيام بهذه المهمة. في هذا العمل نقترح نهجاً يجمع أساليب بحث متعددة. حيث يتم دمج هذه الخوارزميات من خلال مفهوم الهيمنة الاحتمالية. هذا يقدر في وقت لاحق النسبة المئوية لحالات الخدمة التي يهيمن عليها مرشح معين. النتائج التجريبية مشجعة جداً وتوضح فائدة هذا النهج.

الكلمات الرئيسية: ترتيب خدمة ويب، اكتشاف خدمة ويب، مطابقة الخدمة، الهيمنة الاحتمالية، الرتبة التجميع، تدابير التشابه.

