

Table des matières

Introduction générale	10
Contexte.....	10
Problématique.....	10
Contribution.....	11
I. Introduction	12
II. Définitions L'Architecture Orienté Service (SOA).....	12
II.1 Définition Métier	12
II.2 Définition Technique.....	12
II.3 Caractéristiques de SOA	13
II.3.1 Couplage faible.....	13
II.3.2 Frontières explicites	14
II.3.3 indépendance par rapport à La plateforme.....	14
II.4.4 Le langage de programmation	14
II.4.5 le modèle d'un service	14
II.3.Les avantages de SOA	15
III. Les Services Web.....	15
III.1.Définition :	15
III.2.Standard des Services Web	16
III.2.1.Communication avec SOAP.....	16
III.2.2.Description des Web services WSDL.....	17
III.2.3.Découverte du service Web UDDI	19
III.2.4.XML	19
III.3.Caractéristiques des services Web	20
III.4.Quelques avantages des services web.....	21

III.5. Service Web Sémantique	21
IV. Problématique de découverte de service	22
IV.1.Définition	22
IV.2.Approche de découverte.....	23
V. Conclusion.....	24
I. Introduction	27
II. Présentation du Corpus	27
III. Les mesures de similarité.....	28
III.1 Extended Jaccard(EJ) X_H	28
III.2 Cosinus (cos)	28
III.3 Loss of information (LOI)	28
III.3 Jensen-Shannon (JS) :	28
III.4 Logique (log) :	28
IV. L' approches utilisée.....	30
IV.1.L'approche Coombs	30
IV .1.1 algorithme :	30
IV .1.2 Les fonctions utilisées.....	31
IV .1.3 Déroulement de l'algorithme	31
IV.1.4.Exemple d'application.....	32
IV.2.Présentation du prototype.....	34
IV.2.1 Outils et environnement de développement.....	34
IV.2.2 présentation du prototype	35
IV.3.Expérimentation.....	37
IV.4. Conclusion	39
Conclusion générale.....	41
Références bibliographique	43

Liste des figures

Figure I.1. Structure de message SOAP	17
Figure I.2. Structure d'un document WSDL [9].....	18
Figure I.3. Les trois pages de l'annuaire UDDI [9].....	19
Figure I.4. Architecture orientée services (SOA) [5].....	20
Figure II.1. Interface d'accueil	35
Figure II.2. Interface d'élimination des services de score	35
Figure II.3. exemple d'exécution de la figure II.3	36
Figure II.4. calcul de rappel et précision	36
Figure II.5. Affichage de résultat rappel et précision	37

Liste des tables

Tableau I.1. Les couches technologiques des web services [7].....	16
Tableau II.1. : etape1	32
Tableau II.2. etape2	33
Tableau II.3. etape3	33
Tableau II.4. etape4	33
Tableau II.5. etape5	34
Tableau II.6. etape6	34
Tableau II.7. Les rappels des approches	38
Tableau II.8. Les précisions des aproches	38

Introduction

Générale

Introduction générale

Contexte

Dans les dernières années, les services web occupent un espace très important dans l'informatique parce qu'ils représentent une solution largement répondue à plusieurs problèmes posés. Qui sont engendrés par l'utilisation de la technologie hétérogène et l'augmentation des traitements des données et même avec l'exigence d'utilisateur.

L'utilisation de la technologie hétérogène rend la communication entre les applications et les différents systèmes d'informations plus difficiles.

À cause de l'utilisation de cette technologie hétérogène, le système d'information (S.I) est devenu lourd et rigide, cela a conduit à un manque d'interopérabilité. Les experts de ce domaine ont découvert des nouvelles méthodes pour résoudre ces problèmes.

Parmi les technologies qui sont apparues comme solution, c'était la notion « des services web ».

Le service web est la base de l'architecture orientée service(SOA), cette approche a été développée par diverses entreprises de recherche en informatique telle que IBM, BEA, Oracle ...etc. SOA utilise les services qui sont décrits et publiés par le fournisseur, pour que les clients l'utilisent et exploitent dans leurs recherches. Ils semblent être une solution plus efficace par rapport aux anciennes technologies (CORBA, DCOM ...) et qui offre l'interopérabilité et l'intégration de leurs applications.

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.

Problématique

En ce jour, l'internet connaît le développement rapide des services web qui s'accroît de manière explosive, avec l'augmentation des exigences des clients qui sont devenus de nature multi objectifs (fonctionnels et non fonctionnels,...), il est devenu difficile de découvrir les meilleurs services web. Ainsi plusieurs services sont découverts lors de la demande (requête) du client, parce qu'ils ont une spécification similaire qui est décrite par les fournisseurs.

Notre objectif est d'établir une méthode de recherche qui optimise au mieux les critères de rappels et de précision.

Contribution

Dans ce travail, nous proposons une approche de vote pour résoudre le problème de découverte de service, cette approche est nommée méthode de Coombs ou vote exhaustif [1]. c'est un système de vote par classement inventé par Clyde Coombs vers 1954. Cette méthode consiste à mettre le service le plus rejeté (qui est classé dernier dans la majorité des mesures de similarité) dans la queue de la liste fusionnée.

En cas d'équivalence, nous choisirons celui qui a la somme de scores la plus faible.

Plan de mémoire

Notre mémoire est composé des chapitres suivants :

Chapitre 1: Ce chapitre se divisera en trois parties: la première présente l'architecture orienté service (SOA), en exprimant ses caractéristiques et ses avantages, ensuite, dans la deuxième partie nous décrivons qu'est ce qu'un service web, et ses standards, après, nous exposons la problématique de découverte de service web en citant l'approche sémantique.

Chapitre 2: Dans ce chapitre nous allons voir: tout d'abord la description de la collection de test, ensuite nous présentons nos algorithmes, par la suite, nous montrons le prototype ainsi que les résultats expérimentaux, et enfin nous terminons par une conclusion.

Chapitre : I

L'architecture SOA et les services web

I. Introduction

Les entreprises avaient besoin d'une nouvelle approche capable: de manipuler une grande quantité des données entre les différentes applications quelque soit leur plateforme, aussi les aider à augmenter la flexibilité de système d'information(SI), et de réduire le coût d'intégration.

Ce fut la raison de la naissance de l'architecture orientée service(SOA), cette dernière est apparue en 1996 dans une note de recherche du Gartner Groupe¹, Elle permet des interactions entre des applications distantes à travers des nouveaux standards comme (SOAP, WSDL,...).

II.Définitions L'Architecture Orienté Service (SOA)

L'architecture SOA peuvent être définit et expliqué par plusieurs définitions, ces définitions se basent sur les caractéristiques techniques de l'architecture, en outre d'autres porte attention aux aspects métiers. Les définitions suivantes clarifient les différentes vues de laSOA. 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.2 Définition Technique

« L'architecture SOA est un paradigme permettant d'organiser et d'utiliser des savoir faire distribués pouvant être de domaines variés. Cela fournit un moyen uniforme d'offrir, de découvrir, d'interagir et d'utiliser des savoirs faire 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

1

Est une entreprise américaine de conseil et de recherche dans le domaine des techniques avancées. Elle mène des recherches, fournit des services de consultation, tient à jour différentes statistiques et maintient un service de nouvelles spécialisées

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.3 Caractéristiques de 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.3.1 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 système d'exploitation, Le langage de programmation, et les bibliothèques utilisées.

II.3.2 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.3.3 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.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.4.5 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 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 coût. SOA est une architecture fournissant une infrastructure nécessaire, pour intégrer les applications isolées afin de les utiliser au 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 services Web, ces derniers sont des réalisations concrètes 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].

L'architecture orientée service : c'est un modèle qui définit le système par un ensemble des services. Selon Fournier-Morel : les services sont des vues logiques qui permettent la manipulation des données développées ou qui sont déjà codées [2].

Les services exposent des interfaces fonctionnelles (WSDL) et communiquent par des messages via des protocoles (SOAP). Ils sont basés sur le principe de couplage faible où SOA crée des services qui sont indépendants et autonomes entre eux. Cela permet d'échanger les données entre les agents (client /fournisseur) facilement sans connaître le langage de script utilisé, langage de programmation, système

d'exploitation, ainsi offre des services réutilisables qui jouent des rôles différents et communiquent avec plusieurs utilisateurs.

Cette méthode diminue la redondance des services, augmente la flexibilité et renvoi une information très rapide.

II.3. Les avantages de SOA

L'architecture orientée service est apparue comme une solution efficace pour guérir le système d'information, parmi les avantages de SOA nous citons quelques uns :

- ✓ Architecture interopérable et facilement maintenable: indépendance entre les services.
- ✓ Indépendance de plateforme, langage de programmation, système d'exploitation.
- ✓ Possibilité de choisir le protocole de transport.
- ✓ Augmenter la flexibilité.
- ✓ Minimiser le cout de réduction.

III. Les Services Web

L'architecture orientée service se caractérise par l'utilisation des services web qui sont des composants logiciels disponibles sur le réseau internet, elle permet d'échanger les données et la communication entre différentes applications quelque soit leurs plateformes ou langage de programmation.

Les éléments de service web sont identifiés par un URI, leur interface est décrite en XML, le protocole de transforme http permet de communiquer entre différents services web. Il a interagit directement à travers des protocoles (SOAP, WSDL...) [3].

III.1. Définition :

D'après **W3C** « Un service Web 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. Les services Web peuvent interagir entre eux d'une manière prescrite par leurs définitions, en utilisant des messages XML portés par les protocoles Internet ».

« 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»[4].

III.2.Standard des Services Web

Les services web basés sur des standards permettant de circuler les données sur internet, ils reposent sur trois couches fondamentales.

- la communication est réalisée par le standard SOAP.
- la description du service est réalisée par le standard WSDL.
- les registres des services utilisent le standard universel UDDI.

UDDI	Découverte de services
WSDL	Description de services
SOAP	Communications
XML	

Tableau I.1. Les couches technologiques des web services [5]

III.2.1.Communication avec SOAP

SOAP c'est un protocole de communication, proposé par W3C, Il est produit par Microsoft et IBM. Le but principal de SOAP est de transférer les messages qui sont écrits sur le format XML sur des environnements indépendants.

SOAP est un protocole de type requête/réponse, il peut être employé dans tous les styles de communication: synchrones ou asynchrones, point à point ou multi-points [6].Il utilise le protocole HTTP ou d'autres protocoles comme SMTP, FTP, etc.

□ La structure de message SOAP

Un message SOAP est un document XML. composé de deux parties, l'en-tête de protocole de transport et l'enveloppe SOAP.



Figure I.1. Structure de message SOAP

- L'élément Enveloppe: c'est l'en-tête de tous les messages SOAP qui identifie son contenu de message et comment le traiter. L'enveloppe contient la spécification des espaces de nom définissent la version de SOAP utilisée et un attribut optionnelle « **encodingStyle** » permettant d'identifier les règles d'encodage. Une enveloppe est divisée en 2 parties: Le header et le corps.
 - L'élément Header (entête): qui est optionnel et qui contient des informations concernant l'authentification, la gestion des transactions, le payement, etc.
 - L'élément Body: contient les informations de la requête ou de la réponse, le body de la requête identifier l'objet distant, le nom de la méthode à exécuter et les éventuels paramètres ou des rapports d'erreur 'Fault'.
- L'élément Fault: est optionnel et fournit des informations sur d'éventuelles erreurs (le type d'erreur, une description de l'erreur et l'adresse du serveur SOAP).

III.2.2. Description des Web services WSDL

Le langage WSDL est proposé par W3C, Il est produit par Microsoft et IBM. C'est un format de description des Web services fondé sur XML.

WSDL regroupe un ensemble de descriptions nécessaire pour définir un service web, c'est le cœur d'un SW il définit leur interface, les types des données établis la structure et le contenu d'un message, les opérations effectuées sur ce contenu et le mappage des protocoles. WSDL garantit la compatibilité et assure l'interopérabilité même dans une différente implémentation qui peut être difficile [9].

• Structure d'un document WSDL:

Le fichier WSDL contient les éléments suivants :

Types: fournit la définition de types de données utilisés pour décrire les messages échangés entre le client et le serveur sous le format de XML schéma.

Messages: représente une définition abstraite qui définit les données en cours de transmission entre les services web.

PortTypes: combine plusieurs messages pour former une opération. qui définit un couple message-entrée / message – sortie.

Chaque opération a zéro ou un message en entrée, zéro ou plusieurs messages de sortie ou d'erreurs.

Binding: c'est un élément qui spécifie une liaison entre un <portType> et un protocole concret (SOAP, HTTP...).

Service : une collection des <Ports>, contenant chacun un nom, une URL de point d'accès et une référence à une liaison donnée.

Port: représente un point d'accès de services défini par une adresse réseau et une liaison.

Opération: c'est la description d'une action exposée dans le port.

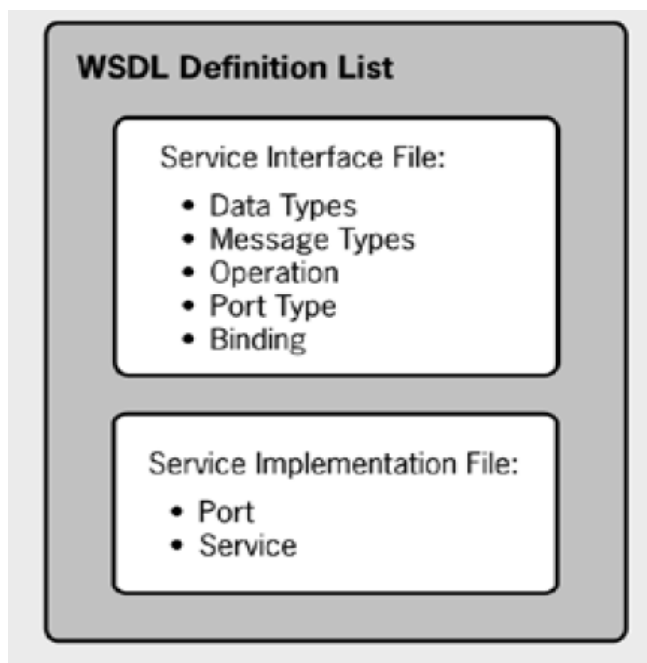


Figure II2. Structure d'un document WSDL [9]

III.2.3.Découverte du service Web UDDI

UDDI c'est une norme standardisé par OASIS. C'est comme une base de registre des services web sur un format XML.

UDDI a pour objectif d'offrir aux fournisseurs la possibilité de stoker et publier leurs services et même modifie, permettre au client de découvrir, consulter les informations sur les services et sélectionner pour répondre à ces besoins [8].

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

- **Les pages blanches:** elles contiennent toutes les informations pertinentes pour identifier l'organisation (nom, l'adresse, identifiants).
- **Les pages jaunes:** elles détaillent la description de l'organisation faite dans les pages blanches (la catégorie de l'entreprise, le secteur d'activité dans lequel exerce l'entreprise, les services offerts par cette organisation, le type de service, prix, qualité de service...etc.).
- **Les pages vertes:** donnent les informations techniques sur les services web.

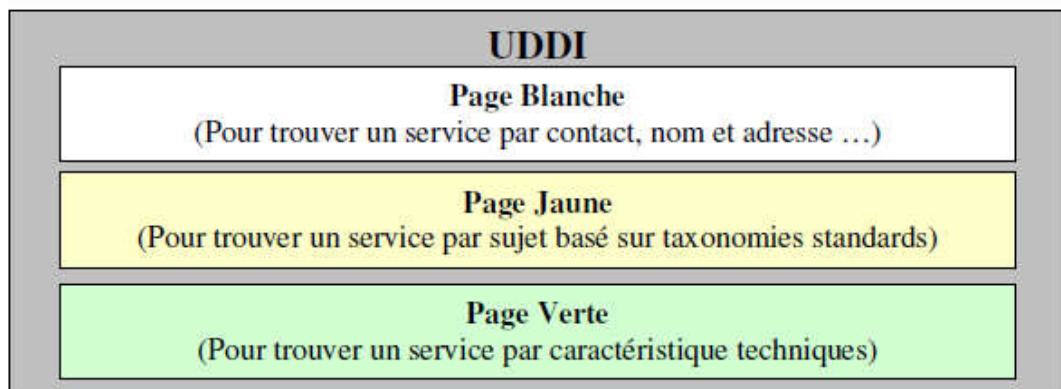


Figure III.3. Les trois pages de l'annuaire UDDI [9]

III.2.4.XML

Les services web sont utilisés dans un différent environnement totalement hétérogène en terme de plate forme, système d'exploitation, langage de programmation. Ceci demande un langage capable d'échanger des messages entre les différents participants. C'est pour cette raison qu'aujourd'hui, de nombreuses entreprises utilisent le format XML pour échanger les données sur le web.

XML c'est un langage balisé, il est apparu en 1998 par W3C, c'est un métalangage de présentation qui transforme les données. Les balises sont personnalisées contrairement à HTML qui possède un ensemble des balises qui sont très définies, il se présente sous forme d'un arbre, la racine constitue le sujet et les

feuilles sont le contenu des documents. il est très facile à comprendre, utiliser et obtenir l'interopérabilité.

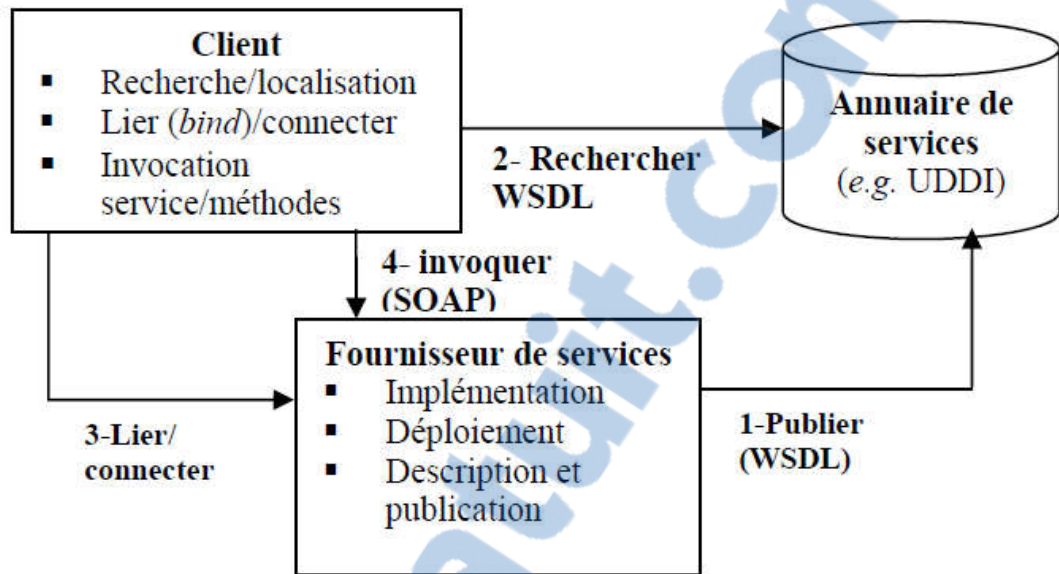


Figure IV.4. Architecture orientée services (SOA)

L'architecture de service web est composée de 3 acteurs: le fournisseur, l'annuaire et le client.

➤ **Fournisseur:** c'est le propriétaire de service web, il le crée et puis le dispose au format WSDL dans l'annuaire de service en utilisant l'opération de "publier".

➤ **Le client:** correspond au consommateur de service web, il peut accéder aux annuaires pour chercher, retrouver et sélectionner un service nécessaire sous format WSDL, utilisant opération "chercher". à partir de cette description il offre La communication entre le demandeur de service et le fournisseur par le biais du protocole SOAP

➤ **L'annuaire des services:** c'est une entité qui offre au fournisseur le moyen de publier leur service et d'autre part il reçoit et répond aux demandes de l'utilisateur avec URL de service, à travers l'utilisation de l'opération "lier". c'est un registre de description des services web.

III.3.Caractéristiques des services Web

Les services Web peuvent être définis, décrits et découverts à travers le Web qui permet une accessibilité facile.

- Un service Web est accessible via le réseau à partir de leur identification URL.
- Un service Web est fonctionnelle quelque soit la plateforme, il est complètement indépendant des langages de programmation et des systèmes d'exploitation.

- Les services Web sont décrits leur interface par des langages standards basés sur XML.
- Les services Web puissent peuvent communiquer par des différentes applications et invoquées à distance a travers l'utilisation des protocoles comme (HTTP, SMTP...) pour échanger les données.
- Les services web peuvent être communiqués entre eux facilement grâce à utilisation le langage XML.

III.4.Quelques avantages des services web

- ✓ Le client doit simplement exprimer ces besoins et les entreprises doivent simplement publier leur service web.
- ✓ Les services Web offrent une interface simple: à partir de WSDL, ils ont été créés en fonction de langage XML, qui accorde la relation de compréhension entre le fournisseur et le client, et aussi entre l'entreprise et la société.
- ✓ Les services Web permettent de garantir l'interopérabilité entre les applications: ces derniers peuvent communiquer, échanger et partager des fonctionnalités quelque soit leur langage de programmation qui évolue sur n'importe quel système d'exploitation.
- ✓ Permettre d'échanger les informations en toute sécurité à l'aide des protocoles standards comme HTTP qui utilise le port 80 qui fonctionne aux nombreux pare-feux sans nécessiter des changements sur les règles de filtrage.
- ✓ Service Web est indépendant des langages de programmation et des systèmes d'exploitation donc il est très flexible [9].

III.5. Service Web Sémantique

L'internet est une intermédiaire qui permet aux utilisateurs d'accès à des services qui sont présents sur le format des pages web, les utilisateurs peuvent lire et comprendre facilement, mais leurs significations sémantiques n'est pas claire et elle est difficilement interprétables par des ordinateurs.

La description des services web est basée sur WSDL est présentée sur le format de XML.

WSDL représenté bien les services web syntaxiquement mais pas sémantiquement. On peut trouver deux services qui ont le même (noms et types) mais fonctionnement sont différents ou inverse.

Pour garantir une meilleure communication et échange des données d'une manière efficace, il faut que l'ordinateur comprend et traite l'information automatiquement. C'est le but de créations des services web sémantiques.

La vision du Web sémantique a été présentée la première fois par Tim Berners-Lee, il a défini le web sémantique comme une «le web sémantique est une extension du web actuel dans laquelle on a donné un sens précis à l'information, de manière à améliorer le travail coopératif des hommes, et aussi des machines» [10].

IV. Problématique de découverte de service

De nos jours, Le nombre des services web s'évolue rapidement, avec l'apparition de web sémantique et l'augmentation des exigences de clients pour trouver les services souhaités, la découverte est devenue plus importante et nécessaire pour partager, échanger et réutiliser des informations du web d'une manière efficace entre les utilisateurs [11].

Pour cela l'objectif de SOA est d'offrir un environnement dynamique et hétérogène pour découvrir et choisir le meilleur service parmi eux et exploiter d'une manière plus "intelligente" par les machines. Ils pourraient "comprendre" les contenus décrits dans les ressources et faciliter les tâches de traitement des informations de façon plus automatique et plus efficace.

Le service est découvert à partir de la comparaison entre la requête de clients et un ensemble de services qui sont disponibles et publié sur le Web, elle utilise les descriptions d'interfaces, descriptions comportementales (fonctionnelles) ou (non fonctionnelles) la qualité de service [12].

La découverte des Web services est conçue comme un problème de rechercher et de sélectionner des services pour répondre aux besoins d'utilisateur, l'importance de cette sélection doit s'adapter aux préférences de l'utilisateur.

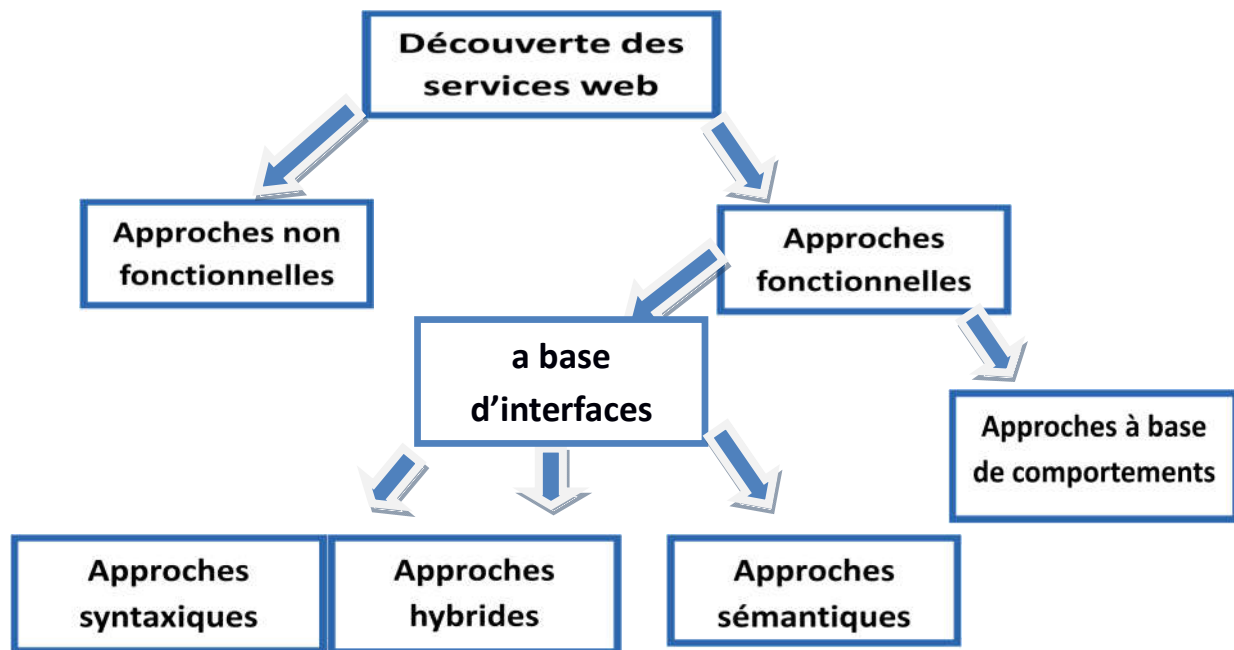
IV.1.Définition

Selon Keller «la découverte est la localisation automatique des services répondant à une requête utilisateur » [13].

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 [14].

IV.2.Approche de découverte

Il existe plusieurs approches de découverte des services web :



Parmi ces approches nous définissons l'approche sémantique.

Le web sémantique est vu comme la solution à l'exploitation pertinente et automatique des informations. L'objectif est d'offrir à la machine la possibilité de traiter une grande masse d'informations de façon non ambiguë et interprétable automatiquement. Tel que différents Web services peuvent offrir une différente fonctionnalité mais prendre les mêmes paramètres d'entrées et des sorties. Pour cette raison le langage est apparu sous forme de SAWSDL [15], la spécification d'annuaires de service UDDI, WSMO et WSDL-S [16].

OWL-S: C'est une ontologie² utilisée pour décrire la description de service web sémantiquement qui se construit naturellement sur OWL. Il a été normalisé par W3C. C'est un (successeur de DAML+OIL [17])

2

« Une ontologie est une spécification formelle et explicite d'une conceptualisation partagée » [21]

Le but d'OWL-S est de permettre l'automatisation de la recherche, la découverte, l'invocation et de l'interconnexion des services web . Les services peuvent recevoir des spécifications sémantiques plus riches.

SAWSDL: Semantic Annotations for WSDL and XML Schéma (SA-WSDL) [18] est une recommandation W3C, elle ajoute des extensions au standard WSDL, elle permet d'utiliser tous types d'ontologies (OWL, WSMML et UML). L'objectif de SAWSDL est de définir comment une annotation doit être réalisée. Il présente deux sortes d'annotations sémantiques une pour identifier les concepts sémantiques représentés par l'attribut model Reference et la deuxième pour faire le lien entre les concepts et le fichier WSDL référencés par les attributs lifting Schema Mapping et loweringSchema Mapping.

WSDL-S: (WebServices DescriptionLanguage): un des standards du corpus Services web. Il sert à décrire et publier le format et les protocoles d'un service web de manière homogène par l'utilisation du format XML. Il est chapeauté par W3C.

WSMO: (Web Service Modeling Ontology) est une ontologie introduite par [19], elle définit les éléments de modélisation pour la description de différents aspects des services web sémantiques. Il est organisé autour de quatre éléments principaux: les services web, les objectifs, les ontologies, et les médiateurs, chacun d'eux est décrit avec un formalisme basé sur le langage WSMML5 [22]. Ce modèle permet de réaliser un couplage faible entre les services web en utilisant un ensemble de médiateurs. Ces derniers assurent les tâches d'intégration d'ontologies, de découverte des services, de composition [12].

V. Conclusion

Dans ce chapitre, nous avons présenté l'architecture orientée services, c'est une nouvelle vision qui répond d'une manière efficace aux problèmes des systèmes d'information.

SOA utilise un ensemble de services assurant une fonctionnalité particulière et accessible via son interface. SOA permet la communication d'une manière transparente aux applications du système d'information, elle est capable de répondre rapidement aux nouveaux besoins d'un utilisateur. Ainsi les services Web qui sont considérés comme les composants de base les plus utilisés, ils permettent d'intégrer,

de gérer et d'automatiser plus facilement et plus rapidement, parce que les informations sont échangées au format XML.

Nous avons exposé les différentes technologies adjacentes aux services web, leurs limites en termes de découverte et au niveau de leur interopérabilité sémantique.

Chapitre: II

Conception et Implémentation du prototype

I. Introduction

Dans ce chapitre nous présentons l'approche utilisée pour la découverte des services web, en se basant sur la probabilité dominance et le classement des services offerts selon les cinq mesures de similarité. Ces méthodes peuvent nous aider à sélectionner les meilleurs services qui répondre aux besoins d'utilisateurs.

Tout d'abord nous présentons les corpus utilisés, le prototype implémentant l'environnement de développement, Après nous montrons les différentes expérimentations menés, puis, on discutera les résultats obtenus.

II. Présentation du Corpus

Nous avons utilisé dans notre approche une collection de test qui était développée par le centre allemand pour la recherche en intelligence artificielle (<http://www.dfki.de/scallops>). C'est le corpus OWLS-TC version 2.2.

Ce corpus décrit un ensemble de services web à travers des documents OWLS, il dispose de 1007 services web segmentés en 07 classes :

- Le domaine militaire.
- Le domaine de nourriture.
- Le domaine de voyages.
- Le domaine de communication.
- Le domaine d'économie.
- Le domaine médical.
- 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: has input » et « profile: has output » ces derniers sont employés comme entrées pour le module de découverte des services web.

Chaque service web (i.e. document OWLS) est étiqueté manuellement par des experts humains comme étant relevant ou non par rapport à une requête donnée. En d'autres termes, chaque service web possède une étiquette binaire (relevant ou non) par rapport à une requête donnée. Ceci permet le calcul des rappels et des précisions de l'approche proposée. 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 vidéos media par titre.

Concepts d'entrée: car.

Concepts de sortie: price.

➤ Service

Ce service donne les prix des livres.

Nom de service: book_price_service.

Concepts d'entrée: livre.

Concepts de sortie: prix.

III. Les mesures de similarité

Nous utiliserons dans cette mémoire les mesures des similarités les plus performances [24], nous citons-en:

III.1 Extended Jaccard(EJ) X_H^I

$$S(x_i, x_j) = \frac{x_i^T \cdot x_j}{\|x_i\|^2 + \|x_j\|^2 - x_i^T \cdot x_j}$$

Tel que :

S : retourne une valeur de mesure de similarité

x_i : c'est le service. (Chaque dimension de X_i représente une fréquence d'un concept)

x_j : c'est la requête.

$x_i^T \cdot x_j$: c'est le nombre des objets communs.

III.2 Cosinus (cos)

$$\cos(\vec{x}, \vec{y}) = \frac{\sum_{j=1}^n x_j \times y_j}{\sqrt{\sum_{j=1}^n x_j^2} \times \sqrt{\sum_{j=1}^n y_j^2}}$$

III.3 Loss of information (LOI)

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

Avec : $X \in \{In, Out\}$.

PC_R et PC_S : ensemble des concepts Input/output de la requête 'R' et le service 'S'.

III.4 Jensen-Shannon (JS) :

On définit la quatrième mesure de similarité comme suit :

$$JS(Pir, Pis) = \frac{1}{2\log(2)} \sum_{K=1}^{|P_{ir}|} [h(Pir(K)) + h(Pis(k)) - (Pir(k) + Pis(k))]$$

Sachant que :

Pir est la distribution des probabilités associée aux concepts d'entrées de la requête.

Por est la distribution des probabilités associée aux concepts de sorties de la requête.

Pis est la distribution des probabilités associée aux concepts d'entrées du service.

Pos est la distribution des probabilités associée aux concepts de sorties du service.

III.5 Logique (log) :

Elle offre cinq (05) scores: Exact match, Plug-in match, Subsume match, Subsumed-by match et Fail, par exemple pour comparer l'ensemble des concepts de sorties de la requête RQout avec l'ensemble des concepts de sorties du service note ASout, on applique:

$$LOGIC(RQout, ASout) = MIN_{PieRQout}(\logMatch1((P_i, ASout)))$$

$$GMatch1(P_l, ASout) = MIN_{PieRQout}(\logMatch1((P_l, ASout)))$$

$$\begin{aligned} LogMatch(P_u, P_t) = & \quad 0.95(plugin), if P_u \text{ is parent of } P_t \\ & \quad 0.85(subsume), if P_t \in P_u \\ & \quad 0.75(subsumed by) if P_t \text{ is a parent of } P_u \\ & \quad 0(fail), otherwise \end{aligned}$$

IV.L' approches utilisée

Nous avons présenté l' algorithme utilisé pour la résolution de notre problématique pour découvrir les meilleurs services web.

« vote exhaustif “Coombs”».

IV.1.L'approche Coombs

Cette méthode consiste à mettre le service le plus rejeté (qui est classé dernier dans la majorité des mesures de similarité) dans la queue de la liste fusionnée.

En cas d'équivalence , nous choisirons celui qui a la somme de scores la plus faible.

IV .1.1 algorithme :

```
/** N: nombre de services.  
M : mesures de similarité */  
Entrée :  
Liste1 ,liste2,.....listem.  
Sortie :  
Liste fusionnée.  
Début  
Charger la dataset  
Pour i ← 1 à N faire  
Boolean elimine =true  
Pour j ← 1 à M faire  
Si score (i,j) > 0 alors  
elimine=false ;  
Fin si  
Fin pour  
Si (elimine =true)  
Pour i ← 1 à M faire  
ListeL = ListeL - {Si}  
Fin pour  
Fin pour.  
Pour i ← 1 à N faire  
Condidat1=dernier (1)  
Condidat2=dernier(2)
```

Condidat3=dernier(3)

Condidat4=dernier(4)

Condidat5=dernier(5)

S=max_occurrence(condidat1 , condidat2 , condidat3 , condidat4 , condidat5)*

Liste fusionnée=ajouter(s,N-i+1 , Liste fusionnée)*

Fin pour

Fin

IV .1.2 Les fonctions utilisées

«**max_ occurrence**» : sélectionner l'élément le plus fréquent dans la dernière position de la liste .

«**dernier**» : sélectionner le dernier élément des liste qui apparaît plus.

«**éliminer**» : éliminer l'élément fusionne.

IV .1.3 Déroulement de l'algorithme

❖ *Etape1*: on cliquons le bouton « *sélection* » de nombre illimité des fichiers .

❖ *Etape 2* : on cliquons le bouton « *éliminer* » pour supprimer les services de score nul dans tous les mesures de similarité (fichier sélectionné).

Observation : Nous avons utilisés les thread par fichier avec synchronisation pour accélérer le processus.

❖ *Etape 3*: les fichiers résultats dans l'étape 2 sont utilisées par algorithme Coombs.

Algorithme Coombs qui permettre

A. Sélectionner les derniers service de chaque fichier (les listes d'entrées).

B. Sélectionner le service le plus apparu en cas d'égalité en choisie le premier .

C. Insérer le service sélectionné dans l'étape B dans la liste résultat(fichier XML résultat) et supprimer le de tous listes d'entrées.

D. Répéter les étapes A / B/ C jusqu'à les listes d'entrées deviennent vides.

❖ *Etape 4* : après l'obtention d'un fichier résultat on sélectionné une requête et on calcule les VP,FP ,FN pour top 10 ,top 20 ,top30,top40,top50,.

❖ *Etape 5* : calculer les rappelles et les précisions.

❖ *Etape 6* : répéter les étapes 4 et 5 pour 27 requêtes.

Note : le résultat obtenu de l'étape 6 une matrice où les lignes sont les requêtes et les colonnes sont rappelles/précisions du top 10 ,top 20,top 30,top 40,top 50.

❖ **Etape 7 :** calculer la moyenne de 29 requêtes de chaque colonne (top 10 ,top 20,top 30,top 40,top 50).

❖ **Etape 8 :** représenter graphiquement la matrice (rappelles/précisions).

IV.1.4.Exemple d'application

Etant donnés cinq services suivants: {S1, S2, S3, S4 et S5}

$||\text{Dataset}||=5$, M : le nombre de mesure de similarité est égale 3.

S1 {0.97, 0.91, 0.78}

S2 {0.91, 1, 0.71}

S3 {1, 0.95, 0.96}

S4 {0.99, 0.89, 0.83}

S5 {0.90, 0.80, 0.70}

Donc on applique l'algorithme « Coombs », on obtient les tableaux suivants au dessous.

Chaque tableau décrit une étape de processus de traitement.

Mesure de similarité 1	Mesure de similarité 2	Mesure de similarité 3	résultat
S3	S2	S3	
S4	S3	S4	
S1	S1	S1	
S2	S4	S2	
S5	S5	S5	

Tableau II.1

❖ Tableau I , contient classement initial des services selon les 3 mesures de similarité

Mesure de similarité 1	Mesure de similarité 2	Mesure de similarité 3	résultat
S3	S2	S3	
S4	S3	S4	
S1	S1	S1	
S2	S4	S2	
			S5

Tableau II.2

❖ Dans tableau II,on ajoute S5 dans fichier résultat et on le supprime de trois listes.

Mesure de similarité 1	Mesure de similarité 2	Mesure de similarité 3	résultat
S3	S3	S3	
S4	S1	S4	
S1	S4	S1	
			S2
			S5

Tableau II.3

❖ Dans *Tableau III* ,on ajoute S2 dans fichier résultat et on le supprime de trois listes .

Mesure de similarité 1	Mesure de similarité 2	Mesure de similarité 3	résultat
S3	S3	S3	
S4	S4	S4	
			S1
			S2
			S5

Tableau II.4

❖ Dans *Tableau IV*, on ajoute S1 dans fichier résultat et on le supprime de trois listes .

Mesure de similarité 1	Mesure de similarité 2	Mesure de similarité 3	résultat
S3	S3	S3	
			S4
			S1
			S2
			S5

Tableau II.5

❖ Dans *Tableau V*, on ajoute S4 dans fichier résultat et on le supprime de trois listes .

résultat
S3
S4
S1
S2
S5

Tableau II.6

❖ Dans *Tableau VI* on obtient résultat final d'exécution de l'algorithme Coombs.

IV.2.Présentation du prototype

IV.2.1 Outils et environnement de développement

Avant l'implémentation de notre application, nous allons tout d'abord spécifier les outils utilisés qui nous ont semblés être un bon choix vu les avantages qu'ils offrent, Notant que RAD Studio XE10.1 UP2 (Berlin) est le choix des développeurs qui recherchent la puissance, la lisibilité et la flexibilité du langage Object Pascal moderne, orienté objet couplé avec des compilateurs natifs et des bibliothèques de composants pour le développement rapide de code source unique sous Windows, macOS, iOS, Android et Linux.

Nous décrivons les expériences permettant l'analyse des performances de l'approche Coombs Nous avons utilisé 29 requêtes pour évaluer la qualité des approches (précision, rappel). L'algorithme est implémenté en RAD Studio XE10.1, et les expériences ont été

menées sur une machine Intel(R) Core I7 avec 6 Go de la RAM, sous le système d'exploitation Windows 7,64bit.

IV.2.2 présentation du prototype

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



Figure II.1 : Interface d'accueil



Figure II. 2 : Interface d'élimination des services de score nul et algorithme de Coombs



Figure II. 3 : exemple d'ex cution de la fen tre Pr c dente (Figure II.2)



Figure II. 4 calule de rappel et pr cision

TOP	Rappel	Pr�cision	Requette
Top 10	100	26,32	governmentmissile_funding_service.owls.xml
Top 20	95	50	governmentmissile_funding_service.owls.xml
Top 30	90	71,05	governmentmissile_funding_service.owls.xml
Top 40	75	78,95	governmentmissile_funding_service.owls.xml
Top 50	62	81,58	governmentmissile_funding_service.owls.xml
Top 60	58,33	92,11	governmentmissile_funding_service.owls.xml
Top 10	80	61,54	surfingorganization_destination_service.owls.xml
Top 20	55	84,62	surfingorganization_destination_service.owls.xml
Top 30	43,33	100	surfingorganization_destination_service.owls.xml
Top 40	32,5	100	surfingorganization_destination_service.owls.xml

TOP	Rappel moy�	Pr�cision mo
TOP10	86,67	37,37
TOP20	67,5	55,24
TOP30	60,56	70,32
TOP40	51,25	76,73
TOP50	46,67	84,23
TOP60	44,44	93,18

Figure II. 5 affichage des r sultats (rappels et pr cisions et moyen)

Algorithm	Top 10	Top 20	Top 30	Top 40	Top 50	Top 60
Ej	0.33	0.59	0.73	0.79	0.83	0.85
Il	0.33	0.59	0.74	0.79	0.84	0.86
Js	0.33	0.59	0.72	0.79	0.83	0.86
Logic	0.3	0.46	0.6	0.66	0.72	0.69
Cos	0.33	0.59	0.72	0.78	0.82	0.86
Coombs	0.39	0.61	0.75	0.80	0.85	0.90

Tableau des rappels des approches : II. 7

Algorithme	Top 10	Top 20	Top 30	Top 40	Top 50	Top 60
Ej	0.81	0.64	0.58	0.51	0.42	0.36
Il	0.81	0.64	0.58	0.5	0.42	0.36
Js	0.8	0.65	0.57	0.49	0.41	0.36
Logic	0.73	0.53	0.48	0.42	0.37	0.31
Cos	0.81	0.64	0.57	0.48	0.4	0.36
Coombs	0.82	0.67	0.58	0.47	0.41	0.37

Tableau des précisions des approches : II. 8

Selon la table des précision, on remarque que notre approche de fusion est plus performante que les cinq mesures de similarités dans plus de 66 % des cas, en parallèle, la table des rappels montre que la méthode Coombs est largement meilleure que les scores des mesures de similarités pour tous les TopK possibles (de 10 jusqu'à 60), et ceci montre l'efficacité de cette approche de vote dans la découverte des services.

• **Précision:** Elle mesure la proportion des services pertinents relativement à l'ensemble des services rendus par le système pour une requête elle est exprimée par :

$$\textit{Précision} = \textit{vp}/(\textit{vp} + \textit{fp})$$

Avec :

Vp: les vrais positifs. (Un résultat correct, et considéré comme juste par le système).

Fp: les faux positifs. (Un résultat erroné, mais considéré comme juste par le système).

• **Rappel:** Elle mesure le pourcentage des services pertinents retournés parmi tous les services pertinents rendus par le système pour une requête ; elle est exprimée par :

$$\textit{rappel} = \textit{Vp}/(\textit{Vp} + \textit{Fn}).$$

Vp: les vrais positifs. (Un résultat correct, et considéré comme juste par le système).

Fn: les faux positifs. (Un résultat erroné, et considéré comme faux par le système).

• **R-précision:** Une mesure communément utilisée est la précision exacte ou R-précision. Précision après que R premiers services ont été retrouvés, où R est le nombre de documents pertinents pour la requête considérée.

IV.4. Conclusion

Dans ce chapitre nous avons conçu et implémenté un prototype permettant de réaliser l'approche «Coombs ou vote exhaustif ».

Coombs : a pour objectif de fusionner un ensemble de listes de services web, en particulier elle permet de mettre les services les plus rejetés dans la queue de la liste globale (ou fusionnée)(**Figure II. 3**).

Conclusion générale

Conclusion générale

La tâche de découvrir les services web devient de plus en plus difficile à cause de l'évolution rapide des nombres des services sur le réseau internet.

Le travail présenté dans ce mémoire tourne autour de la recherche des meilleurs services qui répondent au mieux à une requête –utilisateur (formée principalement par les concepts d'entrées/sorties), pour cela, Nous avons utilisé méthode de fusion de plusieurs mesures de similarités basée sur la règle de Coombs ou vote exhaustif .

Nous avons agrégé cinq algorithmes de matching (Cosinus, Extended Jaccard, Jenson-Shanon, perte d'informations, et l'approche logique). Et les résultats obtenus sont plus que satisfaisants.

Comme perspectives à ce travail nous proposons l'utilisation de plusieurs schémas de fusion, qui peuvent se baser sur la théorie de probabilité, l'apprentissage automatique, le jugement majoritaire....

Références bibliographiques

Références bibliographique

- [17] Ankolekar Anupriya; Steffen Lamparter et al; preference -based selection of highly configurable web services; 16th international conference on the world wide web ; USA New York ; 2007.
- [10] Berners Lee; Hendler et al; The Semantic Web; Scientific American; 2001.
- [21] Borst Pim; Akkermans Hans et al; Engineering ontologies; International Journal of Human-Computer Studies; vol. 46; 2-3; 1997.
- [14] David Booth, Hugo Haas, Francis McCabe, et al; Web services architecture, W3C Working Group Note 11; In <http://www.w3.org/TR/ws-arch/>; 2004.
- [9] David Chappell et Tyler Jewell; Java Web Services; O'Reilly; March 2002
- [18] Farrell Joel; Lausen Holger; Semantic Annotations for WSDL and XML Schema; W3C Recommendation; <https://www.w3.org/TR/sawSDL/>; 28 August 2007.
- [2] Fournier-Morel, Pascal Grojean et al. Cyril Rognon; SOA, le guide de l'architecte de SI ; Dunod; Paris ; 2006, 2008.
- [1] Grofman Bernard., & Feld Scott L. (2004). If you like the alternative vote (aka the instant runoff), then you ought to know about the Coombs rule. *Electoral studies*, 23(4), 641-659.
- [12] hadjila FethAllah Composition et interopération des services web sémantiques ; Thèse de Doctorat en informatique. Université Tlemcen; 2014.
- [4] Julie 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.
- [8] Karupiah Tamilarasi ; M. Ramakrishnan; indexing traditional UDDI for efficient discovery of web services; Indian Journal of Computer Science and Engineering (IJCSE); Vol. 6 ; 1 Feb-Mar 2015.
- [3] Kellert Patrick et Toumani Farouk; Les Web Services sémantiques ; Revue I3 Information-Interaction-Intelligence ; France ; 2004.
- [1] Krafzig Dirk; Banke Karl; et al: Enterprise SOA: Service-Oriented Architecture Best Practice; Prentice Hall PTR; November 2006.
- [19] Lausen Holge ; Polleres Axel et al; Web Service Modeling Ontology (WSMO); In W3C Member Submission; <https://www.w3.org/Submission/WSMO/>; 2 June 2005
- [15] Martin David; Burstein Mark et al; OWL-S: Semantic Markup for Web Services, 22 November 2004.



[24]Matthias Klusch; Benedikt Fries et al; Sycara.Automated semantic web service discovery with OWLS-MX;AAMAS ;Hakodate; Japan 2006.

[6]MELLITI Tarek; Interopérabilité des services Web complexes. Application aux systèmesmulti-agents, thèse de doctorat ; Université Paris IX Dauphine ; 2004

[9]Newcomer Eric ; undersetnding web service - XML, WSDL, SOAP and UDDI.independent technologies guides; 2002.

[11]Rohallah Benaboud1; RamdaneMaamri et Zaidi Sahnoun; agents and owl-s based semantic web service discovery with user preference support; international Journal of Web & Semantic Technology (IJWesT) Vol.4, 2, April 2013

[22] Roman Dumitru ; Lausen Holger et al; D2v1.4. Web Service Modeling Ontology (WSMO); WSMO Working Draft;
<http://www.wsmo.org/TR/d2/v1.4/20070216/>; 16 February 2007

[13]Uwe Keller, Ruben Lara, Axel Polleres, Ioan Toma, Michael Kifer et Dieter Fensel; Wsmo web service discovery; WSMO Working Draft November 2004.

[16]Verma Kunal; Akkiraju Rama; et al; semantic matching of web service policies; second international workshop on semantic and dynamic web processes;P 79 -90; Orlando Florida; 2005.

[5]Veziane Arnaud ; les services web— Présentation générale ; Version 0.0.1 ; Association HE.R.M.E.S, Château du Montet, 2005.

Résumé :

La découverte des services web constitue un problème primordial de l'architecture orienté service .

Nous proposons dans ce travail une solution basée sur la théorie de vote en particulier la méthode de Coombs .

Les résultats montrent que cette méthode de fusion est plus performante que les mesures de similarités d'entrées.

Mot clés : services web, découverte de services , méthodes de vote , règle de Coombs.

Abstract :

The web service discovery is one competing era in this work , we propose a voting model based on the Coombs rule.

The experimental evaluation shows that this fusion method is more effective than the individual similarity measures.

Key words: web services, service discovery, Coombs method.

ملخص:

يعتبر البحث عن خدمات الواب من بين المشاكل الرئيسية للحساب عبر الخدمات .
نقترح عبر هذه المذكرة طريقة دمج تعتمد على الاقتراع بواسطة Coombs.
النتائج المحصل عليها تؤكد نجاعة هذه المقاربة في حل المشكلة .

كلمات مفتاحية: خدمات الواب , البحث عن الخدمات , قاعدة Coombs.