

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE  
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE  
SCIENTIFIQUE

UNIVERSITÉ IBN KHALDOUN TIARET



Faculté des Mathématiques  
et de l'Informatique  
Département Informatique



## MÉMOIRE

Pour l'obtention du Diplôme de Master en Informatique  
Option : Génie Logiciel (GL)

**Présenté par :**

Mr. ABDI Youcef Ali

Mr. BERRAHO Boubakeur Seddik

---

# VERS UNE APPROCHE SOCIALE DE DÉCOUVERTE DES SERVICES WEB SÉMANTIQUE

---

Mémoire dirigé par :

Mr. Hadj MEGHAZI

Maître assistant à l'université Ibn Khaldoun

Soutenu en 2016



# Remerciement

Nous remercions notre dieu **ALLAH** qui nous a donné la santé, le courage, la force et la patience d'achever ce modeste travail.

Avec beaucoup de respect et d'estime que nous remercions notre encadreur monsieur **MEGHAZI** d'avoir proposé, dirigé et suivi ce travail.

Nous tenons aussi à remercier nos collègues pour leurs précieux conseils et leur disponibilité durant toute la période de notre projet.

Nous tenons à adresser nos remerciements à tous les enseignants de département d'informatique.

# Dédicace

A

- *Mes chers parents*
- *Mes chers professeurs*
- *Mes frères*
- *Mes amis*

***ABDI YUCEF ALI***

# *Dédicace*

*Je dédie ce modeste travail et ma profonde gratitude à*

*Ma chère mère pour l'éducation qui l mon prodigue  
avec tous les moyens et au prix de toutes les sacrifices  
qu'ils consentis à mon égard.*

*à mes chères sœurs et mes frère ouadah et Djilali en  
leurs espérant le plein succès dans leur vie.*

*Et toutes mes amies qui m ont toujours encouragée*

*Berraho boubakeur seddik.*

## Sommaire

Résumé.....

Introduction générale.....

Chapitre I .....

Des services web syntaxique aux services web sémantique

.....**Erreur ! Signet non défini.**

1. Introduction..... **Erreur ! Signet non défini.**

2. Les services web ..... **Erreur ! Signet non défini.**

2.1. Qu'est-ce qu'un service web ? ..... **Erreur ! Signet non défini.**

2.2. Architecture des services web..... **Erreur ! Signet non défini.**

2.2.1. Le rôle d'acteurs des services web ..... **Erreur ! Signet non défini.**

2.2.2. La pile des protocoles des services web..... **Erreur ! Signet non défini.**

3. Les services web sémantiques ..... **Erreur ! Signet non défini.**

3.1. L'architecture des services web sémantiques..... **Erreur ! Signet non défini.**

3.2. Descriptions sémantiques des services web ..... **Erreur ! Signet non défini.**

3.2.1. DAML-S ..... **Erreur ! Signet non défini.**

3.2.2. OWL-S..... **Erreur ! Signet non défini.**

3.2.3. WSDL-S ..... **Erreur ! Signet non défini.**

4. Synthèse.....21

Chapitre II .....

Les services web

sociaux.....**Erreur ! Signet non défini.**

1.Introduction ..... **Erreur ! Signet non défini.**

2.Définition et fonctionnement global d'un réseau social de services WEB**Erreur ! Signet non défini.**

2.1. La collaboration : ..... **Erreur ! Signet non défini.**

4.Développement des services du web social..... **Erreur ! Signet non défini.**

Étape 1: Identification des composants du réseau social ..... **Erreur ! Signet non défini.**

Étape 2 : correspondances des analyses des services web ..... **Erreur ! Signet non défini.**

Étape 3 : gestion du réseau social ..... **Erreur ! Signet non défini.**

Étape 4: Évaluation du poids d'arête initial ..... **Erreur ! Signet non défini.**

Étape 5 : naviguer dans les réseaux sociaux ..... **Erreur ! Signet non défini.**

|  |                                    |
|--|------------------------------------|
| Etape 6 : l'évaluation continue des poids des arcs : .....             | <b>Erreur ! Signet non défini.</b> |
| 4. Du Data Mining sur les réseaux sociaux .....                        | <b>Erreur ! Signet non défini.</b> |
| 4.1 Data Mining dans l'analyse des sentiments et de l'opinion publique | <b>Erreur ! Signet non défini.</b> |
| 6. Le tagging dans les réseaux sociaux .....                           | <b>Erreur ! Signet non défini.</b> |

### Chapitre III .....

#### conception et implémentation

|   |                                    |
|---|------------------------------------|
| .....   | <b>Erreur ! Signet non défini.</b> |
| 1. Introduction.....  | <b>Erreur ! Signet non défini.</b> |
| 2. Architecture du réseau social de Web services.....                               | <b>Erreur ! Signet non défini.</b> |
| 2.1. Couche 1 : Annotation et recherche sémantique .....                            | <b>Erreur ! Signet non défini.</b> |
| 2.2. Couche 2 : Découverte sociale .....  | <b>Erreur ! Signet non défini.</b> |
| 3. Architecture globale .....   | <b>Erreur ! Signet non défini.</b> |
| 4. Implémentation et mise en œuvre .....  | <b>Erreur ! Signet non défini.</b> |
| 4.1. Résumé .....   | <b>Erreur ! Signet non défini.</b> |
| 4.2. Fonctionnement globale .....   | <b>Erreur ! Signet non défini.</b> |
| 4.2.1. Découverte syntaxico-sémantique .....  | <b>Erreur ! Signet non défini.</b> |
| 4.2.2. Découverte sociale .....   | <b>Erreur ! Signet non défini.</b> |
| 4.3. Scénario de fonctionnement du réseau social .....                              | <b>Erreur ! Signet non défini.</b> |
| 4.4. Analyse des résultats et comparaison entre la découverte sociale et sémantique | <b>Erreur ! Signet non défini.</b> |
| CONCLUSION GENERALE.....  |                                    |

# LISTE DES FIGURES

---

|   |                                    |
|---|------------------------------------|
| Fig. 1.1 Evolution de l'usage du web.....   | <b>Erreur ! Signet non défini.</b> |
| Fig. 1.2 Communication XML pour les services web.....   | <b>Erreur ! Signet non défini.</b> |
| Fig. 1.3 Modèle d'interactions des services web.....  | <b>Erreur ! Signet non défini.</b> |
| Fig. 1.4 La pile des protocoles pour un service web.....  | <b>Erreur ! Signet non défini.</b> |
| Fig. 1.5 L'évolution du web.....  | <b>Erreur ! Signet non défini.</b> |
| Fig. 1.6 Pile des services web sémantiques.....   | <b>Erreur ! Signet non défini.</b> |
| Fig. 1.7 Le model conceptuel d'OWL-S.....   | <b>Erreur ! Signet non défini.</b> |
| Fig. 1.8 L'externalisation de la représentation et de l'association de la sémantique aux éléments de WSDL[4]..... | <b>Erreur ! Signet non défini.</b> |
| Figure 2.1 – Scénario de fonctionnement d'un réseau social de Web services  | <b>Erreur ! Signet non défini.</b> |
| Figure 2.2 gestion du réseau social.....  | <b>Erreur ! Signet non défini.</b> |
| Figure 2.3 - Fonctionnement de BlogVox.....   | <b>Erreur ! Signet non défini.</b> |
| Figure 3.1 – Inputs Outputs dans le fichier profile OWL-S.....  | <b>Erreur ! Signet non défini.</b> |
| Figure 3.2 – Description textuelle dans le fichier profile OWL-S.....   | <b>Erreur ! Signet non défini.</b> |
| Figure 3.3 – Interface principale.....  | <b>Erreur ! Signet non défini.</b> |
| Figure 3.4 – Liste des Web services de la collection.....   | <b>Erreur ! Signet non défini.</b> |
| Figure 3.5 – communauté des services Web substituteurs issue de la découverte sémantique                          | <b>Erreur ! Signet non défini.</b> |
| Figure 3.6 – communauté des services Web substituteurs issue de la découverte sociale                             | <b>Erreur ! Signet non défini.</b> |
| Figure 3.7 – mises à jour des paramètres sociales dans la base de données ..                                      | <b>Erreur ! Signet non défini.</b> |
| Figure 3.8 – comparaison entre les résultats retournés par la découverte sémantique et la découverte sociale..... | <b>Erreur ! Signet non défini.</b> |



## *RÉSUMÉ*

Quand il s'agit de services Web, le problème qui revient le plus est toujours quelle est la meilleure approche pour permettre une découverte pertinente de ces services à la requête du client, dans cet axe plusieurs approches syntaxiques et sémantiques ont été proposées, Néanmoins il aurait été intéressant de focaliser sur le comportement social de ces services entre eux, voir qui remplace qui en cas de panne, identifier les paires pertinentes à la composition, voir les paires en compétition. Ce travail a pour but de présenter les étapes de création d'un réseau social de services Web sémantiques pour capter l'interaction sociale entre ces derniers, ensuite nous allons voir comment la découverte sociale complète la découverte sémantique et accentue sa pertinence, Enfin, nous allons essayer d'extraire des communautés de services Web similaires en analysant les données collectées à partir du réseau social

Mots clés : service web, Réseaux sociaux, Services Web sémantiques, Réseaux sociaux de services Web, Découverte sociale.

---

# *Introduction Générale*

---

## ***Introduction générale :***

Depuis que les hommes existent, les réseaux sociaux ont été présent, des groupes sociaux se sont formés a propos de divers thèmes tels que la religion, les arts ou encore la vie de tous les jours. Ces réseaux informelles existent dès que deux personnes ou plus partage leurs avis. Tout au long de l'histoire, ces réseaux se sont multipliés et diversifié, les hommes ont toujours cherché a se réunir pour partager leurs opinions. A notre époque, on voit l'apparition de réseaux sociaux a caractères personnel tels que les agences de rencontrent mais aussi a caractères professionnels avec les agences de renseignements. Avec le développement d'Internet, les réseaux sociaux sont devenus la technologie phare du *Web 2.0* ils permettent de capter le comportement sociale des éléments adhérant, localisent des entités spécifiques, tracent les comportements sociaux et offrent un énorme tat de données, ces données une fois analysées permettront d'extraire des communautés de personnes qui partagent tel ou tel comportement, idéologie, passion, religion, ou n'importe quel autre paramètre sociale. En parallèle, la tendance actuelle de la communauté des chercheurs est d'exploiter les technologies du Web sémantique, afin d'enrichir les services Web de descriptions sémantiques. La combinaison des technologies des services Web et du Web sémantique convergent vers des services Web sémantiques. Néanmoins, malgré les diverses approches sémantiques proposées pour palier au problème de la pertinence lors de la découverte des services Web, un élément essentiel est négligé, cet élément n'est autre que le comportement sociale dans le temps de ces entités dites services Web. Pour accentuer la pertinence de la découverte des services Web, il est impératif de prendre en considération l'interaction sociale entre les services Web sémantique. Dans cet axe, plusieurs approches ont été proposées pour la création de réseaux sociaux de services Web, nous avons suivi l'une d'entre elles, la plus aboutie, et nous sommes arrivés à implémenter les mécanismes de base qui nous permettent de dégager des corpus qui peuvent servir comme banc d'essai pour une éventuelle étude qui porte sur la détection des communautés cachées dans les réseaux sociaux des services web .

Comme organisation de notre mémoire on parle dans le chapitre I de dépassement du service web syntaxique aux services web sémantique, concernant le chapitre II les service web sociaux ou nous présentons les étapes de création d'un réseau social de services Web sémantiques pour capter les interactions sociales entre ces derniers, enfin le dernier chapitre la conception et l'implémentation.

---

# CHAPITRE I

# CHAPITRE I

---

## 1. Introduction

Le web sémantique transforme le web actuel, le web syntaxique, en un web dont on se rend compte et on utilise la sémantique afin de surmonter le manque de compréhension et les limites dans l'intégration des structures de données et du vocabulaire.

Avec le web sémantique comme base fondamentale, il est possible que les utilisateurs humains trouvent des données pertinentes et peuvent interagir avec le web de manière sémantiquement définie et précise.

Mais en allant au delà des données et du traitement d'information, le web sémantique ne couvre pas l'introduction de l'aspect dynamique et distribué dans le web courant ; les services web est la solution qui promet de couvrir cet aspect dynamique et distribué en rendant l'infrastructure du web, un moyen pour les traitements distribués à une grande échelle.

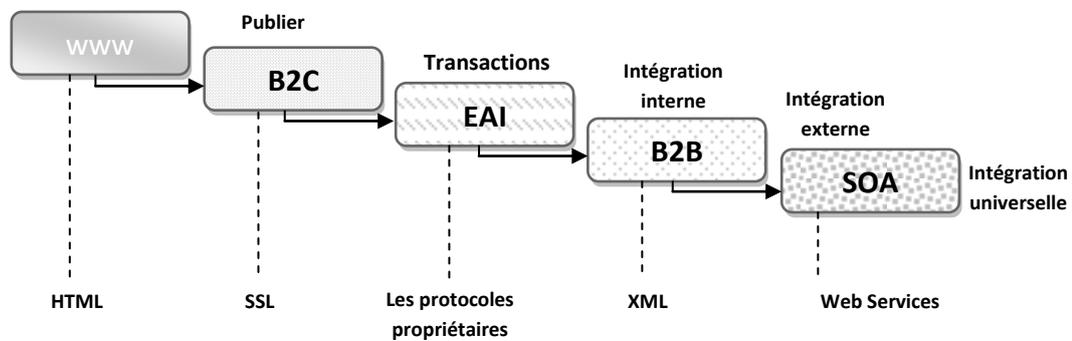
## 2. Les services web

Les services web sont basés sur le principe d'architecture orientée « service » SOA (Service-Oriented Architecture). SOA est une des dernières évolutions de l'informatique répartie, qui permet aux composants logiciels, y compris des fonctions d'applications, objets et processus de différents systèmes, d'être exposé comme services.

L'évolution a commencé quand les organisations ont voulu faire implémenter des solutions telles que le Business-to-Consumer et le Business-to-Business, ils ont réalisé que les technologies du web initiales associées au WWW ne sont pas suffisantes pour vendre des produits sur internet. En utilisant SSL, les organisations ont été en mesure d'implémenter des solutions pour avoir la confiance de leurs clients dans l'obtention des informations confidentielles (par exemple : le numéro d'une carte de crédit).

Afin de répondre aux espérances des clients et des partenaires, les organisations devraient lier leurs systèmes autonomes, hétérogènes et distribués, pour améliorer l'efficacité et la productivité, ce qui a mené au développement et au déploiement des solutions EAI (Entreprise Application Intégration). Les plateformes EAI ont été utilisées

pour intégrer les systèmes incompatibles tels que : les ERP (Entreprise Ressource Planning), les CRM (Consumer Relationship management), les bases de données, les entrepôts de données et autres systèmes interne à l'organisation.



*Fig. 1.1 Evolution de l'usage du web.*

Cependant ces plateformes (EAI) utilisaient des protocoles propriétaires qui posent des problèmes quand il s'agissait de faire intégrer des systèmes internes à l'organisation avec d'autres qui ne le sont pas.

Les systèmes internes et externes devraient communiquer à travers des réseaux permettant ainsi aux entreprises d'achever leurs transactions. Pour atteindre un tel niveau d'intégration, les solutions Business-to-Business (B2B) se sont développées au tour de XML comme langage de représentation de données.

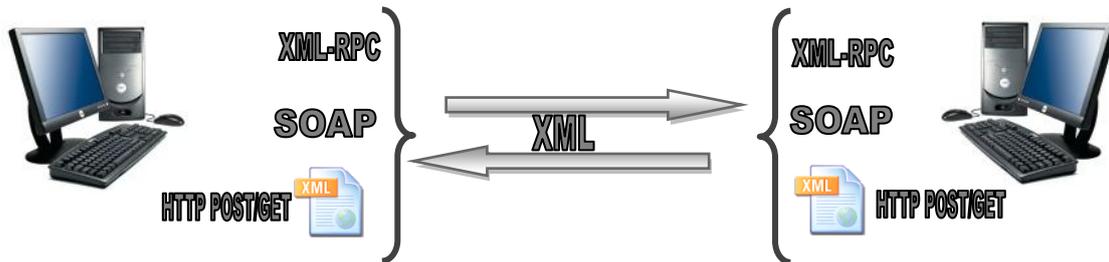
Après avoir vécu et expérimenté l'utilisation de XML, les organisations ont réalisé que leurs solutions architecturales montraient un couplage fort dans l'interaction des applications, ce qui limite la flexibilité et l'adaptation des systèmes. Le concept de l'architecture orientée services (SOA) a été présenté et a défini une méthode de concevoir, développer, déployer et gérer des portions logiques des programmes. Le but de SOA est de faiblir le couplage, structurer et standardiser les fonctionnalités d'entreprises dans les applications logiciels qui interagissent.

## 2.1. Qu'est-ce qu'un service web ?

Un service web est un service qui est disponible sur l'internet, utilise un système de communication XML standardisé et n'est pas spécifique à un système d'exploitation ou un langage de programmation[1].

Selon Gartner Research, « Les services web sont des composants logiciel faiblement couplées fournis au-dessus des technologies standardisées d'internet » [2].

Il y a différentes moyens pour effectuer une communication XML. Par exemple, on peut utiliser XML Remote Procedure Calls (XML-RPC) ou bien SOAP. Une autre alternative, est d'utiliser les méthodes GET/POST de HTTP pour faire circulé des documents XML.



*Fig. 1.2 Communication XML pour les services web*

Une des caractéristiques des services web est qu'ils sont « *auto descriptifs* », ca veut dire, que si un nouveau service est publié, il doit également avoir à publier une interface publique du service. Au minimum, le service devrait inclure une documentation lisible afin que d'autres développeurs puissent plus facilement intégrer celui-là. S'il s'agit de créé un service SOAP, il est recommander d'inclure une interface publique écrite dans une grammaire XML commune. Cette grammaire est utilisée pour identifier toutes les méthodes publiques, les arguments d'une méthode et les valeurs retournées.

Un service web complet est, donc, n'importe quel service qui :

- Est disponible à travers internet ou des réseaux privés (intranet).
- Utilise un système de communication XML standardisé.
- N'est pas lié à un système d'exploitation ou un langage de programmation.
- Est auto-descriptif via une grammaire XML commune.

## 2.2. Architecture des services web

Il y a deux manières de voir l'architecture d'un service web. La première est d'examiner les différents rôles de chaque acteur des services web ; la seconde est d'examiner la pile de protocole émergente des services web.

### 2.2.1. Le rôle d'acteurs des services web

Il y a trois rôles d'acteurs importants pour les services web :

*Fournisseur de services* : C'est le fournisseur du service web. Le fournisseur de service implémente le service et le rend disponible sur internet.

*Demandeur de services* : C'est n'importe quel consommateur du service web. Le demandeur utilise un service web existant en initiant une connexion réseau et en envoyant une requête XML.

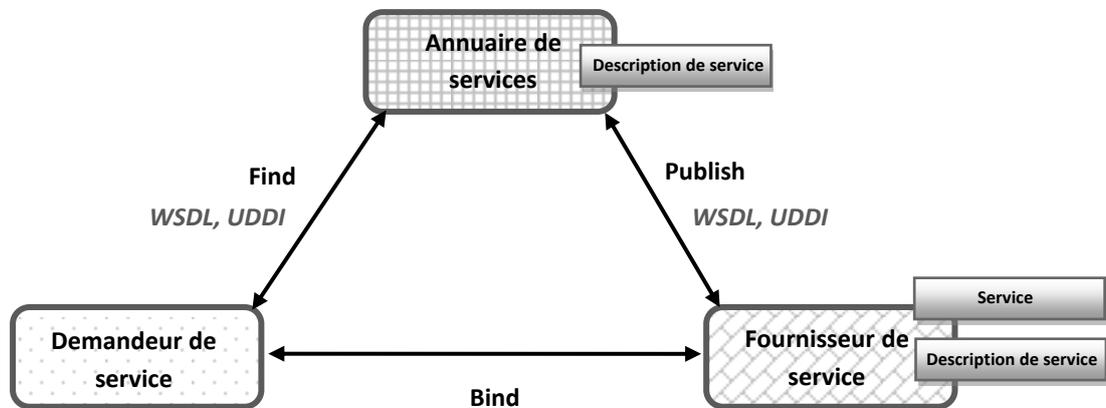
*Annuaire de services* : C'est un annuaire logiquement centralisé des services. L'annuaire fournit un endroit central où les développeurs peuvent publier de nouveaux services ou trouver ceux qui existent. Il sert donc d'un point central aux compagnies et à leurs services.

Dans un scénario de fonctionnement normal, un fournisseur de services héberge un module implémentant un ou plusieurs services web accessible via le réseau. Il définit une description du service et la publie en la faisant enregistrée dans un annuaire des services. Le demandeur de service amorce une opération de recherche pour trouver la description du service. Afin d'invoquer le service et interagir avec l'implémentation du service web, le demandeur de service établit une connexion avec le fournisseur de service.

On peut distinguer trois opérations :

- La publication des descriptions des services (*publish*).

- La recherche et la découverte de la bonne description du service (*find*).
- L'association ou l'invocation des services basés sur la description (*bind*).



*Fig. 1.3 Modèle d'interactions des services web*

### 2.2.2. La pile des protocoles des services web

Une autre manière de voir l'architecture des services web est d'examiner leur pile de protocoles. La pile évolue toujours, mais a actuellement quatre couches principales.

Voilà une courte description de chaque couche :

- **Transport**

Cette couche est responsable du transport des messages entre applications. Actuellement, cette couche inclut HyperText Transfer Protocol (HTTP), Simple Mail Transfer Protocol (SMTP), File Transfer Protocol (FTP), et des nouveaux protocoles, comme Blocks Extensible Exchange Protocol (BEEP).

- **Communication XML**

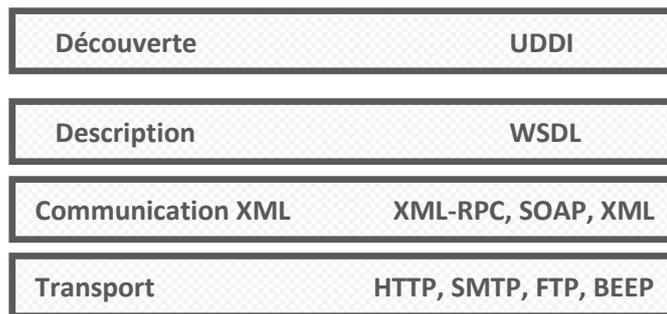
Cette couche est responsable de l'encodage des messages dans un format XML commun de telle sorte que les messages puissent être compris pour chacune des extrémités. Cette couche inclut XML-RPC et SOAP.

- **Description du service**

Cette couche est responsable de la description de l'interface publique d'un service web donné. Actuellement, la description d'un service est manipulée par l'intermédiaire du langage WSDL (*Web Service Description Language*).

- **Découverte des services**

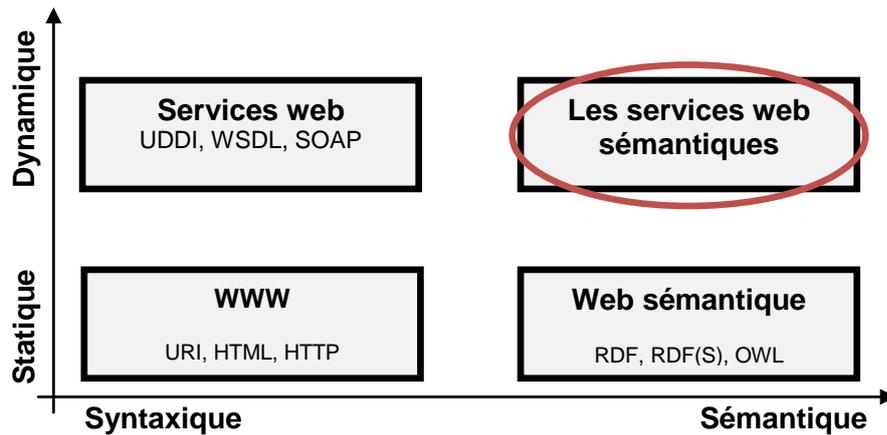
Cette couche est responsable de la centralisation des services dans un annuaire commun, et de fournir facilement des fonctionnalités de type Publish/Find (Publier/trouver). Actuellement, la découverte de services est manipulée par l'intermédiaire d'UDDI (*Universal Description, Discovery and Integration*).



*Fig. 1.4 La pile des protocoles pour un service web*

### 3. Les services web sémantiques

Le terme « services web sémantiques » est réservé pour l'automatisation des tâches d'utilisation des services, tels que : la découverte, la sélection, la composition et l'établissement des services appropriés. Les services web sémantiques se trouvent à la convergence de deux domaines de recherche importants concernant les technologies de l'internet : le web sémantique et les services web. Cette tâche est accomplie en rendant les services web auto-exploitable par machines. De la même façon que le web sémantique a promis de permettre aux machines de tirer parti du contenu statique du web en utilisant les annotations. L'idée d'appliquer des techniques semblables aux services web est très intéressante.



*Fig. 1.5 L'évolution du web*

Les avantages potentiels des services web sémantiques ont mené à l'établissement d'un domaine de recherche important, dans le milieu industriel et académique. Plusieurs initiatives sont apparues pour faire ce qu'on appelle « l'annotation sémantique des services web », ce qui a produit une variété de descriptions des services web et leurs aspects relatifs, ce qui en retour a abouti à de divers genres de supports pour la découverte et la composition.

Le besoin d'automatisation du processus de conception, de mise en œuvre et de la découverte des services web est le même que celui du web sémantique, à savoir comment décrire formellement les connaissances de manière à les rendre exploitables par des machines. En conséquence, les technologies et les outils développés dans le contexte du web sémantique peuvent compléter la technologie des services web.

Le concept fondamental du web sémantique et des SWS<sup>1</sup> (Services Web Sémantique) est l'ontologie, qui produit « une signification bien définie » des informations contenu dans le web. Une ontologie est un ensemble de termes de la connaissance, qui inclut un vocabulaire, des liens sémantiques et quelques règles simples pour un domaine donné. L'avantage principal des ontologies est que plusieurs communautés sur internet ont maintenant des définitions partagées sur un bon nombre de concepts clés. Par exemple, la notion d'ontologie peut jouer un rôle important pour permettre d'explicitier la sémantique des services facilitant ainsi les communications hommes-machines, d'une part, et d'autre part les communications machines-machines.

<sup>1</sup> SWS : Semantic web services.

L'objectif des services web sémantique est de créer un web sémantique des services web dont les propriétés, les capacités, les interfaces et les effets sont décrits de manière non ambiguë et exploitable par des machines et ce en utilisant les couches techniques sans pour autant en être conceptuellement dépendants [3]. La sémantique exprimé permet d'automatisé les fonctionnalités suivantes :

- Processus de description et de publication des services.
- Découverte des services.
- Sélection des services.
- Composition des services.
- Fourniture et administration des services (Négociation des contrats etc.).

Donc les services web ont besoin des langages et d'APIs pour ajouter cette dimension sémantique qui va leurs permettre d'avoir les fonctionnalités suivantes :

- ***Une découverte automatique des services web***

La découverte automatique d'un service web implique la localisation automatique des services web qui produisent une certaine fonctionnalité et qui répondent aux contraintes demandées. Par exemple, un utilisateur qui veut trouver un service qui vent des billets d'avion entre deux villes données et accepte une carte de crédit. Actuellement, cette tâche doit être fait par un humain qui peut utiliser un moteur de recherche pour trouver le service approprié, lire une page web et exécuter le service manuellement, pour déterminer si les contraintes sont satisfaites. Avec des langages appropriés<sup>2</sup>, les informations nécessaires pour la découverte des services web doivent être spécifiées sous forme de balises sémantiques (par exemple une utilisation d'ontologie sous forme « OWL » ou « DAML ») qui peut être interprétées par des machines au niveau du site qui héberge le service web et l'annuaire de services ou au niveau d'un moteur de recherche avancé à base d'ontologies. Ou bien, un service peut pro-activement s'annoncer auprès d'un annuaire, pour que les demandeurs peuvent le trouvé en adressant leurs requêtes à l'annuaire.

---

<sup>2</sup> Ce type de langage correspond à DAML-S et OWL-S qui vont être détaillés plus loin dans ce chapitre.

Ces langages doivent produire des annonces déclaratives des propriétés et des fonctionnalités du service qui peuvent être utilisées pour la découverte automatique du service web.

- ***L’invocation automatique des services web***

L’invocation automatique d’un service web implique l’exécution d’un service web identifié par programme informatique ou un agent. Par exemple l’utilisateur peut demander d’acheter un billet d’avion d’un site donné et d’un vol donné. Actuellement, l’utilisateur doit consulter le site web qui offre le service, remplir un formulaire et cliquer sur un bouton pour exécuter le service. Une alternative est que l’utilisateur envoie directement une requête HTTP au service avec les paramètres appropriés en HTML. Dans les deux cas, un humain dans la boucle est nécessaire. L’exécution d’un service web peut être considérée comme une suite d’appels de fonctions. En utilisant des langages qui fournissent des APIs déclaratives interprétable par machines, on peut exécuter ces appels de fonctions. Un agent logiciel doit être capable d’interpréter les balises du langage utilisé pour qu’il soit capable de comprendre les entrées nécessaires pour l’invocation du service, quelles sont les informations qui vont être retournées et comment le service peut être exécuté automatiquement. Ainsi le langage utilisé doit fournir des APIs déclaratives pour les services web qui vont être nécessaires pour l’automatisation d’exécution des services web.

- ***Une composition et une interopérabilité automatique des services web***

Cette tâche implique une sélection, une composition et une interopérabilité automatique des services web pour l’exécution de certaines tâches, étant donnée une description de haut niveau d’un objectif. Par exemple, un utilisateur peut vouloir avoir prendre tous les arrangements de voyages pour un voyage à une conférence. Actuellement, l’utilisateur doit sélectionner les services web, spécifier la composition manuellement et de faire en sorte que chaque logiciel requis pour l’interopération est créé. Donc il faut des langages qui vont coder les informations nécessaires pour la sélection et la composition des services web sur leurs sites. Un logiciel peut être développé pour manipuler ces représentations avec des spécifications des objectifs de la tâche. Le langage utilisé doit fournir des

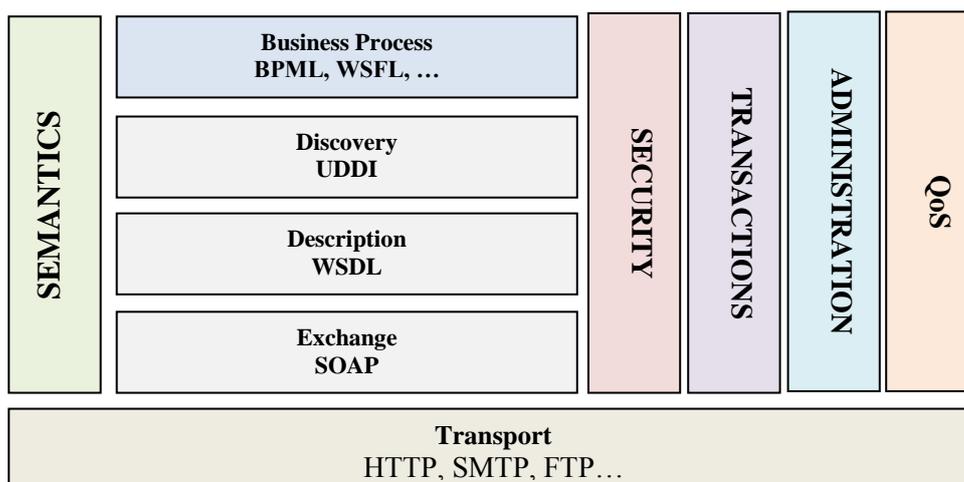
spécifications déclaratives des pré-requis et des conséquences de l'utilisation individuelle d'un service qui sont nécessaires pour l'automatisation de la composition des services et leurs interopérabilités.

- ***Une surveillance d'exécution automatique des services web***

Les services individuels et même la composition des services exigera souvent du temps pour s'exécuter complètement. Un utilisateur peu se demander pendant cette période sur l'état de sa requête, où les plans ont pu avoir changé, ce qui requière une altération dans les actions prises de l'agent logiciel. Par exemple, un utilisateur veut s'assurer qu'une réservation d'une chambre d'hôtel est déjà faite. Ça serait préférable d'avoir un moyen de découvrir l'état de la requête, et si des problèmes sont apparus dans le processus. Ainsi un langage choisi doit fournir des descriptions pour l'exécution des services. Ce point à été le sujet de plusieurs travaux de recherches, mais aucune approche n'a pu produire un langage qui fait une description satisfaisante de l'exécution des services web [4].

### 3.1. L'architecture des services web sémantiques

Le groupe architecture du W3C travail activement à l'élaboration d'une architecture étendue standard. Cette architecture est basée sur celle présenté dans la figure (Fig. 1.8) des services web, la pile est constituée de plusieurs couches, chaque couche s'appuyant sur un standard particulier. On retrouve, au-dessus de la couche de transport, les trois couches formant l'infrastructure de base décrite précédemment.



*Fig. 1.6 Pile des services web sémantiques*

Ces couches s'appuient sur les standards émergents SOAP, WSDL et UDDI. Une couche de « business process » est ajoutée, permettant de rendre les processus métiers accessibles à l'intérieur d'une entreprise et au-delà même de ces frontières. Les couches transversales : sécurité, transactions, administration et QoS, permettent de compléter la couche « *business process* ».

La couche « *SEMANTICS* » vient s'intégrer à chacune des quatre couches supérieures et permettant ainsi d'automatiser ces processus.

### 3.2. Descriptions sémantiques des services web

Actuellement plusieurs descriptions sémantique des services web existent, cependant cette section va donner une brève présentation des descriptions les plus importante dont on site : DAML-S, OWL-S et WSDL-S. Par contre, tout un chapitre est dédié à l'approche WSMO où elle sera détaillée.

#### 3.2.1. DAML-S

DAML-S est un langage de description des services basé sur XML utilisant le modèle des logiques de descriptions (et plus précisément DAML+OIL). Son intérêt est qu'il est un langage de haut niveau pour la description et l'invocation des services web dans lequel la sémantique est incluse (contrairement à UDDI).

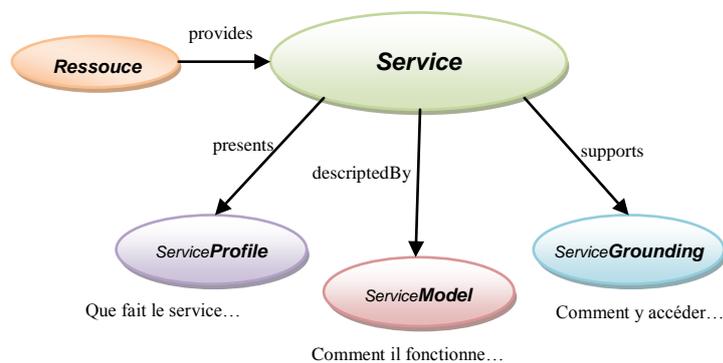
DAML-S est composé de trois parties principales : Le « *Service profile* » qui permet la description, la promotion et la découverte des services, en décrivant non seulement les services fournis, mais également des pré-conditions à la fourniture de ce service. Le « *Service model* » qui présente le fonctionnement du service en décrivant dans le détail et d'une manière relativement abstraite les opérations à effectuer pour y accéder. Certains éléments du *Service Model* peuvent être utilisés à la manière du *Service Profile* afin de fournir des informations supplémentaires à un utilisateur dans le cas où les opérations à effectuer seraient également un critère de choix. C'est le *Service Model* qui va permettre une composition des services si nécessaire. Il permet également d'effectuer un contrôle du déroulement du service. Le « *Service Grounding* » va présenter clairement et dans le détail la manière d'accéder à un service. C'est dans cette partie que les protocoles et les formats des messages échangés, sont spécifiés.

### 3.2.2. OWL-S

OWL-S est une ontologie OWL des services, pour décrire les différents aspects des services web. L'initiative d'OWL-S est issue de l'ontologie DAML pour les services (DAML-S) dont sa première version en mai 2001, c'était le premier effort progressif vers l'annotation sémantique des services web. OWL-S dans sa version 1.1 a été soumis au W3C en novembre 2004 par Nokia, l'Université de Maryland, le National Institute of Standards and Technology (NIST), Network Inference, SRI International, France Telecom, Stanford University, Toshiba, et l'Université de Southampton.

En passant de DAML+OIL à OWL, OWL-S a essayé d'adopter des recommandations existantes du web sémantique les couplant ainsi au monde des services web, en liant les descriptions d'OWL-S avec celles existantes de WSDL.

L'ontologie OWL-S comprend des concepts de haut niveau, elle définit le concept de haut niveau « Service » et trois sous ontologies OWL-S qui sont : « *Service Profile* », « *Service Model* » et « *Service Grounding* », qui sont illustrés dans la figure 1.11.



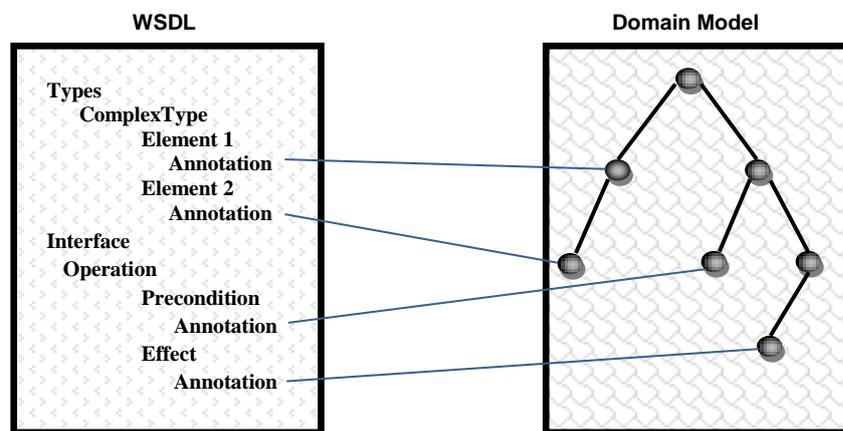
*Fig. 1.7 Le model conceptuel d'OWL-S*

### 3.2.3. WSDL-S

En utilisant XML-Schema et ayant le soutien des industriels, WSDL-S constitue une approche incrémentale pour l'ajout d'annotations sémantiques aux documents WSDL. Dans WSDL-S les utilisateurs peuvent ajouter des annotations sémantiques aux documents WSDL en utilisant l'extension définie dans la spécification WSDL. Les annotations

sémantiques peuvent être des références à des concepts définis dans des ontologies externes.

WSDL-S en tant que tel ne prescrit aucun langage particulier pour les ontologies et il n'est pas défini pour être lié aux langages des représentations sémantiques. Les utilisateurs peuvent utiliser OWL, WSMO (voir le deuxième chapitre) ou n'importe quel langage de modélisation. WSDL-S est issu du projet METEOR-S<sup>3</sup> de l'université de Georgia qui a été significativement modifié par IBM et l'équipe METEOR-S. la figure 1.12 illustre le principe de WSDL-S et met le point sur comment le model de domaine est maintenu externe au modèle des services web et comment les associations entre les concepts de WSDL et leurs annotations sémantiques correspondantes sont maintenues en utilisant des références (URI).



*Fig. 1.8 L'externalisation de la représentation et de l'association de la sémantique aux éléments de WSDL[4].*

## 4. Synthèse

Les services web deviennent des composants technologiques importants dans le domaine d'intégration d'applications. Aujourd'hui, les services web sémantiques constituent une voie nouvelle permettant de mieux exploiter les services en automatisant, autant que possible, les différentes tâches liées à un service.

Les services web sémantiques se retrouvent à l'intersection de deux domaines de recherche actifs qui sont le web sémantique et les services web. Le web sémantique ayant

<sup>3</sup> METEOR-S : <http://lsdis.cs.uga.edu/projects/meteor-s/>.

pour objectif d'enrichir le web en lui ajoutant une couche de métadonnées compréhensibles aux machines pour permettre aux programmes informatique de faire un traitement automatique de l'information. Et les services web qui sont des composants logiciels faiblement couplés fournis au dessus des technologies standardisées, offrant ainsi un vrai support pour l'intégration des applications et des services.

Entre dynamique et sémantique, les services web sémantiques est l'évidente solution qui promet l'automatisation des tâches de : description, découverte, sélection, composition et fourniture des services afin de créer ce qu'on appelle un web sémantique des services web.

---

# Chapitre II

## Chapitre II

---

## ***1.Introduction***

De par sa nature, l'architecture orientée service SOA (et son produit phare, les Web services (WS)) engendre certains problèmes qui restent encore non résolus. Comment trouver les services répondant aux requêtes de l'utilisateur ? Comment les faire connaître ? Quel est le degré de fiabilité du dit service ?...etc. Tous ces problèmes ont limité la propagation des WS. L'état actuel de la SOA a limité l'utilisation généralisée des services Web (ou des services conformes aux principes de la SOA en général) car plusieurs questions importantes demeurent sans réponse, y compris où publier des services pour une meilleure et rapide exposition, comment découvrir les services tout en respectant les besoins des utilisateurs, comment avoir confiance à des services quand ils sont trouvés, et comment remplacer facilement les services quand ils échouent. Ces problèmes ont fait que les services web ont échoué malgré leur potentiel :

- Ne savent que sur eux-mêmes, et non pas sur leurs utilisateurs ou pairs;
- limitent l'intervention des utilisateurs considérablement et fonctionnent comme des boîtes noires;
- ne considèrent que leurs propres détails fonctionnels et non fonctionnels internes en cours d'exécution et ignorent les détails extérieurs, tels que les interactions des utilisateurs passées;
- ne peuvent pas déléguer leurs invocations;
- ne coopèrent instantanément et volontairement pas les uns avec les autres ou avec eux-mêmes;

Les services Web sociaux (SWSc) permettent d'établir des réseaux de contacts, et de garder la trace des expériences et des partenariats passés. C'est l'union de deux branches importantes, à savoir l'informatique sociale et l'informatique orientée service.

L'informatique sociale permet la reproduction des relations et comportement humains. De même, c'est aussi le partage de contenu et la liberté de l'information. Quant à outils l'informatique orientée service consiste à construire des applications sur la base de l'offre et de la demande, avec comme philosophie la modularité et l'absence de hiérarchie cette fusion, donnant naissance aux SWSc et a introduit la notion de mémoire. Elle leur permet donc de

"savoir" avec qui ils ont interagi par le passé, et avec qui ils préféreraient travailler dans le futur [5].

## **2. Définition et fonctionnement global d'un réseau social de services WEB**

Il s'agit de graphes dont les nœuds sont les Web services et les arcs désignent les relations (interactions) entre ces services, les types d'interactions possibles sont les suivantes :

### **2.1. La collaboration :**

En se combinant, les SWSc peuvent travailler ensemble pour résoudre des requêtes complexes. De plus, un SWSc peut gérer son propre réseau de collaborateurs, de telle sorte à pouvoir décider s'il préfère travailler un collaborateur donné ou non, suivant leurs expériences précédentes.

### **2.2. La compétition :**

Lorsqu'ils offrent des fonctionnalités semblables, les SWSc peuvent entrer en compétition les uns avec les autres. Etant donné que seul l'un d'entre eux fera partie de la composition de SWSc qui traitera la requête client, le choix sera suivant le critère de qualité de service. Par conséquent, un SWSc sera conscient de son réseau de rivaux, et aura pour but d'améliorer sa qualité de service vis-à-vis de ses paires.

### **2.3. La substitution :**

Les services Web peuvent s'entraider lorsqu'une faille se produit, s'ils offrent les mêmes fonctionnalités. Par conséquent, un SWSc disposera de son réseau de remplaçants, de telle sorte à pouvoir choisir le meilleur substitut possible en cas de panne.

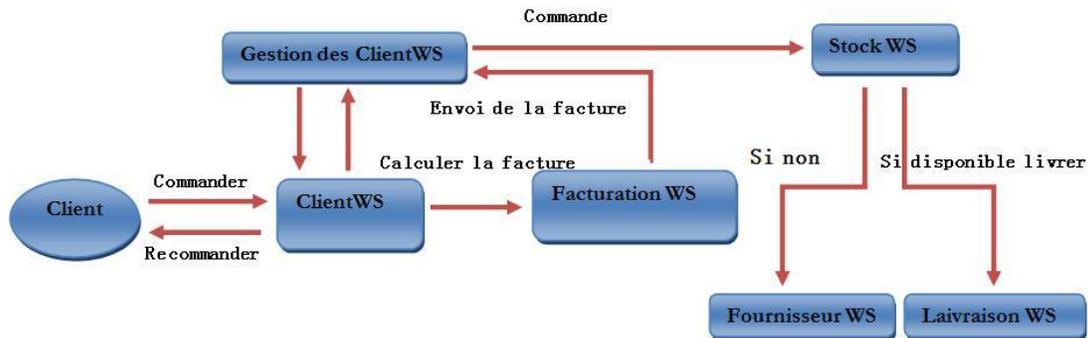
Ces trois (03) structures peuvent aussi être considérées comme des réseaux de comportement. De même, ils peuvent servir à la mise en place d'autres réseaux, lorsque de nouvelles interactions arrivent comme la délégation/supervision.

Deux services Web reliés par un arc de collaboration peuvent se recommander mutuellement à être inclus dans des compositions de Web services pour répondre à une requête donnée.

Les WS défaillants se feront remplacés par les WS auxquels ils sont reliés avec l'arc de substitution.

Et enfin, les services Web reliés par des arcs de compétition auront les mêmes propriétés et seuls l'un d'eux sera utilisé pour répondre à la requête d' l'utilisateur [6].

Pour illustrer le fonctionnement d'un réseau social de WS, considérer le scénario suivant pour l'achat en ligne.



*Figure 2.1 – Scénario de fonctionnement d'un réseau social de Web services*

Généralement, les services Web cités précédemment sont utilisés sans aucune considération pour les interactions qu'ils ont pu avoir avec d'autres WS dans le passé. Hors avec les SWSs, ces interactions peuvent se révéler précieuses.

Lors de la préparation d'une commande dans ce scénario donc, plusieurs relations peuvent se mettre en place.

- ✓ S'ils ont les mêmes fonctionnalités, les WS peuvent être en compétition les uns contre les autres. C'est ce qui a amené le choix de ClientWS plutôt que CommanderWS pour traiter les commandes des utilisateurs. Ou encore, des WSs peuvent en recommander d'autres pour prendre leur relève en cas de faille, comme LivraisonWS et TransportWS par exemple.
- ✓ Des WSs peuvent se recommander les uns les autres pour faire conjointement partie de compositions, s'ils sont complémentaires. Comme pour StockWS et LivraisonWS [9].

### **3. Approches existantes pour le développement de réseaux sociaux de Web services**

Même si le concept est relativement récent, il existe déjà plusieurs approches pour les réseaux sociaux de Web services :

- Réseau social intégrant des composantes du Web 2.0, des Web services enrichis sémantiquement. Les interactions sont les différentes étiquettes que les utilisateurs assignent aux WS.
- Une framework pour la composition sémantique de Web services dans les réseaux sociaux. L'élément social est représenté par la confiance entre fournisseurs, clients et les services eux-mêmes. Framework composée de plusieurs modules : extraction et construction sémantique de RS, stockage de RS...etc.
- *SoCo* : Social Composer. Recommander la meilleure décision à prendre en ce qui concerne la sélection de WS. Prise en compte des interactions entre utilisateurs et WS. Mémoire : se souvenir des compositions passées. SoCo est composé de différents modules : extraction et modélisation d'infos sociales, gestionnaire de recommandations, de connections...etc.
- *LinkedWS* comme modèle de réseaux sociaux de WS. Ne pas traiter les WS comme des éléments à part, ils rivalisent entre eux, ils collaborent et proposent des substitutions[6].

Parmi ces approches nous allons nous intéresser à une approche dite *Approche méthodique*.

### **4. Développement des services du web social**

Le développement de services Web sociaux nécessite six étapes. L'idée est d'identifier les paires d'un service de contact possible[7]

#### **Étape 1: Identification des composants du réseau social**

Les composants d'un réseau social sont les nœuds et les arcs. Dans notre travail, ils correspondent, respectivement, à des services Web et les interactions entre eux. Ces interactions sont la concurrence, la collaboration et la substitution.

Avec une arc de collaboration, un service Web engagé dans une composition en court recommande qu'un concepteur de service inclue autres pairs. Le fournisseur peut accepter ou rejeter la recommandation

Avec une arc de substitution, un service Web accepte que les pairs similaires vont le remplacer quand il échoue .Ce type de substitution est possible en raison de la similitude de leurs fonctionnalités

Pour parvenir à une substitution parfaite des services Web, leurs propriétés non fonctionnelles devraient également être similaires, bien que cela ne soit pas nécessaire Enfin, avec une arête de concurrence, un service Web a les mêmes propriétés fonctionnelles que d'autres pairs, et un utilisateur ne doit invoquer qu'un, en raison de sa capacité à satisfaire ses exigences

## **Etape 2 : correspondances des analyses des services web**

Pour établir la nature des interactions sociales entre les services Web, nous faisons correspondre leurs propriétés fonctionnelles et non fonctionnelles respectives, formant le profil d'un service Web. La correspondance démontre si deux services Web  $s_i$  et  $s_j$  sont similaires ou complémentaires, Nous décomposons le profil en cinq catégories: conditions préalables : (P), les entrées (I), les sorties (O), les effets (E), et qualité de service(QoS) Les approches traditionnelles de correspondance mesurent les relations de subsomption entre les concepts de ces catégories. Nous adoptons une approche qui établit le degré de similitude(DS) entre  $s_i$  et  $s_j$ . pour chaque paire de concepts  $c_{s_i}$  et  $c_{s_j}$  dans la même catégorie de profil, le résultat de correspondance (MS) entre cette paire est une fonction de :

- $l$ , le nombre d'arêtes de  $c_{s_i}$  à  $c_{s_j}$  dans l'ontologie;
- $h$ , la profondeur de  $c_{s_i}$  dans l'ontologie; et
- $d$ , la densité sémantique locale de  $c_{s_i}$  et  $c_{s_j}$ .

Donc, nous calculons le résultat de correspondance comme suit:

$$MS(c_{s_i}, c_{s_j}) = f_1 * f_2 * f_3$$

$$\text{avec } f_1 = e^{\alpha l} \text{ avec } \alpha \text{ un constant et } f_2 = \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}} \text{ avec } \beta \text{ un facteur et } f_3 = \frac{e^{\gamma l} - e^{-\gamma l}}{e^{\gamma l} + e^{-\gamma l}}$$

avec  $\gamma$  un facteur Nous calculons le degré de similitude ([0, 1]) entre  $s_i$  et  $s_j$  en utilisant la moyenne pondérée des scores correspondant entre chaque paire de concept dans le profil des catégories :

$DS(s_i, s_j) = \frac{\sum_k w_k * Ms(csk, csk)}{\sum_k w_k}$  Où k est le nombre total de concepts similaires Et  $W_k$  est le

poids de l'appariement de score entre une paire de concepts

Nous adoptons une approche similaire pour calculer le degré de complémentarité (DC) entre  $s_i$  et  $s_j$ , que nous noterons DC ( $s_i, s_j$ ). Nous disons  $s_i$  et  $s_j$  sont complémentaires lorsque les effets de conditions préalables correspondent aux effets de toutes les  $s_j$ . Dans ce qui suit, nous nous concentrons seulement sur les services Web similaires.

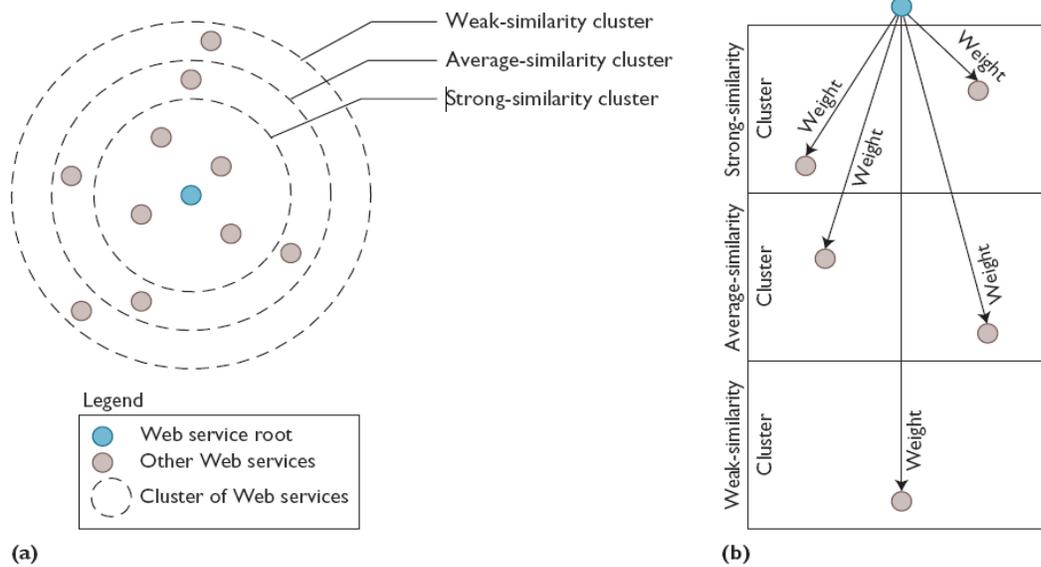
Avant de passer à l'étape suivante, nous allons voir comment les réseaux sociaux sont structurés.

Deux options existent. Dans la première, on développe trois réseaux sociaux distincts, un par type d'interaction sociale. Si un service Web n'est pas et ne sera pas engagée dans une certaine interaction, alors il n'y aura pas d'interaction dans le réseau social. Dans la deuxième option, nous développons un réseau social pour les trois interactions. Nous structurons alors le poids d'une arête comme un triplé de valeurs qui correspondent chacun à une interaction sociale spécifique.

### **Etape 3 : gestion du réseau social**

Cette étape nécessite un type particulier de nœud, que nous appelons le service racine. Nous le définissons par rapport aux trois états qui définissent un cycle de vie du réseau social:

- l'état de construction. Tout service Web qui fera partie du futur réseau social est le service racine. Ainsi, la sélection de la racine est aléatoire. Les autres services Web seront connectés à cette racine.



**Figure 2.2** gestion du réseau social.

- L'état de l'exploitation. Lorsque nous décidons d'utiliser un réseau social pour une composition en cours, deux cas se présentent. Le premier est la collaboration / compétition - qui est, tout composant service Web dans cette composition peut être une racine.

L'objectif est de chercher les collaborateurs / concurrents potentiels. Le deuxième cas est la substitution, dans laquelle les composant service Web qui ont échoué forment la racine.

- en cas d'expansion, tous les services qui vont intégrer le réseau sont les racines

Après avoir identifié les types d'interactions nécessaires entre les services Web, nous construisons soit de nouveaux réseaux sociaux à partir de zéro ou nous étendons ceux qui existent déjà (l'ajout de nouveaux nœuds et arc). Quand nous construisons un réseau social pour la première fois, nous regroupons les services Web en trois classes en fonction de leurs degrés de similitude avec le service Web racine : faible (par exemple  $0 < DS \leq 0,33$ ), moyen ( $0,33 < DS \leq 0,66$ ) et fort ( $0,66 < DS \leq 1$ ). Nous utilisons le degré de similarité plus tard pour affecter des valeurs initiales de poids d'arc (Notez que 0,33 et 0,66 ne sont que des exemples de valeurs).

Si un pair a un degré de similitude fort avec la racine, nous le plaçons dans la classe de forte similitude de la racine.

Le même processus de placement s'applique aux degrés moyens et faible-similarité, et continue tant que les services demeurent disponibles et accepte de faire partie du réseau social. Tant que la segmentation des nœuds est en cours, les services Web se connectent aussi bien au sein des réseaux sociaux, qui va finalement étendre ces réseaux .

#### Étape 4: Évaluation du poids d'arête initial

Le poids initial d'un arc noté WE entre  $s_i$  et  $s_j$ , où  $s_i$  est le service Web racine, correspond au degré de similitude entre eux selon l'équation 3, qui calcule WE au temps ( $t_0$ ):

$$WE_{t_0}(s_i, s_j) = DS(s_i, s_j). \quad (3)$$

Cette valeur initiale changera on fur et à mesure que l'utilisation du réseau social devient effective. Chaque fois que nous découvrons un service Web en utilisant le réseau social d'un pair, nous mettons à jour automatiquement les poids des arcs.

#### Etape 5 : naviguer dans les réseaux sociaux

Des outils appropriés sont nécessaires pour aider les concepteurs des services pour naviguer dans les réseaux sociaux des services Web. Chaque service Web racine est un point d'entrée. Tout dépend si on cherche des collaborateurs, des substituteurs ou des compétiteurs, nous aurons besoin des facteurs qui guident l'identification des pairs. Ici, nous proposons certains facteurs dans le cas de substitution.

$P_c$  est la priorité d'inspecter une classe spécifique de service web. Les classes forts ont une haute priorité. La priorité d'une classe est fixé.

$O_{s_j}$  est le coût qu'un service web  $s_j$  nécessite quand il est sélectionné par le biais d'un réseau social. Ce coût est proportionnel à la priorité d'une classe et inversement proportionnel au poids de l'arc qui relie le service Web à sa racine  $s_i$ , et qui a besoin d'un substituteur:

$$O_{s_j} = \frac{P_c}{1 + (P_c \times WE_{t_0}(s_i, s_j))} \quad (04)$$

Puisque le poids d'arcs changent au fil du temps, le coût d'utilisation d'un service Web est dynamique. Par conséquent, nous faisons soit la promotion d'un service Web, ce qui lui fait passer à un meilleur classe (par exemple, à partir d'un classe faible à un classe moyen), ou dégrader (déplacer d'une classe forte à une classe moyenne par exemple).

$E_{sj}$  représente le niveau de satisfaction des expériences précédentes quand un service Web  $s_j$  a été sélectionné à travers d'un réseau social et évidemment déployé avec succès (sans échec).

$L_{sj}$  représente la charge actuelle de  $s_j$  par rapport à ses propriétés non fonctionnelles. Plus la charge est importante, moins le service Web est attractif.

Nous définissons une fonction de sélection appelée sélections  $j$  comme une agrégation de ces facteurs:

$$Selection_{sj} = \alpha_1 * O_{sj} + \alpha_2 * E_{sj} + \alpha_3 * (1 - L_{sj}), \quad (5)$$

où  $\alpha_i$  est compris entre 0 (exclu) et 1. Après l'application de l'équation 5 à tous les services Web dans un réseau social, le pair avec la valeur la plus élevée de *Sélections j* est considéré comme le substitut le plus approprié.

#### **Etape 6 : l'évaluation continue des poids des arcs :**

L'évaluation continue reflète le rôle de réseaux sociaux dans la découverte des services Web. Elle se produit lorsque nous mettons à jour les poids des arcs à chaque fois qu'un substituteur, de collaborateur, ou de compétiteur est découvert en utilisant ces réseaux. On obtient des valeurs initiales des poids d'arcs à l'aide d'équations (03). Pour mettre à jour ces poids, nous proposons des fonctions appropriées en fonction du type d'interaction entre les services Web. L'équation 6 évalue les poids dans un réseau social de substitution en utilisant les prix basés sur les récompenses: (6)

$$WE_{t+\delta t}(s_i, s_j) = WE_t(s_i, s_j) + \alpha \times \left( \frac{|s_j selection_{t+\delta t}|}{|s_i failure_{t+\delta t}|} - WE_t(s_i, s_j) \right), \quad (6)$$

où  $\alpha$  est une constante entre 0 et 1,  $A_t$  est la période de mise à jour,  $|sélection s_j|$  est le nombre de fois que  $s_j$  est substitué pour  $s_i$  selon l'utilisation du réseau social de

substitution de  $s_i$ , et  $| \text{failure } s_i |$  est le nombre de fois que le service  $s_i$  a échoué. Equation 6 suppose que  $s_j$  n'a pas échoué, alors la mise à jour des poids est finalisée. Nous exécutons la navigation à travers l'évaluation du réseau social et l'évaluation du poids des arcs d'une manière itérative à chaque période de temps  $\hat{t}$ .

#### ***4. Du Data Mining sur les réseaux sociaux***

Les réseaux sociaux offrent un énorme tas de données idéal pour le *Data Mining*. Les relations entre individus et corrélations ainsi que les groupements d'individus ne sont pas distinguables. Le *Data Mining* palie à ce problème grâce aux divers outils de classification et d'extraction de règles de corrélation qu'il offre. Le challenge est que les données sont énormes à l'image de

*Facebook* par exemple avec plus de 3 milliard qui diffusent et commentent. Des données issues de réactions aux événements et changeantes dans le temps donc dynamiques, des données bruitées. Les ingénieurs de *Facebook* ont réalisé un système de suggestion de groupes à l'aide d'arbres de décisions et de méthodes de clustering .

Les variantes du datamining exercées sur les média sociaux sont les suivantes :

##### **4.1 Data Mining dans l'analyse des sentiments et de l'opinion publique**

Des chercheurs de l'université du Maryland ont développé un outil baptisé *BlogVox* qui permet à l'aide de SVM de déterminer si un blog exprime une opinion ou non après prétraitement des données comme l'indique le schéma suivant :

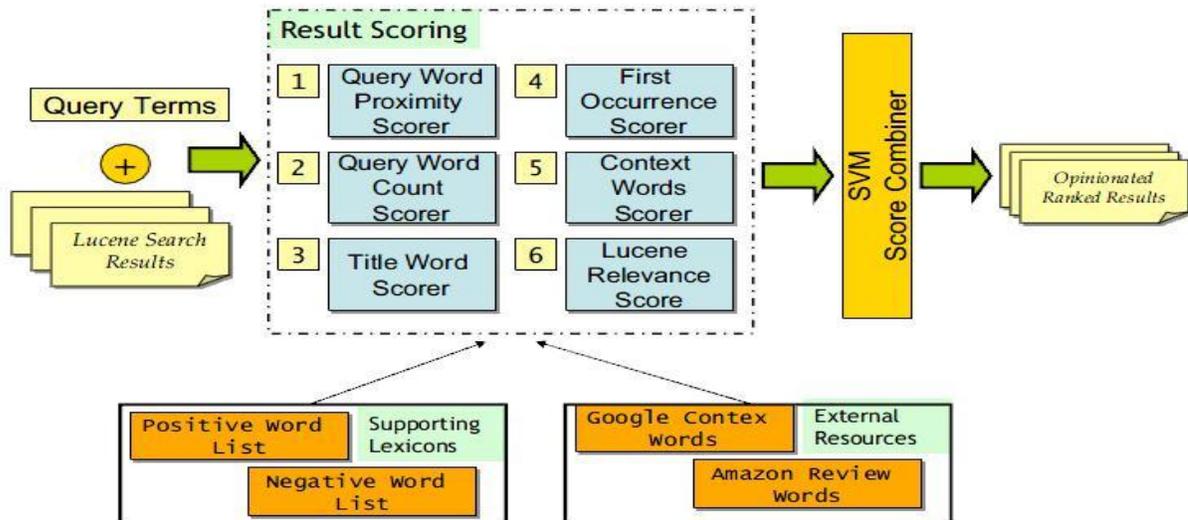


Figure 2.3 - Fonctionnement de BlogVox

## 5.2. Data Mining dans l'identification de nœuds à forte influence

Un individu est dit influent s'il affecte le comportement social d'autres individus, l'intérêt d'identifier ces individus influents est commerciale, extraire les paramètres d'influence et les utiliser pour ajuster les publicités et augmenter les chances de vendre des services et produits.

Les paramètres d'influences à exploiter sont résumés dans l'approche *iFinder* qui définit l'influence en quatre (04) paramètres :

- **La reconnaissance** : qui compte le nombre d'arcs entrants à la publication, c.à.d. les individus qui prennent part à une nouvelle diffusion.
- **La génération de l'activité** : qui compte le nombre de commentaires reçus pour la diffusion.
- **La nouveauté** : qui mesure l'impact de la nouvelle information dans le réseau, c.à.d. le taux et la vitesse de partage.
- **L'éloquence** : qui est une approximation de la taille du post [10].

## 6. Le tagging dans les réseaux sociaux

Le tag dans les réseaux sociaux consiste à annoter les ressources telles que les photos, de noms et de lieu. Ceci a pour effet de rendre plus compréhensible la ressource ce qui va

générer une folksonomie qui est en fait une classification de la ressource issue d'utilisateurs du réseau et non de professionnels.

Le tag peut être motivé par l'envie d'attirer l'attention sur une certaine donnée, pour retrouver par la suite cette information, pour décrire la ressource et le contexte aux moyens de tags ou exprimer son avis vis-à-vis l'information exprimée dans la ressource. Ceci va prendre une allure déterminante lorsqu'il s'agira de classifier les ressources du réseau ou d'extraction de connaissance. L'autre faire valoir des tags est purement publicitaire, on le retrouve avec Amazon qui paye des utilisateurs qui créent des tags, cette technologie a été facilitée avec l'avènement du *drag & drop* et les interfaces de plus en plus ergonomes et dynamiques en vue de *smartphone* et de technologie miniature.

- **Variantes de tags**

Plusieurs types de tags traînent sur les media sociaux, on y retrouve des tags basés sur le contenu qui décrivent le contenu de la ressource, des tags basés sur le contexte de la ressource qui décrivent le contexte de la ressource comme par exemple le contexte dans lequel a été prise une photo (un anniversaire, un mariage...etc), des tags de propriété qui indiquent le propriétaire de la ressource, des tags qui dénotent les éléments de la ressource comme par exemple le cas d'une publication où on cite le titre, les personnes, les mots-clés ...etc. Il y a aussi des tags d'organisations et de groupements où l'on présente les acteurs d'un groupement particulier, des tags personnels où l'on se présente soit même ou que l'on cite ses propres objets[8].

---

Chapitre III

**Chapitre III**

---

## 1. Introduction

Mammar ZAKARIA a proposé dans [5] une nouvelle approche pour la création de réseaux sociaux de services Web, la méthode proposée comporte six étapes essentielles à la création d'un réseau social de service Web. Nous avons suivi les étapes décrites et nous les avons concrétisées à travers un réel réseau de services.

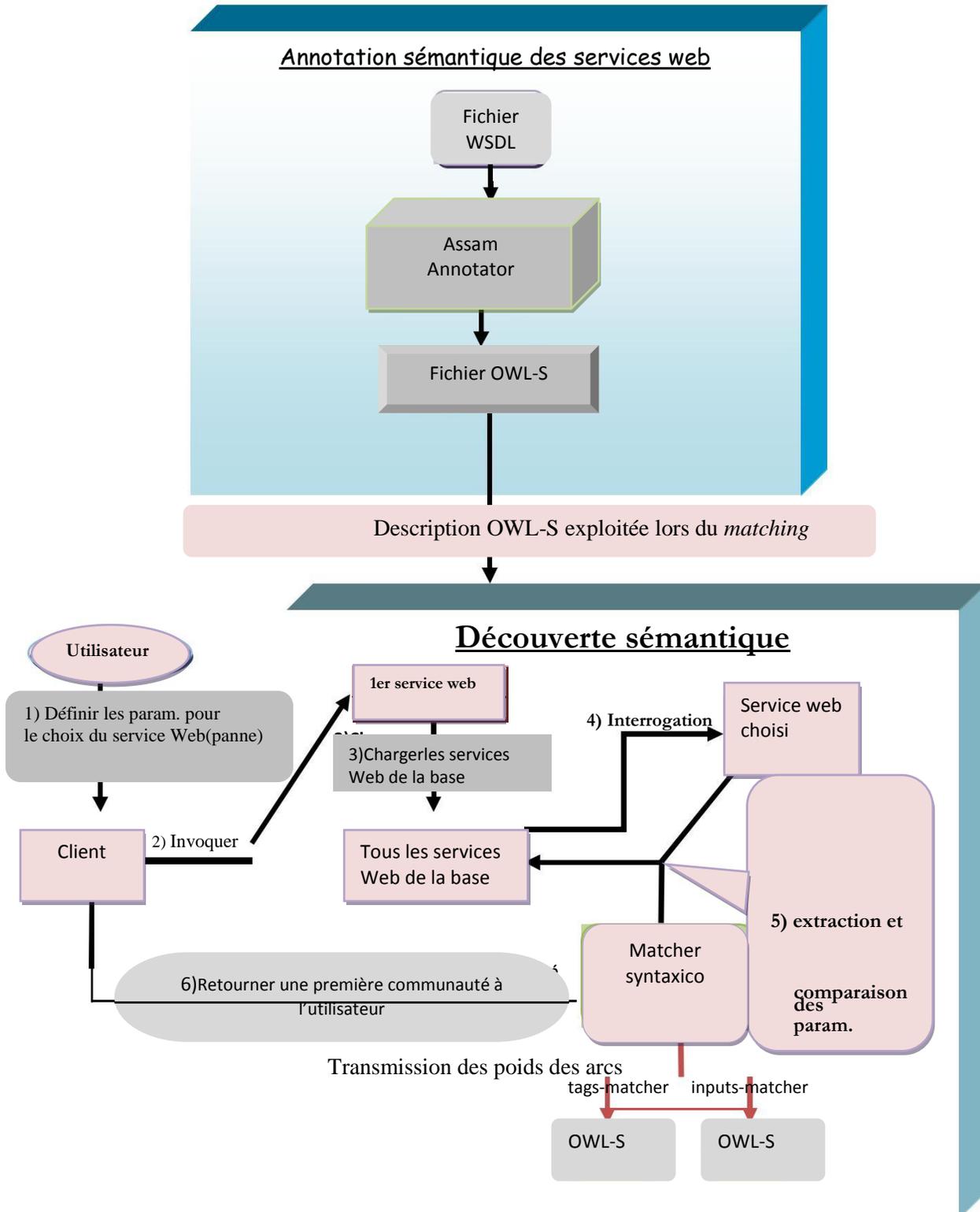
Le réseau social que nous avons construit se compose de deux couches essentielles à savoir :

- **la couche 1** : Il s'agit de la couche qui gère l'annotation sémantique des descripteurs des services Web de la collection ou l'annotateur prend comme argument un descripteur de service WSDL et retourne une description OWL-S ceci prend son intérêt lors de la découverte syntaxico-sémantique ou un utilisateur invoque un service Web Loader des services de la collection puis il simule la panne de l'un des services en l'invoquant, l'invocation du service en question appelle une fonction de matching sémantique entre les différentes descriptions des services de la collection et le service en panne se basant sur leurs paramètres décrit en OWL-S , en retour on obtient une première communauté de remplaçants potentiels .
- **la couche sociale** : Il s'agit de la plus importante couche lors de l'étape de découverte des services Web, cette couche trace au fil du temps l'interaction sociale entre les services et invoque des fonctions de mise à jour des différents paramètres sociaux comme le poids des arcs entre les services Web , et raffine les premières communautés de remplaçants retournés par la découverte précédente , ceci est fait en ajoutant des arcs , supprimant d'autres , en mettant en avant d'autres .

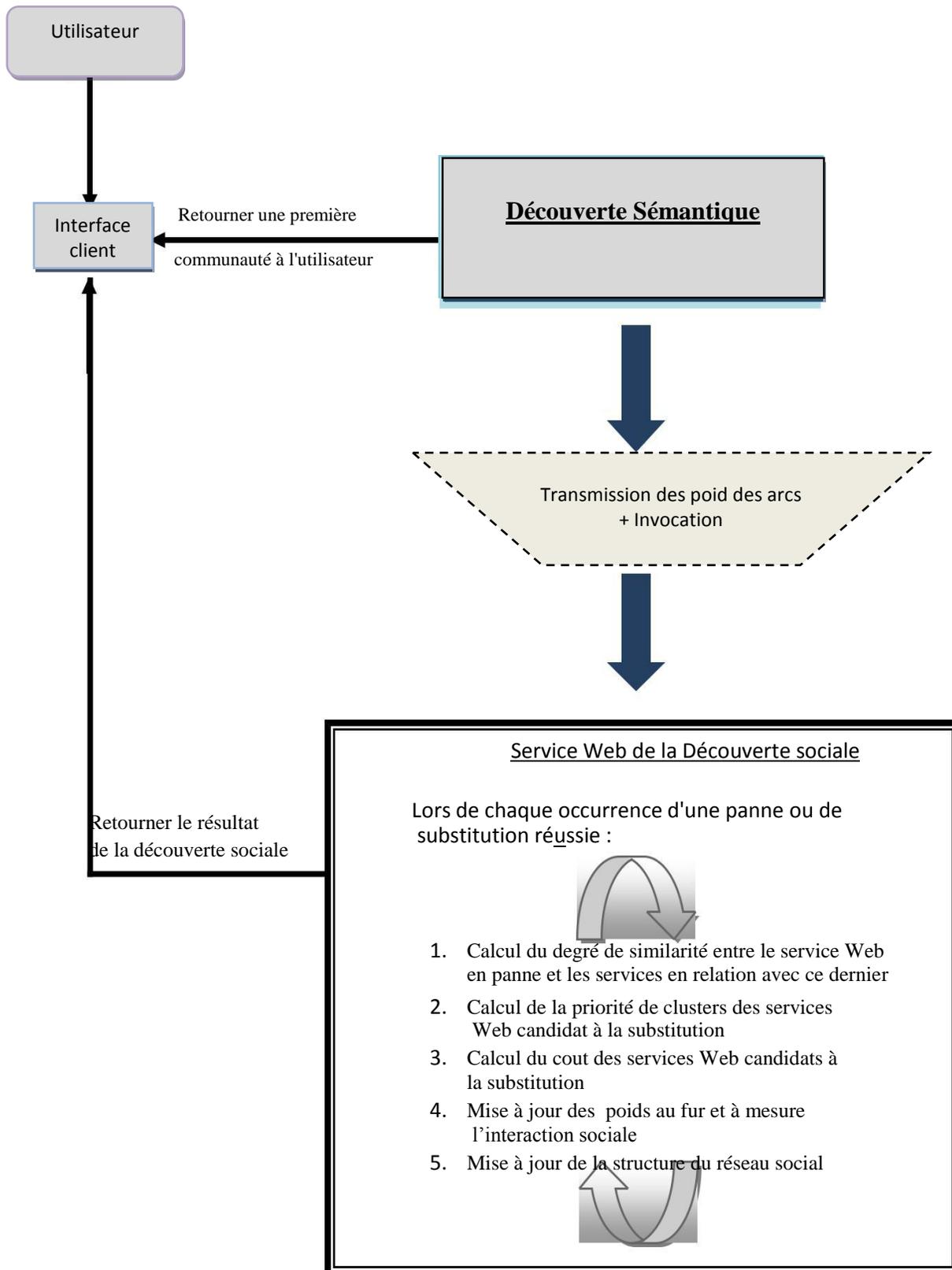
L'architecture suivante dénote les différentes étapes de création d'un réseau social de services Web.

## 2. Architecture du réseau social de Web services

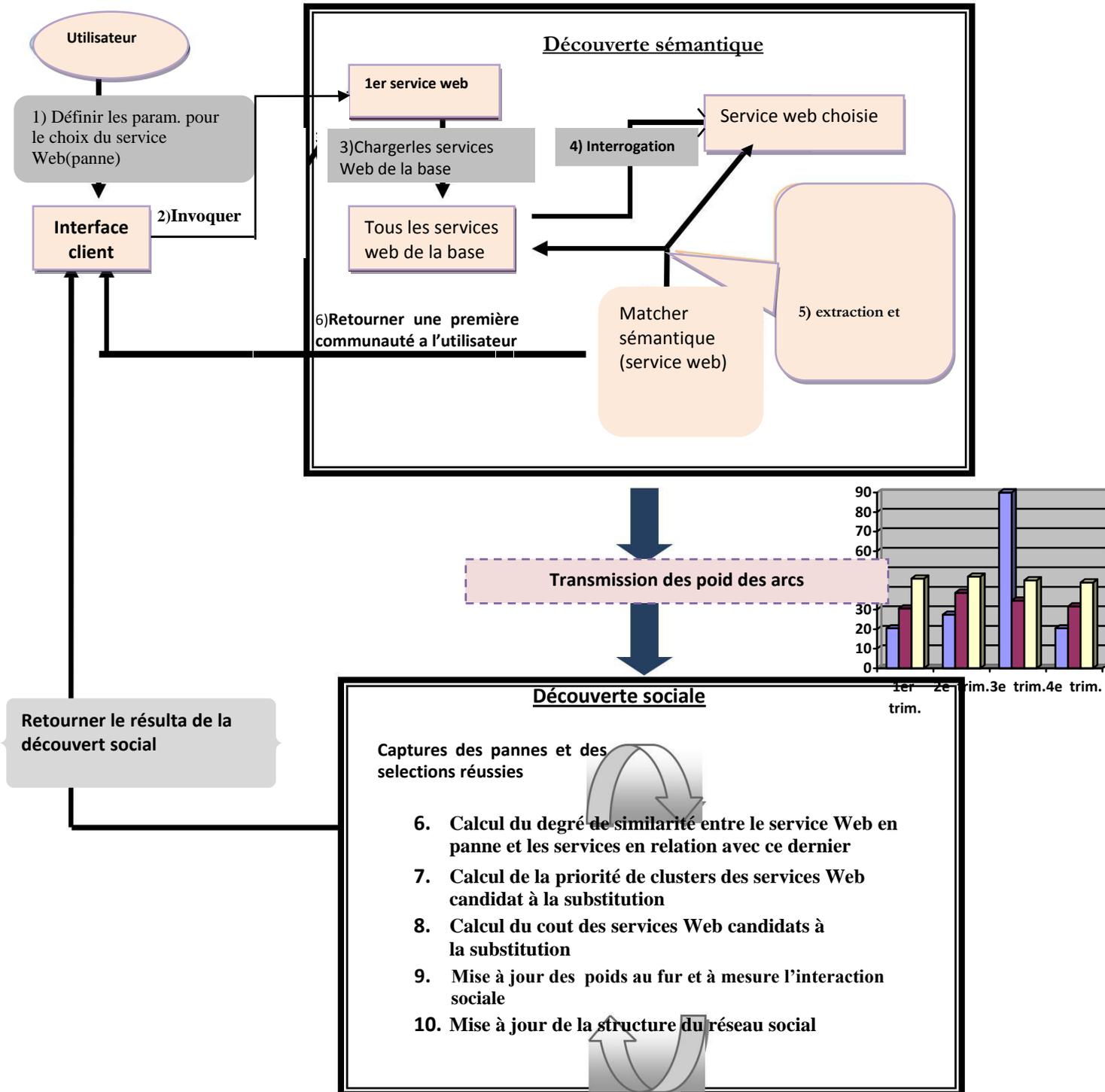
### 2.1. Couche 1 : Annotation et recherche sémantique



## 2.2. Couche 2 : Découverte sociale



### 3. Architecture globale



## 4. Implémentation et mise en œuvre

### 4.1. Résumé

Dans ce chapitre nous montrons l'avantage d'une découverte sociale par rapport à la découverte sémantique des services Web et de dégager des communautés de service Web proches de fonctions grâce aux données issues de la découverte sociale.

Pour ce faire nous avons construit un véritable réseau social de services Web selon l'approche proposée par Mammar Zakaria dans [8], le réseau social permet de capter le comportement social des services Web du réseau, ces données sociales permettront à un service WEB en cas de panne de proposer une communauté de services Web remplaçants pertinents et ainsi de s'auto guérir !

A noter que les types d'interactions possibles au sein d'un réseau social de service WEB sont la composition entre services, la compétition et la substitution, nous avons focalisé le travail sur la substitution entre services WEB.

### 4.2. Fonctionnement globale

Nous avons construit un réseau social de services Web ou nous avons focalisé l'interaction entre les services du réseau sur la relation de substitution. Ainsi, un service Web se verra capable de découvrir dans une première étape sémantiquement une communauté de services remplaçants en cas de panne. Par la suite les poids des relations de cette découverte seront sujets à des modifications issues de l'interaction sociale, un ensemble de paramètres sociales seront calculés à l'aide de formules visant à capter la similarité entre services au fur et à mesure leurs interactions dans le réseau le tout fait l'objet de la découverte sociale.

#### 4.2.1. Découverte syntaxico-sémantique

La découverte syntaxico-sémantique a été rendue possible grâce au fait que nous avons annoté chaque service Web d'une description sémantique *OWL-S*, cette description contient les informations suivantes :

- La description textuelle des fonctions du service ainsi que des paramètres d'entrée sortie des fonctions des services Web dans le fichier *profil*
- Des détails sur les points d'accès du service dans le fichier *grounding*

Pour implémenter cette découverte nous avons exploité le fichier *profil.owl* à lui seul . La description textuelle du service étant présente dans *profil.owl* et les input et output toujours présents dans le *profil.owl*.

Pour le cas des inputs, nous avons élaboré un service Web qui s'occupe d'extraire les paramètres d'entrées de chaque fonction puis de les relier au numéro de service comme l'explique l'algorithme suivant :

**Definput\_output\_matcher (path,ikey0,ikey1) :#ikey0 et ikey1 sont les id de Ws**

1. Pour tous les fichiers *Profile.owl* , extraire les inputs et les mettre dans une liste

```
listInput = re.findall(r'"+"<profile:hasInputrdf:resource="\&the_concepts;#"+"+"+"_[0-9]*\>"', result)
```

2. Ajouter au dictionnaire des contenus, la liste des inputs avec le nom du fichier correspondant

```
dictContent [fileId] = listInput
```

3. Pour chaque 2 fichiers descripteurs (liaison) calculer le nombre de paramètre d'entrées communs enregistrées dans le dictionnaire des contenus
4. Mettre à jour la table des poids dans la table *semweights*
5. Retourner le poids de la relation entre les 2 Web services passés en paramètres

La figure suivante illustre un exemple des paramètres d'entrées sorties au sein de la description OWL-S

```
<profile:hasInput rdf:resource="&the_concepts;#username_38055"/>
<profile:hasInput rdf:resource="&the_concepts;#password_38056"/>
<profile:hasInput rdf:resource="&the_concepts;#sessionId_38057"/>
<profile:hasOutput rdf:resource="&the_concepts;#SummaryReport_38059"/>
<profile:hasInput rdf:resource="&the_concepts;#username_38062"/>
<profile:hasInput rdf:resource="&the_concepts;#password_38063"/>
<profile:hasInput rdf:resource="&the_concepts;#callId_38064"/>
<profile:hasOutput rdf:resource="&the_concepts;#CallDetails_38066"/>
```

**Figure 3.1 – Inputs Outputs dans le fichier profile OWL-S**

Pour le cas de la description textuelle nous avons élaboré un service qui s'occupe d'extraire le texte descripteur présent entre les balises `<profile:textDescription></profile:textDescription>`, puis de le nettoyer des mots vides et communs, en résultait un ensemble de termes pertinents décrivant chaque service. Ceci étant propagé sur l'ensemble des services de la collection, il ne restait plus qu'à comparer ces tags descripteurs tel l'indique l'algorithme suivant.

**Deftext\_matcher (path,ikey0,ikey1) :#ikey0 et ikey1 sont les id de Ws**

1. Pour tous les fichiers Profile.owl, extraire les inputs et les mettre dans une liste

```
listDescr =re.findall(r""+"<profile:textDescription>"+".+"+"</profile:textDescription>", result)
```

2. Nettoyer les descriptions textuelles des mots vides et communs ce qui donne lieu à des tags descripteurs
3. Ajouter au dictionnaire des descriptions, la liste des termes descripteurs avec le nom du fichier correspondant
4. Pour chaque 2 fichiers descripteurs (liaison), calculer le strength
5. Mettre à jour la table *networkstate*
6. Retourner le poids de la relation entre les 2 Web services passés en paramètres ikey0 et ikey1

La figure suivante illustre un exemple d'une description textuelle au sein de la description *OWL-S*

```
<profile:textDescription>
Provides an interface for retrieving detailed event logs for your organizations messaging history
</profile:textDescription>
```

**Figure 3.2 – Description textuelle dans le fichier profile *OWL-S***

**La découverte syntaxico-sémantique** se situe à l'intersection des deux services comparateurs cité ci-dessus, cette découverte retourne pour deux Web services un poids. Dès cet instant, les relations sociales venaient d'être pondérées et on pouvait dès lors supprimer certains arcs dont le poids était en dessous d'un certain seuil et classer les relations sociales par rapport au poids.

Cette première découverte donne lieu à une première communauté de substituteurs de services, les meilleurs substituteurs seront naturellement ceux dont la relation avec le service en panne a le plus grand poids, mais cette découverte n'était en fait que le paramètre d'entrée de la découverte sociale qui allait pallier au problème de la découverte sémantique cités dans le chapitre Web sémantique. En effet ces poids demeuraient rigides par rapport au temps et aux interactions que subissait le réseau, les actions temps réel que subissait le réseau n'étaient point captées, ce qui devait naturellement fausser la pertinence des résultats retournés.

Ces actions sont à titre d'exemple :

*Est-ce que vraiment le service x est disponible à l'invocation ?*

*Est-il fiable ? A-t-il déjà réussi de réelles substitutions ? Combien me coute-t-il de l'invoquer ?*

La découverte sociale tente de répondre à ces questions et les poids sont mis à jour automatiquement avec le temps.

Pour capter l'interaction sociale nous avons créé un ensemble de classes dont les fonctions servent au calcul des paramètres sociaux décrites dans le papier [7], ces fonctions seront détaillée selon de ce chapitre.

### 4.2.2 Découverte sociale

Le réseau social étant un graphe, nous l'avons modélisé grâce à une matrice dont la structure est la suivante :

( *id\_sws1* , *id\_sws2* , *l* , *h* , *d* , *MatchingScore* , *DSim* , *fails\_s1* , *select\_s2* , *weight* ) Où :

*id\_sws1*: Le numéro du service i.

*id\_sws2*: Le numéro du service j.

*l*, Nombre d'arcs à partir du service i vers le service j dans la même ontologie.

*h*, La profondeur du concept i dans l'ontologie

*d*, La densité sémantique locale entre le concept i et j

*MatchingScore* : Calculé grâce à l'équation (1) du papier

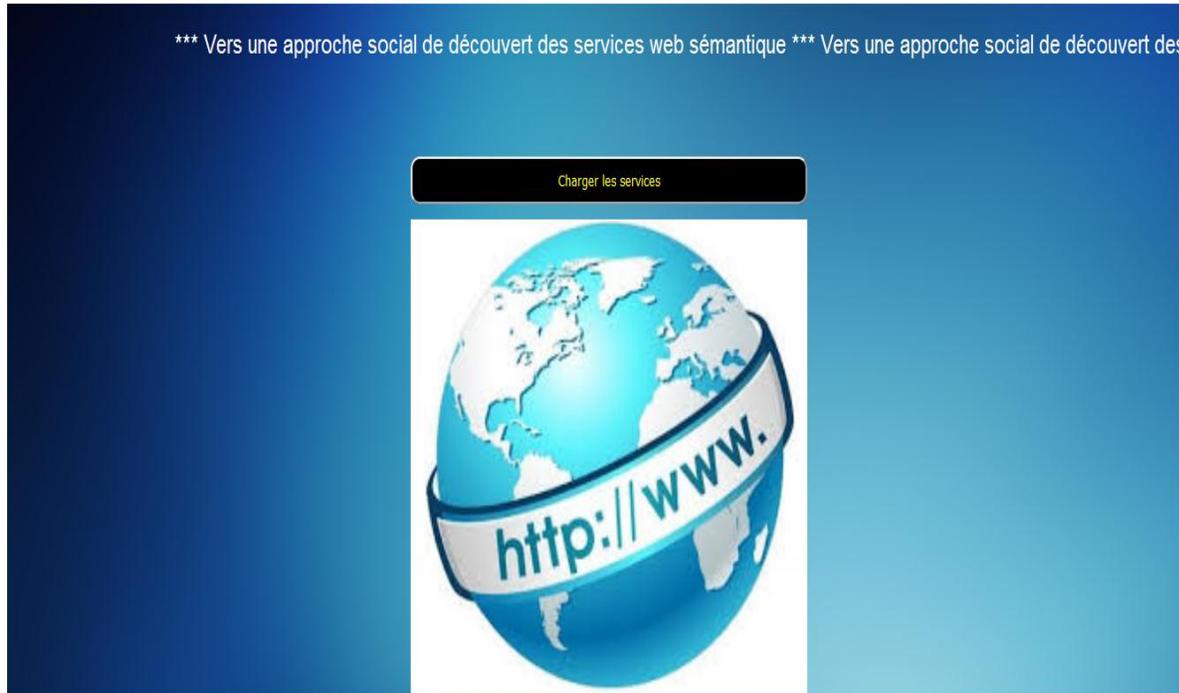
*fails\_s1* : Le nombre de pannes du service sws1 a l'instant courant

*select\_s2* : le nombre de sélections du service sws2 pour remplacer sws1 a l'instant courant.

*weight* : le poids de la relation entre deux services Web, il dépend des paramètres cités précédemment, il est mis à jour régulièrement

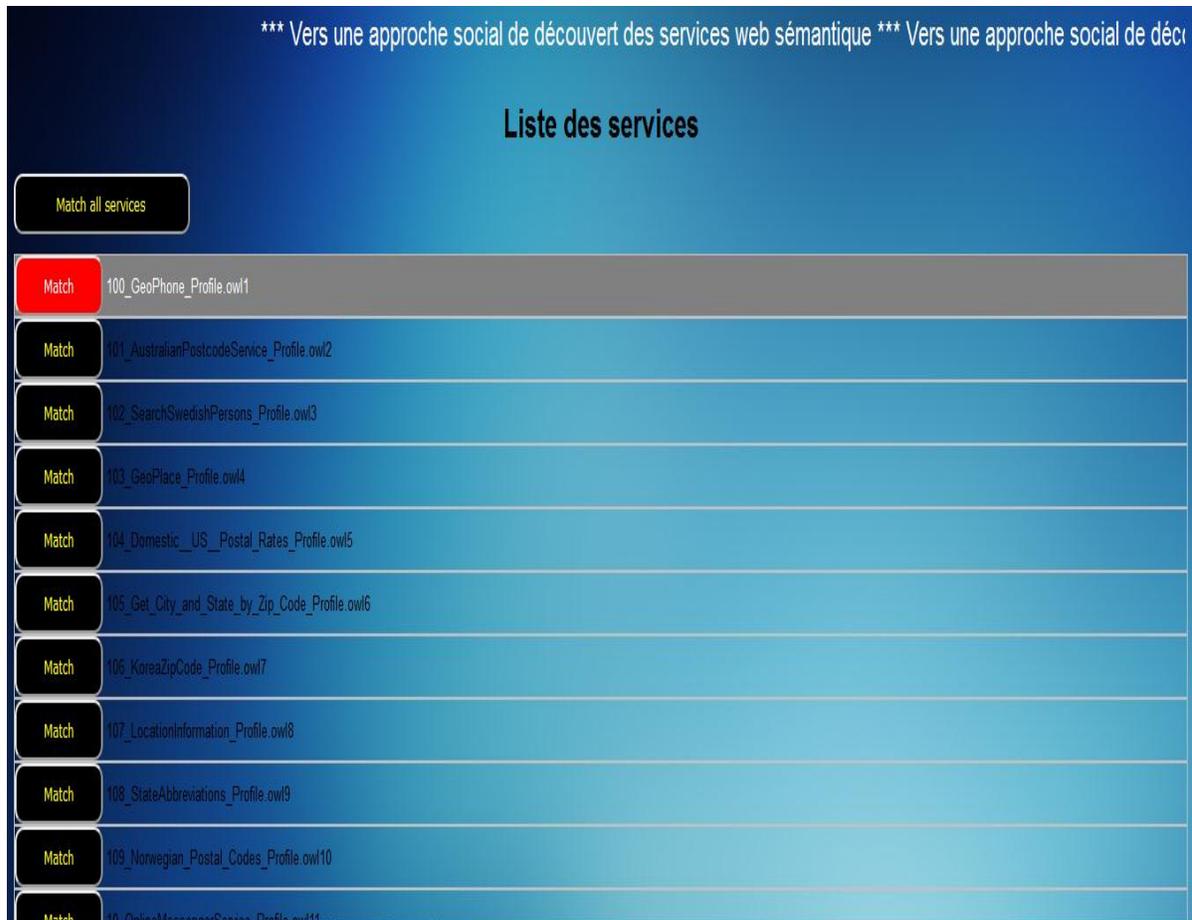
### 4.3. Scénario de fonctionnement du réseau social

Pour donner de la valeur à la découverte sociale, nous partons d'une collection de réels fichiers WSDL que nous avons annoté manuellement à l'aide d'Assam Annotator (présente sur <http://andreas-hess.info/projects/annotator/index>), donc à chaque descripteur WSDL est assigné une description sémantique *OWL-S 1.0*.



**Figure 3.3 – Interface principale**

le clic sur le bouton Charger Web Service va charger tous les services Web de la collection. Une fois ces résultats retournés dans l'interface, l'utilisateur se retrouve en face d'un large choix et va invoquer le service requis d'un simple clic sur ce dernier.



Le clic sur « *Match* » va invoquer à son tour une application Python qui procédera à une découverte syntaxico-sémantique d'une première communauté de substituteurs du service Web choisi.

La fonction syntaxique de la découverte va purger les descriptions textuelles des services présentes dans les descriptions *OWL-S* de ces derniers , purger des mots vides pour ne garder que des termes ou tags pertinents descripteurs, puis un matching entre les tags descripteurs du service requis et ceux des autres services aura lieu, tandis que la fonction sémantique de la découverte va extraire les inputs et outputs de chaque services Web à partir de sa description sémantique puis invoquer un module de matching entre les inputs du services Web requis et ceux des services Web de la collection, les résultats des deux matching seront retournés dans l'interface cliente, les services substituteurs du service Web requis par l'utilisateur ainsi que le poids de leurs liaisons avec ce service Web .

A ce moment-là, la découverte syntaxico-sémantique s'achève avec succès et il est proposé aussi de passer à une découverte sociale de la communauté de substituteurs comme est montré dans la figure suivante :

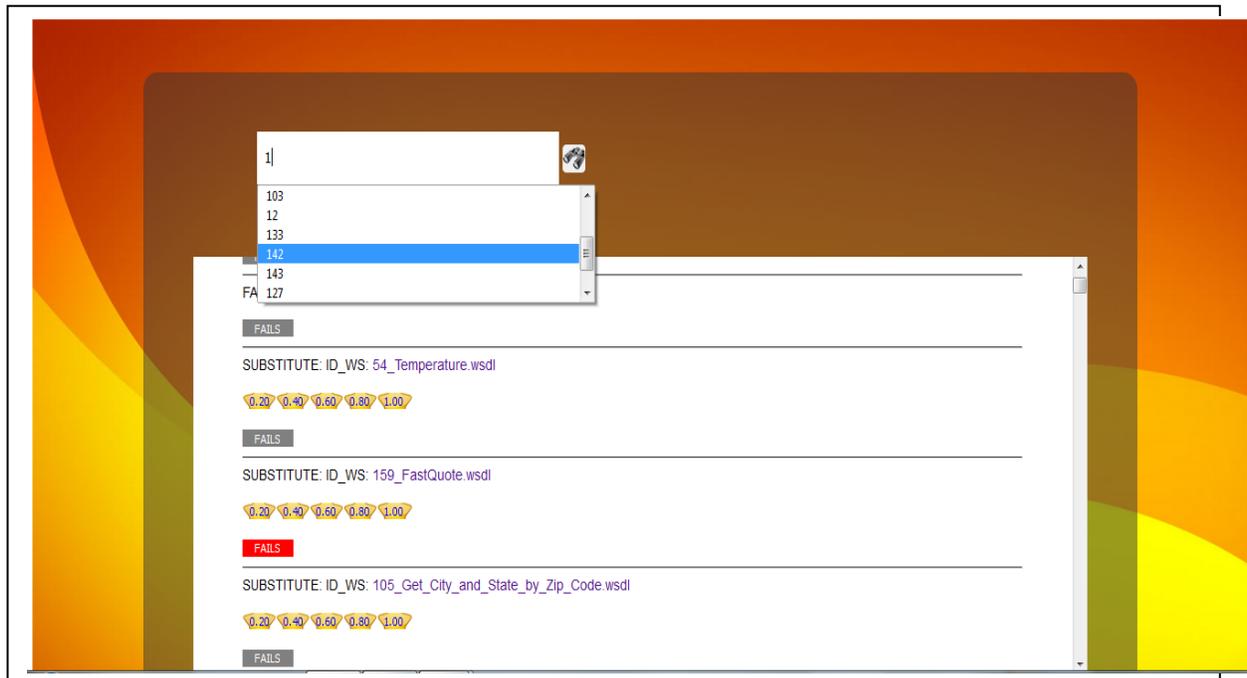


| Services web                         | Degree de similarité |
|--------------------------------------|----------------------|
| 114_ ZipCodeResolver.wsdl            | 0.0526315789474      |
| 4_Prepaid_Calling_Card_Disenser.wsdl | 0.0526315789474      |
| 18_SendEmail.wsdl                    | 0.0526315789474      |
| 141_GermanBankCodes.wsdl             | 0.157894736842       |
| 109_Norwegian_Postal_Codes.wsdl      | 0.0526315789474      |
| 115_ Zip_Code_Lookup.wsdl            | 0.210526315789       |
| 8_SendFaxWebservice.wsdl             | 0.105263157895       |

**Figure 3.5 – communauté des services Web substituteurs issue de la découverte sémantique**

Le clic sur le bouton "view social discovery" va invoquer le service WebJAVA « socialDiscovery » qui aura pour fonction de tracer le comportement social des éléments du réseau social en fonction du temps, dans un premier lieu ce service retourne une communauté composé de tous les service Web ayant une liaison avec le service requis, le poids initial de ces liaisons n'est autre que le poids retourné par la découverte syntaxico-sémantique, et pour capter le comportement social des service Web du réseau social , il est proposé dans l'interface cliente à côté de chaque service substituteur un rating Bar, le clic sur ce dernier voudra dire que ce service vient de réussir une substitution du service Web requis avec un degré de satisfaction de l'invocateur, ceci mettra à jour les paramètre sociaux dans la base de données et invoquera une fonction de mise à jour du poids de la liaison entre le substituteur et le service requis.

Dans ce même axe de capture de l'évolution du réseau social en fonction du temps il est proposé à l'utilisateur un autre bouton « FAIL » à côté du service requis, le clique sur ce bouton voudra dire que ce service vient de tomber en panne, donc sa fiabilité devra être mise à jour dans la base de données, ce qui entrainera l'invocation d'une fonction de mise à jour du poids de la liaison (l'arc) entre le service Web requis et tous ses substituteurs.



**Figure 3.6 – communauté des services Web substituteurs issue de la découverte sociale**

Les principaux paramètres sociaux qui sont révisé au fur et à mesure des sélections et les pannes sont :

- **le degré de similarité** entre deux service Web d'une même liaison, lié au nombre d'arcs entre ces services et au poids de cette relation.
- **Le poids de la liaison** qui dépend du nombre de substitutions réussies d'un service remplaçant avec le service requis, et du nombre de pannes du service requis, donc de la fiabilité de ce service ainsi que du poids précédent calculé par la formule suivante

$$WE_{t+\delta t}(s_i, s_j) = WE_t(s_i, s_j) + \alpha \times \left( \frac{|s_j selection_{t+\delta t}|}{|s_i failure_{t+\delta t}|} - WE_t(s_i, s_j) \right), \quad (6)$$

- **La priorité du cluster** de chaque service remplaçant. En effet un service remplaçant qui entreprend une liaison forte avec le service requis (poids de la relation élevé) est mis dans un cluster dit cluster à forte similarité tandis qu'un service qui entreprend une relation moyennement forte (poids de la relation moyen) est mis dans un cluster à moyenne similarité, le cluster à faible similarité est assujetti au même procédé.
- La priorité des clusters prend tout son sens lors du calcul du **cout de la sélection** d'un service Web  $S_j$  pour remplacer un service Web  $S_i$ . En effet ce coût dépend de la priorité du cluster qui contient le service  $S_j$  ainsi que du poids de liaison entre  $S_i$  et  $S_j$ .

$$O_{s_j} = \frac{P_c}{1 + (p_c \times WE_{t_n}(s_i, s_j))}$$

A noter qu'au fur et à mesure l'évolution du réseau social et des interactions entre services Web, un service Web peut changer de cluster, comme par exemple passer du cluster de moyenne similarité à celui de forte similarité.

La figure suivante montre les valeurs mises à jour de ces paramètres sociaux au sein de la table networkstate de la base de données :

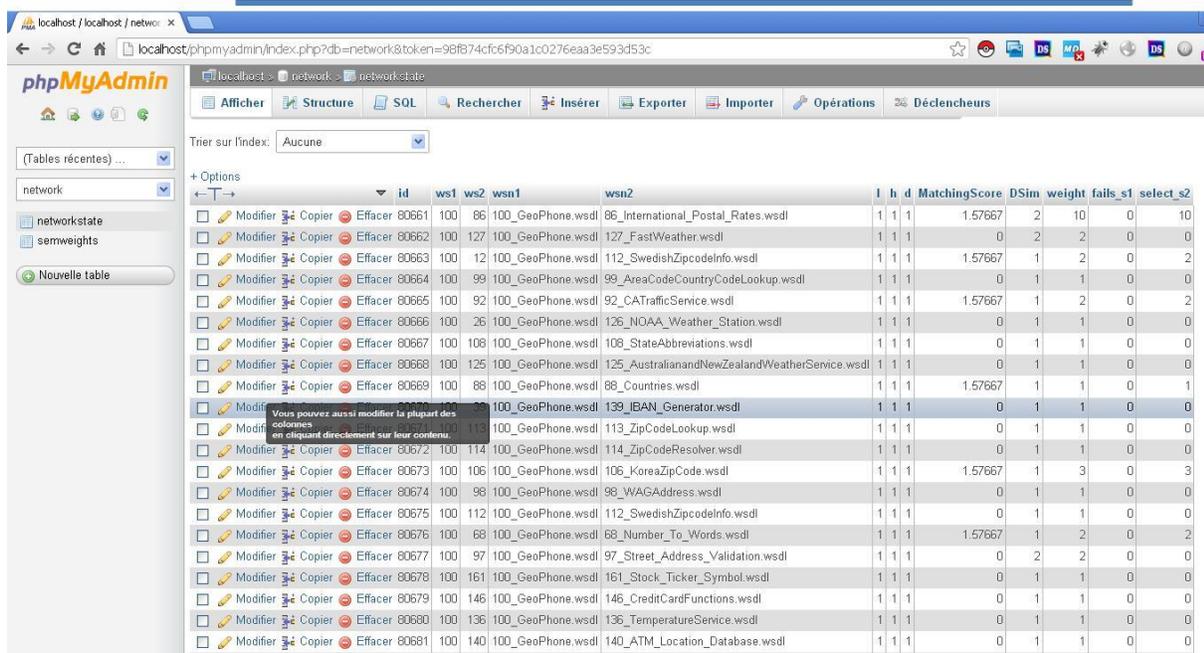


Figure 3.7 – mises à jour des paramètres sociaux dans la base de données

#### 4.4. Analyse des résultats et comparaison entre la découverte sociale et sémantique

Soit le tableau suivant qui montre une partie des résultats obtenus à partir de la base de données :

| Wsn1                    | Wsn2                               | Weight at T0        | ws1 fails | ws2 selections | Weight at t0+Dt |
|-------------------------|------------------------------------|---------------------|-----------|----------------|-----------------|
| 130_Weather_By_Zip_Code | 115_Zip_Code_Lookup                | 2.0                 | 3         | 3              | 0.75            |
| 130_Weather_By_Zip_Code | 132_Weather_Forecast_By_Zip_Code   | 3.0                 | 3         | 5              | 1.25            |
| 130_Weather_By_Zip_Code | 127_FastWeather                    | 2.0                 | 3         | 1              | 0.25            |
| 130_Weather_By_Zip_Code | 114_ZipCodeResolver                | 2.0                 | 3         | 0              | 0               |
| 130_Weather_By_Zip_Code | 131_Weather_Fetcher                | 2.0                 | 3         | 0              | 0               |
| 130_Weather_By_Zip_Code | 105_Get_City_and_State_by_Zip_Code | 3.0                 | 3         | 0              | 0               |
| 130_Weather_By_Zip_Code | 129_Unisys_Weather_Forecast        | 2.0                 | 3         | 6              | 1.5             |
| 130_Weather_By_Zip_Code | 93_USZIPCodeUtility.wsdl           | 2.0                 | 3         | 0              | 0               |
| 130_Weather_By_Zip_Code | 110_Korea_Zip_Code_Service         | 2.0                 | 3         | 2              | 0.5             |
| 130_Weather_By_Zip_Code | 126_NOAA_Weather_Station           | N'est pas découvert | 3         | 5              | 1.25            |
| 130_Weather_By_Zip_Code | 135_World_Weather_Forecast_by_ICAO | N'est pas découvert | 3         | 6              | 1.5             |

Figure 3.8 – comparaison entre les résultats retournés par la découverte sémantique et la découverte sociale

Pour commencer nous remarquons que les poids des relations issus de l'évolution sociale sont nettement plus significatifs que ceux issus de la découverte sémantique et nous expliquons ceci par le fait que la découverte sociale prend en considération des paramètres tels que le nombre de substitution réussis d'un service Web  $S_i$  par un service Web  $S_j$ , le coût de sélection du service Web  $S_j$ , la fiabilité du service Web  $S_i$  (le nombre de pannes ou de non réponse à la requête client), le poids précédent des relations, le degré de similarité entre les service Web, tandis que la découverte sémantique fait abstraction de ces paramètres cruciaux à la pertinence des substitueurs et néglige par ce fait l'évolution dans le temps des interactions sociales. Néanmoins la découverte sociale dépend de la découverte sémantique, à l'instant  $t_0$  les poids des relations sociales ne sont au fait que les poids calculés par la découverte sémantique.

Au fur et à mesure l'expansion du réseau sociale de nouvelles relations apparaissent d'autres disparaissent, de nouveaux poids sont calculés tout ceci est bien cerné par la découverte sociale. Bien plus encore, les données collectées grâce au réseau social seront soumis à du datamining grâce auquel seront dégagées des communautés de services proches de fonctions et de description.

## Conclusion générale et perspectives

### 1. Conclusion

Plusieurs améliorations et extensions peuvent être envisagées pour enrichir les approches de découverte, de sélection et de composition, La majorité des travaux portant sur la découverte négligent l'aspect social. Dans notre travail nous avons essayé de marcher sur les pas de mamar Zakaria et al [] pour implémenter approche pour la découverte des services Web, dite « sociale », cette approche se distingue des approches existantes par le fait qu'elle met l'accent sur l'interaction sociale entre les services Web, leurs comportements en vu des différentes situations, comme elle trace entre ces derniers, des relations de substitutions , de compétition, et de coordination. Pour capter le comportement sociale des services Web, cette approche impose des étapes essentielles à la création d'un réseau social de services Web , nous avons développé un réseau sociale de services Web, en premier lieu nous avons annoté une collection de services Web sémantiquement, ayant comme résultats des services Web sémantiques OWL-S , puis nous avons mis en place une architecture de trois (03) couches pour la découverte , la première sert a charger tous les services web de la collection , la seconde couche s'occupe de la découverte sémantique , elle est constituée d'une application Web qui s'occupe d'extraire les données sémantiques et de les comparer , techniquement parlé, nous avons usé de la puissance de Python dans l'extraction des données à partir de descriptions des services Web sémantique . La 3ème couche est la couche sociale de la découverte, nous avons proposé une implémentation réfléchie qui permet de capter les pannes et les remplacements réussies entre services, une implémentation à base d'un service Web JAVA dynamique , ainsi nous avons raffiné notre maîtrise de l'environnement technique des services Web .

En termes de résultats, nous avons capté les données du réseau social dans une structure de donnée que nous avons par la suite projeté de soumettre à l'analyse, ce support pourra même être proposé dans d'autres applications plus poussées vu sa complétude en termes de paramètres sociaux. L'analyse nous a permis dans un premier lieu de comparer la découverte sociale avec la découverte sémantique ou les résultats retournés

ont montré tout l'avantage d'une découverte sociale temps réel par rapport à une découverte sémantique rigide dans le temps.

Comme perspective de notre travaille nous projetons de générer plusieurs corpus qui vont être utilisé comme des scénarios réels pour alimenter une étude qui va servir à chercher et détecter des communautés cachées dans les réseaux sociaux dégagés par notre solution. Cette étude va se baser sur des techniques issues du domaine de l'analyse des réseaux sociaux et exploiter leurs potentiels. Les communautés résultats de cette étude ont pour rôles de mieux regrouper les services ayant des profils similaires et un historique d'interaction commun afin de cibler les requêtes efficacement et réduire l'ampleur de leur diffusion.

**BIBLIOGRAPHIE**

- [1] *Ethan Cerami*, “Web Services Essentials, Distributed Applications with XML-RPC, SOAP,UDDI & WSDL”. *O'Reilly First Edition February 2002, ISBN: 0-596-00224-6*.
- [2] *Ramesh Nagappan, Robert Skoczylas, Rima Patel Sriganesh*, “Developing Java™ Web Services Architecting and Developing Secure Web Services Using Java”, *Wiley Publishing Inc 2003, ISBN 0-471-23640-3*
- [3] *Patrick Kellert et Farouk Toumani*, “Les Web services sémantiques”. *Revue hors série 2004.disponible depuis: [http://www.revuei3.org/hors\\_serie/annee2004/revue\\_i3\\_hs2004\\_01\\_07.pdf](http://www.revuei3.org/hors_serie/annee2004/revue_i3_hs2004_01_07.pdf). [dernier accès : 31/05/2010]*.
- [4] *Jorge Cardoso*, “Semantic Web Services: Theory, Tools, and Applications”. *University of Madeira, Portugal, Information Science reference, Hershey - New York, 2007.pp 1-20, 134-152, 191-210, 240-312; ISBN 978-1-59904-047-9*.
- [5] *Maamar, Z. Hacid, H. Huhns, M.N.* <<Why Web Services Need Social Networks>>. *IEEE Internet Computing*.vol 15, no 2. p. 90-9 .
- [6] *Ralyté, J. Mirbel, I. Deneckère, R.* 2011. *Engineering Methods in the Service-Oriented Context*.Etats-Unis : Springer. 232 p.
- [7] *ZakariaMaamar, NouraFaci,Leandro ,Krug Wives, YouakimBadr, Pedro Bispo Santos and José Palazzo M. de Olivera* .2011.<<Using Social Networksfor Web Services Discovery>>. *IEEE Internet Computing*, vol 15, no 4. p. 48-54.
- [8] *Manish Gupta, RuiLi,ZhijunYin, Jiawei Han*.2011.<<Social Network Data Analytics>> .*In An Overview Of Social Tagging* ,p.448-497.
- [9] *EytanBakshy,ItamarRosenn,Cameroun Marlow, LadaAdamic*, 28 Feb 2012.*The role of Social Networks in Information Diffusion...*10p
- [10] *Geoffrey Barbier, Huan Liu*.2011. <<Social Network Data Analytics>> .*Datamining in Social Media*,p.328-352.