République **A**lgérienne **D**émocratique et **P**opulaire
Ministère de l'**E**nseignement **S**upérieur et de la **R**echerche **S**cientifique

UNIVERSITE **M**OULOUD **M**AMMERI DE **T**IZI-**O**UZOU

FACULTE DU **G**ENIE **E**LECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE

Mémoire de Fin d'Etudes de MASTER ACADEMIQUE

Domaine : **Mathématiques et Informatique**
Filière : **Informatique**
Spécialité : **Système Informatique**

Présenté par
HIMEUR Imene
KHITOUS Lilia

Thème

Découverte de services web adaptés à l'utilisateur en utilisant l'annotation des services par des tags

Mémoire soutenu publiquement le 08/09/ 2016 devant le jury composé de :

Président : M Idir RASSOUL
Encadreur : M Rachid AHMED-OUAMER
Examinateur : Mme Rachida AOUDJIT
Examinateur : M Mohammed DAOUI

Remerciements

A Mr le professeur Ahmed-Ouamer :

Nous vous sommes extrêmement reconnaissantes d'avoir accepté de nous encadrer pour la réalisation de notre projet de fin d'études, pour l'intérêt que vous avez apporté à notre travail, pour vos conseils précieux, votre soutien constant et la confiance totale que vous nous avez accordée.

Veuillez trouver ici l'expression de notre respectueuse considération et notre profonde admiration.

A notre président de jury Monsieur RassoulIdir,

Vous nous faites l'honneur d'accepter de présider notre jury de thèse.Nous sommes sensibles à l'attention que vous avez bien voulu porter à ce travail. Nous vous exprimons ici l'assurance de notre profond respect.

A notre Examinatrice Madame Aoudjit Rachida,

Nous tenons à vous remercier de nous avoir fait l'honneur d'accepter d'être membre de notre jury et d'avoir accepté de juger ce travail. Veuillez trouver ici l'expression de notre grand respect et nos vifs remerciements.

A notre Examinateur Monsieur Daoui Mohammed,

Nous tenons à vous remercier d'avoir bien voulu participer à l'évaluation de ce travail. Nous sommes honorés de votre présence. Veuillez trouver ici l'expression de notre estime et notre considération.

Dédicaces

Je dédie ce travail à :

Mon Père et à ma Mère« Kamel et Ourdia » qui me sont les plus chères au monde et auxquels je ne saurai jamais exprimer ma gratitude et ma reconnaissance en quelques lignes. Que Dieu vous protège et vous donne santé etlongue vie.

Mes très chers grands-parents maternels « Akli et Tassadite » que Dieu vous protège et vous donne santé et longue vie.

Mon unique et chère sœur « Rania» qui a toujours été là pour moi et qui n'a jamais arrêté de m'encourager et me soutenir.

Mon unique et cher frère: « Rayane » pour son soutien.

Mes très chères tantes maternelles qui m'ont toujours soutenu ainsi que leurs maris respectifs.

Mon oncle Hamza et sa future femme.

Mes très chères tantes paternelles, ainsi que mon oncle.

Mes cousins et cousines.

La mémoire de mon grand-père et ma grand-mère disparus qu'ils reposent en paix.

Mon binôme Imene ainsi qu'à sa famille.

A tous mes amis, particulièrement Selma, Ilham, Soria, Sedik, que je remercie pour leurs soutient et avec lesquels j'ai passé mes plus beaux jours.

« Lilia »

Je dédie ce travail à :

La mémoire de mon grand père Mohammed.

La mémoire de ma grand-mère Khedoudja.

Mes très chers parents, Ali et Nora, qui ont toujours été là pour moi, « Vous avez tout sacrifié pour vos enfants n'épargnant ni santé ni efforts. Vous m'avez donné un magnifique modèle de labeur et de persévérance. Je suis redevable d'une éducation dont je suis fière ».

Ma très chère sœur Katia qui m'a énormément guidée et soutenue ainsi qu'à son mari Nouredine.

Mes chers frères Mohamed et Amayas qui m'ont énormément encouragé.

Mes meilleurs Sifax et Nibel qui ont toujours crus en moi.

Mon binôme Lilia ainsi que sa famille.

Tous mes amis pour leurs soutiens et leurs encouragements.

Et tout autre personne ayant contribué de près où de loin à la réalisation de ce modeste travail.

«Imene»

Découverte de services web adaptés a l'utilisateur en utilisant l'annotation des services par des tags

Résumé :

La multiplication rapide des services web publiés sur internet rend leur découverte plus compliqué. Pour résoudre ce problème de découverte, la plus part des approches proposés se basent uniquement sur la description du service sans tenir compte du retour d'usage. En effet l'annotation par des tags (ou mots clés) des services peut indiquer l'avis des utilisateurs sur les fonctionnalités ou la qualité des services utilisés. Cette opinion peut donc être exploitée lors du processus de découverte de services web.

L'objectif du sujet est d'exploiter l'activité d'annotation des services par des tags pour améliorer le processus de découverte de services et permettre a l'utilisateur l'accès le plus pertinent aux ressources.

La recherche de services web utilisera des requêtes simples ou composées. L'exploitation des ontologies pour la désambigüisation sémantique des tags sera traitée afin de renforcer la similitude entre une requête utilisateur et les services web annotés. Le résultat du processus de découverte de services web sera donné sous la forme d'une liste de services classés selon le poids des tags qui leur seront associés.

Il serait enfin utile d'indiquer l'influence du poids des tags et son impact sur la qualité de découverte et la popularité du service.

Mots clés : Service web – Web sémantique – Découverte de services – Annotation sémantique – Ontologie.

Abstract

The fast development of web services published on the Internet makes their discovery more complicated. To solve this problem of discovery, most of the approaches proposed are based only on the description of the service without using the return of use. Indeed the annotation by tags (or keywords) of services may indicate user feedback on features or quality of the services used. This opinion can be exploited during the service discovery step.

The aim of this thesis is to use the annotation tags, to improve the service discovery and allow the user the most relevant resource access.

Ontology are used to disambiguate the tags in order to strengthen the similarity between a user request and the annotated web services. The result of the Web service discovery,will be displayed as a list of services classified by them tag's weight.

At last we indicated the influence of tag's weight and its impact on services popularity.

Keywords: Web Service – semantic web– Discovery Service–semantic Annotation – Ontology.

Sommaire

Introduction Générale:

Chapitre I :

Chapitre II :

Chapitre III :

Liste des figures :

Chapitre I :

Chapitre II :

Chapitre III :

Chapitre IV :

Liste des Tableaux

Chapitre III :

Introduction générale

Introduction Générale

Contexte :

L'essor du Web, conforté par le rapide développement des nouvelles technologies de l'information et de la communication a conduit à la production d'un volume d'information sans précédent, et la quantité d'information manipulée par diverses organisations dans le monde ne cesse de croitre.

Les services Web sont devenus le facteur le plus significatif technologique par produit, ils sont des briques de bases logicielles s'affranchissant de toute contrainte de compatibilité logicielle ou matérielles, ils appartiennent à des applications capables de collaborer entre elles de manière transparente pour l'utilisateur. Leur mise en œuvre repose sur une architecture décentralisée. Les services sont publiés dans des annuaires par des fournisseurs qui les hébergent. Ils sont accessibles via un réseau pour les clients qui les découvrent, les sélectionnent, les invoquent et les utilisent.

Le problème de la recherche dans les services Web a attiré l'attention des chercheurs au cours de la dernière décennie. La raison à cela est que la technologie évolue et que beaucoup de services sont disponibles, il devient important d'être en mesure de localiser les services qui répondent à nos besoins dans un grand nuage dense d'offres. Plusieurs propositions ont été avancées pour résoudre ce problème et plusieurs normes ont été définies, mais aucun d'entre eux a été efficace ou est maintenant acceptée comme le moyen d'effectuer une recherche de service.

Problématique:

Aujourd'hui, la problématique qui se pose est celle d'une recherche d'information intelligente sur le Web. Le Web Sémantique est un espace d'échange qui reste à construire. Un de ses intérêts est d'une part d'apporter suffisamment de renseignements sur les ressources, en ajoutant des annotations sous la forme de métadonnées et d'autre part de décrire leur contenu de manière à la fois formelle et signifiante à l'aide d'une ontologie pour être interprétables aussi bien par les humains que par les machines.

Contribution :

Dans ce mémoire nous allons proposer une approche qui exploite les annotations par des tags(mots-clés) des services web qui peuvent indiquer l'avis de l'utilisateur sur les fonctionnalités ou la qualité des services utilisés dans le but d'améliorer le processus

de découverte de ces derniers et permettre a l'utilisateur l'accès le plus pertinent aux ressources.

Des ontologies seront exploitées pour la désambigüisation sémantique des tags et les résultats seront filtrés selon leurs poids et leurs impacts sur la qualité et la popularité des services.

Plan du mémoire :

Notre travail débute par une introduction générale qui présente le travail à faire ainsi que l'organisation suivie pour sa réalisation.

Ce mémoire est structuré selon 4 chapitres :

Le chapitre 1: Ce chapitre présente la technologie des services Web et les principaux Standards qu'elle supporte, parmi lesquels nous citons les protocoles SOAP, WSDL, UDDI. Nous exposons ensuite l'utilisation des technologies du Web sémantique (i.e. l'ontologie, l'annotation sémantique, le raisonnement sémantique).

Le chapitre 2: Celui-ci traitera tout ce qui est en rapport avec l'utilisation des services web : publication, annotation et découverte des services web.

Le chapitre 3: Dans celui-ci nous présentons la conception détaillée de notre solution (différents diagrammes UML).

Le chapitre 4: Enfin ce dernier est consacré à la présentation de l'application réalisée.

Et pour finir une conclusion générale qui résume notre travail et les perspectives à venir.

Chapitre I

Partie 1 :

I. Introduction :

En quelques années, les services web sont devenus le nouveau point de convergence technologique de l'industrie du logiciel. Un bon nombre d'éditeurs de serveurs d'applications, d'outils de développement, ont placé ces services web au cœur de leur stratégie. Dans ce chapitre, nous décrivons les technologies liées aux services web.

II. Définition et description des services web :

Il s'agit d'une technologie permettant à des applications de dialoguer à distance via Internet, et ceci indépendamment des plates-formes et des langages sur lesquelles elles reposent. Pour ce faire, les services Web s'appuient sur un ensemble de protocoles Internet très répandus (XML, HTTP), afin de communiquer. Cette communication est basée sur le principe de demandes et réponses, effectuées avec des messages XML. **[1]**

III. L'intérêt des services web :

Le but principal est de fournir une architecture générale basée sur des standards ouverts pour les applications réparties sur internet :

- ✓ Interopérables
- ✓ Sans composant spécifique à un langage ou un système d'exploitation
- ✓ Faiblement couplées : limiter au maximum les contraintes imposées sur le modèle de programmation des différents éléments de l'application par exemple ne pas imposer un modèle objet
- ✓ « Déplorables » à large échelle : le web. **[2]**

IV. L'architecture générale d'un service web :

Jusqu'ici, l'accès via Internet à une ressource applicative ou à une base de données s'effectuait par l'envoi d'une requête s'appuyant sur des langages tels que (PHP, JSP, …). Il s'agissait donc d'un dialogue entre une couche de présentation

reposant sur HTML (protocole http) et des applications installées sur un serveur distant.

Avec les services Web, un dialogue est désormais instauré entre applications qui peuvent être installées sur des machines distantes, et ceci grâce à des standards XML. En effet, afin de dialoguer via Internet, ces applications doivent « parler » le même langage, langage basé sur le XML.

Dans la **figure (I.1)**, l'architecture de référence des services Web se base sur les trois concepts suivants :

- ✓ Le fournisseur de service : c'est le propriétaire du service.
- ✓ Le client (ou le consommateur de service) : c'est un demandeur de service. D'un point de vue technique, il est constitué par l'application qui va rechercher et invoquer un service.
- ✓ L'annuaire des services : c'est un registre de descriptions de services offrant des facilités de publication de services à l'intention des fournisseurs ainsi que des facilités de recherche de services à l'intention des clients.

Les interactions de base qui existent entre ces trois éléments sont les opérations de publication, de recherche, d'invocation et de lien (bind).

Pour bien comprendre le fonctionnement de cette architecture, nous expliquons ci-dessous le rôle de chacun des éléments précédents :

- ✓ Le fournisseur de service créé le service Web, puis publie son interface ainsi que les informations d'accès au service, dans un annuaire de services Web.
- ✓ L'annuaire de service rend disponible l'interface du service ainsi que ses informations d'accès, pour n'importe quel demandeur potentiel de service.
- ✓ Le consommateur de service accède à l'annuaire de service pour effectuer une recherche afin de trouver les services désirés. Ensuite, il se lie au fournisseur pour invoquer le service. **[2]**

Figure I.1. Structure des services Web

V. Les principales technologies :

Une caractéristique qui a permis un grand succès de la technologie des services web est qu'elle est construite sur des technologies standards de l'industrie. Dans cette section il y a une description de ces technologies.

V.1. XML (eXtensible Markup Language):

V.1.1. Définition :

Le XML, acronyme de eXtensible Markup Language (qui signifie: langage de balisage extensible), est un langage informatique qui sert à enregistrer des données textuelles. Ce langage a été standardisé par le W3C en février 1998 et est maintenant très populaire. Ce langage, grosso-modo similaire à l'HTML de par son système de balisage, permet de faciliter l'échange d'information sur l'internet.

Contrairement à l'HTML qui présente un nombre finit de balises, le XML donne la possibilité de créer de nouvelles balises à volonté. **[3]**

V.1.2. XML Namespaces:

XML Namespaces est une recommandation du W3C, visant à résoudre le problème de l'ambiguïté des balises dans un document XML.

Les namespaces permettent ainsi de résoudre le problème des différences d'interprétation du même document XML par des applications différentes.

En s'appuyant sur le dispositif des URI, qui en assure l'unicité, et au prix d'une écriture un peu plus« bavarde», les balises et les attributs XML sont alors dotés d'une interprétation spécifique, nom ambiguë.

V.1.3. XML schéma :

XML Schéma publié comme recommandation par le W3C est un langage de description de format de document XML permettant de définir la structure et le type de contenu d'un document XML. Cette définition permet notamment de vérifier la validité de ce document.

Il est possible de décrire une organisation de vocabulaires d'origines différentes, par l'usage des espaces de noms. Il est possible de combiner les schémas eux-mêmes, et d'exprimer une combinaison pour le document contenu.

Une instance d'un *XML Schema* est un peu l'équivalent d'une *définition de type de document* (DTD). XML Schema amène cependant plusieurs différences avec les DTD : il permet par exemple de définir des domaines de validité pour la valeur d'un champ, alors que cela n'est pas possible dans une DTD ; en revanche, il ne permet pas de définir des entités, XML Schema est lui-même un document XML, alors que les DTD sont des documents SGML. **[4]**

V.1.4. Document Type Definition (DTD) :

La *document* type définition (DTD), ou définition de type de document, est un document permettant de décrire un modèle de document SGML (**Standard Generalized Markup Language**) ou XML. Le modèle est décrit comme une grammaire de classe de documents : *grammaire* parce qu'il décrit la position des termes les uns par rapport aux autres, *classe* parce qu'il forme une généralisation d'un domaine particulier, et *document* parce qu'on peut former avec un texte complet.

V.1.5. Avantages du XML :

- **Lisibilité :** il est facile pour un humain de lire un fichier XML car le code est structuré et facile à comprendre. En principe, il est même possible de dire qu'aucune connaissance spécifique n'est nécessaire pour comprendre les données comprises à l'intérieur d'un document XML.

- **Disponibilité :** ce langage est libre et un fichier XML peut être créé à partir d'un simple logiciel de traitement de texte (un simple bloc-notes suffit).
- **Interopérabilité :** Quelques soit le système d'exploitation ou les autres technologies, il n'y a pas de problème particulier pour lire ce langage.
- **Extensibilité :** De nouvelles balises peuvent être ajoutée à souhait.
- Plusieurs parseurs XML différent doivent en principe (s'ils sont bien codés) produire le même résultat.
- Tous les navigateurs internet récents intègrent un parseur XML, pour lire les documents de ce langage informatique. **[5]**

V.2. SAOP (Simple Object Access Protocol) :

SOAP (Simple Object Access Protocol) est un protocole d'invocation de méthodes sur des services distants. Basé sur XML, SOAP est un format de communication pour assurer communication de machine à machine. Le protocole permet d'appeler une méthode RPC (Remote Procedure Call) et d'envoyer des messages aux machines distantes via HTTP. Ce protocole est très bien adapté à l'utilisation des services web car il permet de fournir au client une grande quantité d'informations récupérées sur un réseau de serveurs tiers.

SOAP permet donc l'échange d'information dans un environnement décentralisé et distribué, comme Internet par exemple. Il permet l'invocation de méthodes, de services, de composants et d'objets sur des serveurs distants et peut fonctionner sur de nombreux protocoles (des systèmes de messagerie à l'utilisation de RPC). Cependant, il fonctionne particulièrement bien avec le protocole HTTP, protocole très souvent utilisé avec SOAP.

SOAP utilise les protocoles HTTP et XML. HTTP comme mécanisme d'invocation de méthodes en utilisant un des balises spécifiques pour indiquer la présence de SOAP comme «protocole ». Cela permet de franchir aisément les firewalls et proxy et facilite le traitement en cas de filtrage. XML pour structurer les requêtes et les réponses, indiquer les paramètres des méthodes, les valeurs de retours, et les éventuelles erreurs de traitements.

XML joue un rôle prépondérant dans SOAP, on peut distinguer plusieurs éléments :

- ✓ une enveloppe, expliquant comment la requête doit être traitée et présentant les éléments contenus dans le message.

- ✓ un ensemble de règles de codage, permettant de différencier les types de données transmises.
- ✓ une convention de représentation, permettant de représenter les appels aux procédures de traitement et les réponses.

Même si SOAP ressemble beaucoup en termes de fonctionnalités à des protocoles comme IIOP pour CORBA, ORPC pour DCOM, ou JRMP pour JAVA, il possède un avantage indéniable. En effets, SOAP utilise un mode texte alors que les autres fonctionnent en mode binaire, cela facilite le passage des équipements de sécurité. **[6]**

V.2.1. Structure d'un message SOAP :

SOAP définit un format pour l'envoi des messages. Les messages SOAP sont structuré en un document XML et comporte 2 éléments obligatoires : Une enveloppe et un corps (une entête facultative).

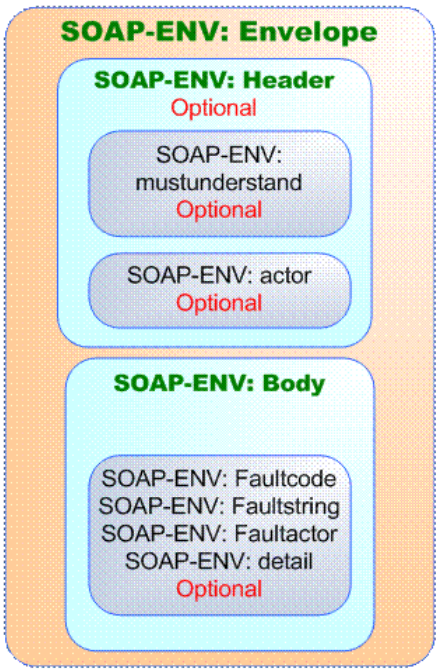

Figure I.2. Structure d'un message Soap

V.2.1.1. L'enveloppe SOAP :

L'enveloppe contient l'espace de nommage définissant la version de SOAP utilisée, et les règles de sérialisation, et d'encodage.

V.2.1.2. Le header SOAP :

Cette partie du message est optionnelle. Elle sert à transmettre des informations nécessaires pour l'exécution de la requête SOAP aux dictionnaires qui recevront le message. On y précise généralement des informations liées aux transactions, à l'authentification, etc.

Le header est composé d'un ou plusieurs champs : l'attribut actor, désignant le destinataire du header, l'attribut mustUnderstand, qui indique si le processus est optionnel.

L'attribut actor permet de préciser l'application à laquelle est destinée l'information contenue dans le header. L'URI http://schemas.xmlsoap.org/soap/actor/next précise en particulier que ces informations sont destinées à la première application qui reçoit le message. Dans le cas où l'actor n'est pas précisé, le header est analysé par le destinataire final du message.

V.2.1.3. Le Body SOAP :

Le body SOAP contient toutes les informations que l'on veut transmettre à l'application distante. Le contenu du Body est normalisé dans SOAP RPC, pour modéliser une requête et sa réponse. Le body de la requête contient l'identifiant de l'objet distant, le nom de la méthode à exécuter et les éventuels paramètres. Le body de la réponse contient le résultat de l'exécution de la requête.

V.2.2. SOAP RPC :

SOAP RPC définit les conventions permettant d'utiliser SOAP comme un RPC, et le format des messages pour effectuer une requête ou envoyer une réponse. SOAP RPC se base sur les spécifications de XML-RPC.

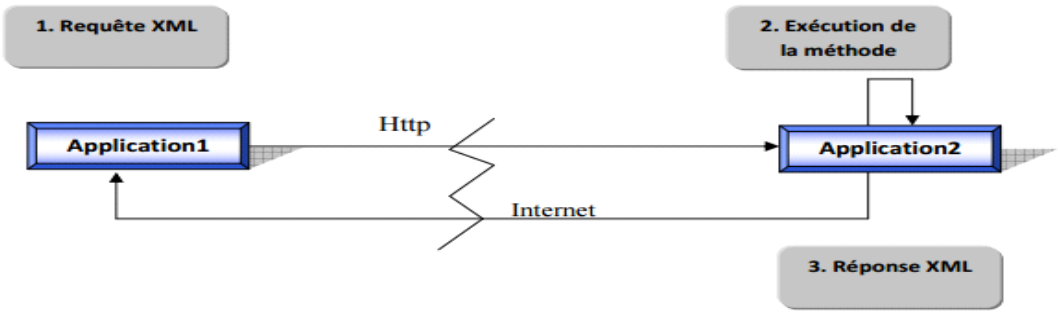

Figure I.3. SOAP RPC

V.3. Apache-Axis: Une mise en œuvre de SOAP:

Axis est essentiellement un moteur de SOAP. Une structure pour construire des processus de SOAP comme des clients, des serveurs, des « *Gateway*», etc.

La version actuelle d'Axis est écrite en Java. Mais il n'est pas juste un moteur de SOAP, il inclut aussi :

- ✓ Un serveur autonome simple.
- ✓ Un serveur que branche dans des moteurs servlet comme Tomcat.
- ✓ Appui vaste pour WSDL.
- ✓ L'outillage d'émetteur (i.e. WSDL2Java) qui produit des classes Java à partir du fichier WSDL.
- ✓ L'outillage d'émetteur (i.e. Java2WSDL) qui produit le fichier WSDL à partir de la classe de l'interface Java.
- ✓ Un outil pour surveiller des paquets de TCP/IP, et
- ✓ Quelques programmes exemple. **[7]**

V.4. WSDL (Web Services Definition Language):

WSDL est un langage qui permet de décrire les services web, et en particulier, les interfaces des services web. Ces descriptions sont documents XML. WSDL décrit un service web en deux étapes fondamentales : une abstraite et une concrète. Dans chaque étape, la description utilise un nombre constructions pour favoriser la réutilisation de la description et pour séparer les préoccupations de conception indépendantes. **[8]**

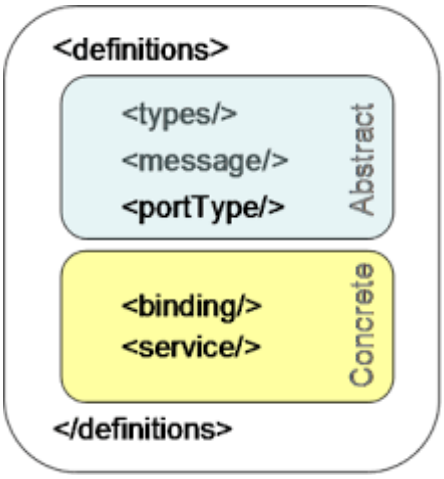

Figure I.4. La spécification d'un service Web avec WSDL

V.5. UDDI (Universal Description, Discovery and Integration Service):

Pour utiliser un Web Services, il faut premièrement savoir s'il existe. UDDI (Universal Description, Discovery and Integration Service) est la norme qui définit le mécanisme pour découvrir dynamiquement des services. Un client pointe vers un registraire UDDI, qui lui donnera la définition du service recherché. Le registraire UDDI est lui-même un Web Service qu'un client peut questionner.

Ainsi, lorsque l'on veut mettre à disposition un nouveau service, on crée un fichier appelé Business Registry qui décrit le service en utilisant un langage dérivé d'XML suivant les spécifications UDDI.

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

- ✓ **Les pages blanches** incluant l'adresse, le contact et les identifiants relatifs aux services Web.
- ✓ **Les pages jaunes** identifiant les secteurs d'affaires relatifs aux services Web.
- ✓ **Les pages vertes** donnant les informations techniques.

En utilisant l'API UDDI l'utilisateur peut alors stocker ces informations dans un nœud (node) UDDI. Elles sont ensuite répliquées de nœud en nœud (principe assez proche dans l'idée des réplications de DNS).

Une fois ceci est fait, le service Web peut alors être connu de tous ceux qui le recherchent. Le modèle UDDI comporte 5 types de structures de données décrites sous forme de schéma XML. **[8]**

- ✓ **Business entity:** elle contient des informations sur l'entreprise qui publie les services dans l'annuaire, cette structure contient les autres éléments de l'UDDI.
- ✓ **Business service:** ensemble d'informations sur les services publiés par l'entreprise (nom du service, les catégories…).
- ✓ **Binding Template:** ensemble d'informations sur le lieu d'hébergement du service (c.à.d. l'adresse du point d'accès du service).
- ✓ **Tmodel:** ensemble d'informations sur le mode d'accès au service (définitions wsdl), il peut s'agir aussi d'une spécification abstraite ou d'une taxonomie.
- ✓ **Publisher Assertion:** ensemble d'informations contractuelles entre les partenaires.

VI. Avantages et inconvénients des services web :

VI.1. Avantages:

- Ils sont indépendants des plateformes et des langages (XML)
- La plupart des services web utilisent le protocole HTTP pour transmettre les messages entre clients et serveur d'autres protocoles comme TCP/IP, SMTP, JMS sont possibles.
- Ils autorisent un couplage faible entre le client et le serveur le client n'a pas connaissance du service web jusqu'à ce qu'il l'utilise.
- Des mécanismes de découverte sont prévus un client peut déterminer le fournisseur le plus apte à effectuer une action voulue.
- La technologie Web Service met en œuvre des fonctionnalités d'auto-description des fournisseurs et des services qui leurs sont associés l'utilisation d'un fournisseur de services web peut être gérée de manière dynamique. **[9]**

VI.2. Les inconvénients :

- Les services web sont un concept, leur implémentation étant laissée libre dès le départ.
- Il existe aujourd'hui plus de 50 spécifications différentes concernant les services web, dont certaines incompatibles entre elles manquent de polyvalence.
- Les services web ne proposent que des invocations de services simples
- Les technologies CORBA et EJB proposent des services de cycle de vie, de persistance, de transaction, de sécurité surcoût dans la transmission des informations.
- Le transfert de données en XML est moins efficace qu'en binaire. **[9]**

Partie 2 :

VII. Introduction au web sémantique :

La recherche d'information (précision et complétude) : rechercher des documents sur le Web est souvent une tâche laborieuse. Les recherches sont imprécises et requièrent une activité de « tri manuel » des documents retournés pour espérer trouver le(s) document(s) recherché(s) d'ailleurs sans aucune assurance. Ici, le Web sémantique devrait largement faciliter l'appariement sémantique entre la requête de l'utilisateur et les documents indexés (manuellement ou de manière semi-automatique).

VIII. Définition du web sémantique:

Le Web sémantique, ou toile sémantique, est une extension du Web standardisée par le World Wide Web Consortium(W3C). Ces standards encouragent l'utilisation de formats de données et de protocoles d'échange normés sur le Web, avec comme format de base le Resource Description Framework (RDF).

Selon le W3C, le Web sémantique fournit un modèle qui permet aux données d'être partagées et réutilisées entre plusieurs applications, entreprises et groupes d'utilisateurs ». L'expression a été inventée par Tim Berners-Lee (inventeur du Web et directeur du W3C), qui supervise le développement des technologies communes du Web sémantique. Il le définit comme « une toile de données qui peuvent être traitées directement et indirectement par des machines pour aider leurs utilisateurs à créer de nouvelles connaissances ».

Pour y parvenir, le Web sémantique met en œuvre le Web des données qui consiste à lier et structurer l'information sur Internet pour accéder simplement à la connaissance qu'elle contient déjà. **[10]**

IX. Objectifs du web sémantique :

Un des principaux objectifs du Web sémantique est de permettre aux utilisateurs d'utiliser la totalité du potentiel du Web : ainsi, ils pourront trouver, partager et combiner des informations plus facilement. Aujourd'hui tout le monde est

capable d'utiliser des forums, d'utiliser des réseaux sociaux, de chatter, de faire des recherches ou même d'acheter différents produits. Néanmoins, il serait mieux que la machine fasse tout ceci à la place de l'homme, car actuellement, les machines ont besoin de l'homme pour effectuer ces tâches.

La raison principale est que les pages Web actuelles sont conçues pour être lisibles par des êtres humains et non par des machines. Le Web sémantique a donc comme principal objectif que ces mêmes machines puissent réaliser seules toutes les tâches fastidieuses comme la recherche ou l'association d'informations et d'agir sur le Web lui-même. **[10]**

X. Le Web sémantique souvent critiqué:

Comme vous avez pu le deviner, le Web sémantique permet à tous d'en savoir plus sur les sujets qui les intéressent. Cela signifie a contrario que diverses institutions peuvent récolter des informations sur vous librement et donc constituer des dossiers. Ces institutions peuvent être des agences de publicité, de sécurité ou même des services de renseignements. Et ceci tout en restant dans la légalité, car dans 90 % des cas, ces informations sont mises en ligne volontairement ou non par les utilisateurs, sans qu'ils se doutent du « danger » que peut représenter cette action.

Le Web sémantique est aussi critiqué à cause de sa lourdeur : les langages utilisés pour le Web sémantique sont très verbeux, car dérivés du XML et donc souvent pénibles à utiliser. De ce fait, l'écriture d'ontologies est souvent très problématique, car elle exige une spécialisation dans un domaine particulier et, lorsque l'on ne maîtrise pas ce domaine, elle devient très difficile à créer. Ainsi, certaines personnes disent qu'il est préférable d'utiliser des « words tags » (ce sont une série de mots clés qui permettent de qualifier une ressource) à la place des ontologies. **[10]**

XI. Composants principaux du Web sémantique :

XI.1. LES METADONNEES :

Les métadonnées sont définies en tant que données ou informations concernant d'autres données. Berners-Lee donne toutefois une définition des métadonnées qui est mieux adaptée au cadre du Web sémantique10: « *(dans la*

conception Web)… les métadonnées représentent des informations compréhensibles par des ordinateurs et qui concernent des ressources Web ou d'autres choses »
[11]. Cette définition nous amène vers le concept de données compréhensibles par un ordinateur.

En effet, le but ultime des métadonnées identifié par Berners-Lee est la production d'informations qui s'auto-décrivent. Pour ce faire, les métadonnées doivent contenir une référence vers une spécification des sémantiques qui leurs sont associées. **[11]**

La sémantique des ressources du Web actuel doit donc être rendue compréhensible pour les ordinateurs. Pour ce faire, il faut déterminer une représentation formelle et standardisée qui puisse être utilisée par les machines lors de l'échange d'informations. Il s'agit donc de spécifier un vocabulaire (ou des *sémantiques*) à utiliser. Dans le cas du Web sémantique, cette spécification repose sur l'usage d'ontologies. **[12]**

XI.2. Ontologie :

XI.2.1. Définition de l'ontologie:

En philosophie, l'ontologie est l'étude de l'être en tant qu'être, c'est-à-dire l'étude des propriétés générales de ce qui existe.

Par analogie, le terme est repris en informatique et en science de l'information, où une ontologie est l'ensemble structuré des termes et concepts représentant le sens d'un champ d'informations, que ce soit par les métadonnées d'un espace de noms, ou les éléments d'un domaine de connaissances. L'ontologie constitue en soi un modèle de données représentatif d'un ensemble de concepts dans un domaine, ainsi que des relations entre ces concepts. Elle est employée pour raisonner à propos des objets du domaine concerné. Plus simplement, on peut aussi dire que l'ontologie est aux données ce que la grammaire est au langage.

Les concepts dans une ontologie représentent les objets, notions ou les idées du domaine.

Les relations représentent les liens conceptuels pouvant exister entre les concepts. **[12]**

XI.2.2. Types d'ontologie :

Les méthodes en Ingénierie des connaissances ont répertoriés plusieurs types d'ontologie liés à l'ensemble des objets conceptualisés et manipulés.

Ces ontologies sont :

XI.2.2.1. Ontologie de domaine:

Une ontologie de domaine est la représentation d'un domaine particulier de connaissances. Elle exprime des conceptualisations spécifiques à un domaine. Elle regroupe les concepts et les relations permettant de couvrir les connaissances d'un domaine.

La plus part des ontologies disponibles sur le Web sont des ontologies de domaines. A titre d'exemple, l'ontologie de domaine médicale, l'ontologie du domaine du tourisme et celles de domaine de la bibliographie et l'anatomie humaine, etc.

XI.2.2.2. Ontologie générique :

Un critère également utilisé pour caractériser une ontologie est sa granularité, c'est-à-dire le niveau de détails de sa conceptualisation. En fonction de l'objectif opérationnel, une connaissance plus ou moins fine du domaine est nécessaire, et des propriétés considérées comme accessoires dans certains contextes peuvent se révéler indispensables pour d'autres applications.

Les ontologies génériques ou de haut niveau ont une granularité large, du fait que les notions sur lesquelles elles portent peuvent être raffinées par des notions plus spécifiques. Elle regroupe les concepts les plus abstraits du domaine. Elles sont conçues pour les réutiliser, les raffiner en cas de besoins et les intégrer dans différentes applications.

XI.2.2.3. Ontologie de tâche :

C'est une ontologie spécifique pour résoudre un problème bien définit comme les problèmes de planification et de diagnostic. Elle offre un ensemble de termes au moyen desquelles nous pouvons décrire généralement comment résoudre un type de problèmes. Elle contient des noms, des verbes et des adjectifs pour la description de tâches.

XI.2.2.4. Ontologie de représentation :

Spécifie un formalisme de description qui fournit une structure de représentation et des primitives pour décrire les concepts des ontologies de domaine et des ontologies génériques.

XI.3. Annotation:

XI.3.1. Quelques définitions:

Selon le dictionnaire Larousse, l'annotation est l'action de faire des remarques sur un texte pour l'expliquer ou le commenter.

Une annotation est une information graphique ou textuelle attachée à un document et le plus souvent placée dans ce document. Cette place est donnée par une ancre qui peut être un ensemble de documents, un document, un passage, une phrase, un terme, un mot, une image.

On peut aussi choisir une définition de l'annotation réduite à ses éléments les plus significatifs comme suit: « Une annotation est une note particulière attachée à une cible celle-ci peut être une collection de documents, un document, un segment de document (paragraphe, groupe de mots, mot, image ou partie d'image, etc.) ou une autre annotation. À une annotation correspond un contenu matérialisé par une inscription qui est une trace de la représentation mentale que l'annotateur se fait de la cible. Le contenu de l'annotation pourra être interprété à son tour par un autre lecteur. Nous appelons l'ancre ce qui lie l'annotation à la cible (un trait, un passage entouré, etc.). »

D'où on conclut que les éléments de l'annotation sont : La cible, l'ancre et l'information de l'annotation. **[12]**

XI.3.1.1. La cible de l'annotation:

C'est l'objet annoté. La cible peut être une collection de documents, un document, une partie du document, une phrase, un terme ou mot, une image ou une partie d'image, un son, une vidéo ou une autre annotation.

XI.3.1.2. L'information de l'annotation:

C'est l'information assignée à l'objet annoté. Elle peut prendre plusieurs formes:

- ✓ des icônes (par exemple pour décrire des avis en utilisant des étoiles, des points d'interrogation...),
- ✓ des symboles de liens (pour décrire des associations, des relations entre mots, paragraphes ou chapitres),
- ✓ des notes textuelles en marge, en bas de page ou en fin de documents,
- ✓ icônes (numéros, étoiles...),
- ✓ des mises en forme typographiques (sur lignage, soulignage, italique...),
- ✓ des redécoupages de texte (à l'aide d'accolades, de numérotation de passages...),
- ✓ des images,
- ✓ des sons...

XI.3.1.3. L'ancre de l'annotation:

C'est le point qui attache l'annotation à l'objet annoté. Elle peut être :

- ✓ Multi-cibles (porte sur plusieurs cibles simultanément) ou uni-cible.
- ✓ Explicite (L'ancre désigne explicitement sur quoi porte l'annotation) ou tacite.
- ✓ Conventionnelle (Le consultant connait, par convention, grâce à la forme de l'ancre, la cible de l'annotation et l'emplacement de l'information de l'annotation) ou non conventionnelle.

XI.3.2. Les annotations sémantiques :

Dans le cadre du web sémantique se trouve un cas très particulier d'annotation: les annotations sémantiques. Elles sont utilisées dans les théories de classification, le résumé automatique de documents est plus généralement pour favoriser l'interopérabilité.

Le Web Sémantique est une infrastructure organisée dans un empilement de couches. L'annotation sémantique se situe au cœur de cet empilement. Elle ne présente pas une couche mais elle fait partie intégrante de la couche ontologie. Elle se concrétise généralement dans la phase d'instanciation de l'ontologie. Les instances produites représentent l'information pertinente dans le document. La tâche d'annotation consiste donc à prendre en entrée une ressource documentaire et fournir en sortie le même contenu enrichi par des annotations sémantiques basées sur des représentations de la connaissance plus ou moins formelles.

Nous définissons l'annotation sémantique comme une représentation formelle d'un contenu exprimée à l'aide de concepts, de relations et d'instances décrits dans une ontologie.

Elles sont le plus souvent reliées à la ressource documentaire source et ne possèdent pas d'ancrage particulier. Elles sont des annotations opérationnelles car elles sont destinées à être traitées par des machines (par opposition aux annotations libres en langage naturel ou composées de symboles souvent tacites). Les annotations sont aussi potentiellement intéressantes dans la recherche d'information.

Les annotations des documents et la requête sont exprimées en utilisant le vocabulaire de l'ontologie, le moteur de raisonnement recherche dans les annotations les résultats correspondant à la requête en exploitant les inférences contenues dans l'ontologie.

Les annotations sémantiques ont donc pour objectif d'exprimer la «sémantique» du contenu d'une ressource pour:

- ✓ Désambiguïser le document pour un traitement automatique,
- ✓ Améliorer sa compréhension, sa recherche et donc sa réutilisation par des utilisateurs finaux**.**

XII. Architecture des web sémantique:

La vision courante du Web sémantique proposée par [Berners-Lee et al., 2001] peut être représentée dans une architecture en plusieurs couches différentes comme le montre la **figure II.1**.

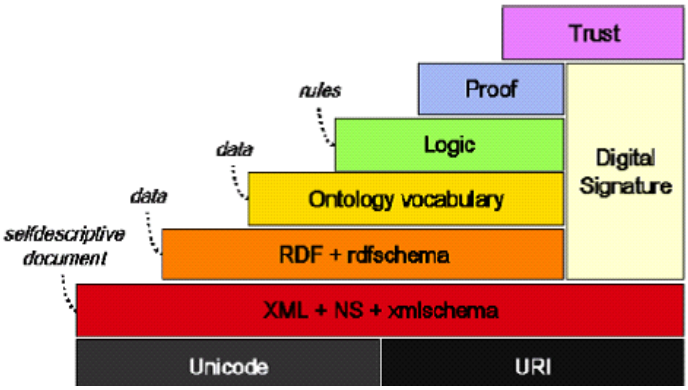

Figure II.1 Structure du Web sémantique

- ✓ **Les couches les plus basses :** assurent l'interopérabilité syntaxique : la notion d'URI fournit un adressage standard universel permettant d'identifier les ressources tandis qu'Unicode est un encodage textuel universel pour échanger des symboles.
- ✓ **XML :** fournit une syntaxe pour décrire la structure du document, créer et manipuler des instances des documents. Il utilise l'espace de nommage (Namespaces) afin d'identifier les noms des balises (tags) utilisées dans les documents XML.
- ✓ **Le schéma XML :** permet de définir les vocabulaires pour des documents XML valides. Cependant, XML n'impose aucune contrainte sémantique à la signification de ces documents, l'interopérabilité syntaxique n'est pas suffisante pour qu'un logiciel puisse comprendre le contenu des données et les manipuler d'une manière significative.

- ✓ **Les couches RDF M&S (RDF Model and Syntax) et RDF Schéma :** sont considérées comme les premières fondations de l'interopérabilité sémantique. Elles permettent de décrire les taxonomies des concepts et des propriétés (avec leurs signatures). RDF fournit un moyen d'insérer de la sémantique dans un document, l'information est conservée principalement sous forme de déclarations RDF. Le schéma RDFS décrit les hiérarchies des concepts et des relations entre les concepts, les propriétés et les restrictions domaine/co-domaine pour les propriétés.
- ✓ **La couche suivante Ontologie :** décrit des sources d'information hétérogènes, distribuées et semi-structurées en définissant le consensus du domaine commun et partagé par plusieurs personnes et communautés. Les ontologies aident la machine et l'humain à communiquer avec concision en utilisant l'échange de sémantique plutôt que de syntaxe seulement.
- ✓ **Les règles :** sont aussi un élément clé de la vision du Web sémantique, la couche règles offre la possibilité et les moyens de l'intégration, de la dérivation, et de la transformation de données provenant de sources multiples, etc.
- ✓ **La couche Logique** : se trouve au-dessus de la couche Ontologie. Certains considèrent ces deux couches comme étant au même niveau, comme des ontologies basées sur la logique et permettant des axiomes logiques. En appliquant la déduction logique, on peut inférer de nouvelles connaissances à partir d'une information explicitement représentée.
- ✓ **Les couches Preuve (Proof) et Confiance (Trust) :** sont les couches restantes qui fournissent la capacité de vérification des déclarations effectuées dans le Web Sémantique. On s'oriente vers un environnement du Web sémantique fiable et sécurisé dans lequel nous pouvons effectuer des tâches complexes en sûreté.
- ✓ D'autre part, la provenance des connaissances, des données, des ontologies ou des déductions est authentifiée et assurée par des **signatures numériques**, dans le cas où la sécurité est importante ou le secret nécessaire, le chiffrement est utilisé. **[13]**

XIII. Web Services sémantiques :

Les Web services sémantiques sont des Web services décrits de telle sorte qu'un agent logiciel puisse interpréter les fonctionnalités offertes par le Web service. Un agent logiciel doit être capable de lire la description d'un Web service pour

déterminer si le Web service fournit les fonctionnalités désirées, et s'il est lui-même capable d'utiliser ce service. Pour permettre cela, la description du Web service doit être supplémentée en information sémantique interprétable par la machine. Les paramètres du Web service doivent être décrits de façon qu'un agent logiciel puisse avoir connaissance de leur signification. Cela est fait par la définition de vocabulaires organisés en ontologies.

Dans le domaine des Web services, les ontologies sont utilisées pour annoter les descriptions des fonctionnalités des Web services de sémantique. Ainsi pour qu'un agent logiciel puisse avoir connaissance de la sémantique d'une description d'un Web service, il lui suffit juste d'accéder à une ontologie du domaine.

La combinaison des technologies du Web sémantique à celles des Web services permettra de :

- ✓ Sélectionner des Web services ;
- ✓ Automatiser la découverte de Web services, c'est-à-dire la localisation automatique des Web services qui fournissent une fonctionnalité particulière et qui répondent aux propriétés demandées par l'utilisateur. Pour pouvoir effectuer une découverte automatique, le procédé de découverte devrait être basé sur la similitude sémantique entre la description déclarative, faite par l'utilisateur, du service demandé et celle du service offert ;
- ✓ Automatiser la composition ;
- ✓ Automatiser l'invocation d'un Web service, cela implique l'automatisation de l'exécution du Web service par le programme d'utilisateur ou par un agent logiciel. **[14]**

XIII.1. Définition du web service sémantique :

Un Web service sémantique est le résultat de l'intégration de concepts sémantiques à la description du Web service. Le Web service sémantique est la convergence de deux technologies récentes du Web : les Web services et le Web sémantique. **[15]**

XIII.2. Description sémantique des Web services :

Deux méthodes possibles pour la description sémantique de Web services :

- ✓ La première approche consiste à annoter les langages existants (WSDL) avec l'information sémantique. Le principal avantage de ce genre de solutions est la facilité pour les fournisseurs de services d'adapter leurs descriptions existantes aux annotations proposées. Exemple : SAWSDL;
- ✓ La deuxième approche réécrit entièrement la description syntaxique du Web services en utilisant les technologies du Web sémantique. Exemple : OWL-S. **[15]**

XIII.3. RDF (Resource Description Framework):

Resource Description Framework (RDF) est un modèle de graphe destiné à décrire de façon formelle les ressources Web et leurs métadonnées, de façon à permettre le traitement automatique de telles descriptions.

Développé par le W3C, RDF est le langage de base du Web sémantique. L'une des syntaxes de ce langage est RDF/XML. D'autres syntaxes de RDF sont apparues ensuite, cherchant à rendre la lecture plus compréhensible ; c'est le cas par exemple de Notation3 (ou N3).

En annotant des documents non structurés et en servant d'interface pour des applications et des documents structurés (par exemple bases de données, GED, etc.) RDF permet une certaine interopérabilité entre des applications échangeant de l'information non formalisée et non structurée sur le Web.

Un document structuré en RDF est un ensemble de triplets (*sujet*, *prédicat*, *objet*) :

- ✓ **Le sujet** représente la ressource à décrire ;
- ✓ **Le prédicat** représente un type de propriété applicable à cette ressource ;
- ✓ **L'objet** représente une donnée ou une autre ressource : c'est la valeur de la propriété. **[16]**

XIII.4. OWL-S: Ontology Web Language – Service:

OWL-S est un sous ensemble du langage OWL dédié à la description sémantique de Web services. Il permet de décrire d'une façon non ambigüe les Web services de telle sorte qu'un agent logiciel puisse exploiter automatiquement ces informations.

OWL-S permet la découverte automatique et la composition de Web services. Une description OWL-S se compose de trois éléments***: le service profile***, ***le service model***, et ***le service grounding***, qui décrivent respectivement que fait le service, comment le service fonctionne et comment accéder au service. **[17]**

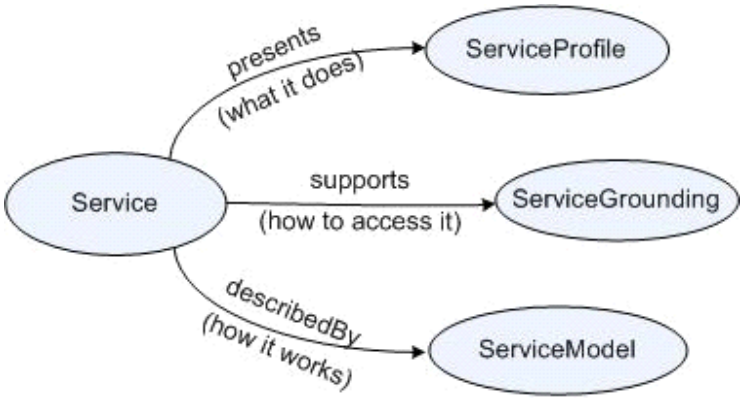

Figure II.2. Niveau supérieur de l'ontologie OWL-S

XIII.4.1. Service profile:

OWL-S fournit cette classe pour décrire un Web service. Cette classe Service Profile spécifie trois informations :

- ✓ **Nom du service, contacts et description textuelle du service** : le nom du service est utilisé comme identificateur du service, tandis que les informations contacts et la description textuelle sont destinées aux utilisateurs humains.
- ✓ **Description fonctionnelle du service** *:* Elle spécifie ce que l'on peut attendre du service en termes d'entrées attendues et de résultats produits en sortie. Les transformations Web Services Sémantiques d'informations sont représentées par des Inputs et des Outputs. Le changement d´état du monde réel causé par l'exécution du service est représenté par pré-conditions et effets qui sont les pré-conditions et les post conditions de son exécution. Les Inputs et les Outputs font références à des classes d'OWL décrivant les types des instances à envoyer au service et aux réponses respectives attendues.

- ✓ **Propriétés additionnelles** *:* Plusieurs propriétés sont utilisées pour définir un Web service. La première est la catégorie du Web service. La seconde est la qualité de Web service QoS. Enfin, le Web service peut fournir une liste de paramètres de façon libre.

XIII.4.2. Service model:

Les Web services peuvent être décrits en OWL-S en tant que processus grâce à la classe Process. La classe ainsi définie est une sous-classe de Service model. Pour

décrire un processus, on spécifie ses entrées, sorties et ses états (pré-conditions et effets). Les transitions d'un état à un autre sont décrites par les pré-conditions et les effets de chaque processus. Il existe trois types de processus :

- ✓ **Les processus atomiques (Atomic Process)**: exécutable en une seule étape, un processus atomique ne peut pas être décomposé de façon plus profonde. Son exécution correspond `a une unique avancée dans l'exécution du service, il est directement invoqué par l'utilisateur du service.

- ✓ **Les processus composites (Composite Process)*:*** un processus composite est constitué par l'assemblage d'autres processus (eux-mêmes peuvent être composites ou non composite). Les processus composites associent des processus à l'aide de structures de contrôle permettant de décrire leur logique d'exécution.

- ✓ **Les processus simples (Simple Process):** Les processus simples ne sont pas invocables mais comme les processus atomiques, leurs exécutions s'effectuent en une seule étape. Les processus simples sont employés comme éléments d'abstraction. Un processus simple peut être employé pour fournir une vue d'un certain processus atomique, ou une représentation simplifiée d'un certain processus composite.

XIII.4.3. Service grounding:

La classe OWL-S Service Grounding définit les détails techniques permettant d'accéder au Web service.

Les deux premières classes Service Profile et Service Model d'une description OWL-S Web Services Sémantiques s'attachent à abstraire la représentation d'un Web service. Service Grounding est la forme concrète d'une représentation abstraite, elle fournit les détails concrets d'accès au Web service, tels les protocoles, les URIs, les messages envoyés, etc.

XIII.5. Semantic Annotations for WSDL (SAWSDL):

SAWSDL est un langage sémantique de description de service Web. Il est évolutif et compatible avec les standards des services Web existants, et plus spécifiquement avec WSDL.

SAWSDL augmente l'expressivité du langage WSDL avec la sémantique en utilisant des concepts analogues à ceux utilises dans OWL-S. **[18]**

SAWSDL fournit un mécanisme permettant d'annoter sémantiquement les types de données, les opérations, les entrées/sorties de WSDL. En plus de la découverte et l'invocation automatiques des services Web, SAWSDL vise la réalisation des objectifs suivants :

- ✓ **Un langage au dessus des standards existants** *:* Les entreprises ont déjà investi dans des projets d'intégrations basés sur les services Web et utilisant les standards existant. Par conséquent, les concepteurs de SAWSDL considèrent que n'importe quelle approche pour la description sémantique des services Web devrait être compatible avec l'existant. A cet effet SAWSDL, une extension sémantique de WSDL a été conçue.

- ✓ **Concevoir un langage incrémental:** il est relativement facile de modifier les outils existants autour de WSLD afin qu'ils s'adaptent à SAWSDL. Ce qui fait de SAWSDL une approche incrémentale.

- ✓ **L'annotation doit être indépendante du langage de représentation de la sémantique :** Il y a un certain nombre de langages potentiels pour représenter la sémantique comme Web Service Modeling Language (WSML) et Unified Modeling Language (UML). Chaque langage offre différents degrés d'expressivité. La position des concepteurs de SAWSDL est qu'il n'est pas nécessaire d'attacher les standards des services Web à un langage sémantique particulier. Cette approche est la vision décrite par Sivashanmugam et al. et donne plus de flexibilité aux développeurs.

- ✓ **L'annotation sémantique :** est faite dans SAWSDL des documents WSDL est possible grâce à l'extensibilité de WSDL 2.0. En effet, conceptuellement WSDL 2.0 est doté des constructions suivantes : interface, opération, message, binding, service et endpoint. Les trois premiers à savoir interface, opération et message concernent la définition abstraite du service tandis que les trois autres sont relatifs à l'implémentation du service. SAWSDL fournie des mécanismes pour référencer des concepts de modèles définis à l'extérieure du document WSDL. Cela se fait grâce à l'attribut ≪SAWSDL≫.

XIII.6. Web Service Modeling Ontology (WSMO):

WSMO ou service Web Ontology Modeling est un modèle conceptuel pour les aspects pertinents liés aux services du Web sémantique . Il fournit une ontologie qui prend en charge le déploiement et l'interopérabilité des services Web sémantique . **[19]**

Le WSMO comporte quatre volets principaux:

- ✓ **Objectifs** - indiquent ce que l'utilisateur attend du service.
- ✓ **Ontologies** - définissent la terminologie utilisée par les autres éléments.
- ✓ **Médiateurs** - lient les différents éléments afin de permettre l'interopérabilité entre les composants hétérogènes.
- ✓ **Web Services** - définissent les fonctionnalités offertes.

XIV. Les avantages de l'utilisation du Web sémantique pour la description des services Web :

Les avantages de l'utilisation du Web sémantique pour la description des services Web sont nombreux. En plus, de rendre l'interface du service Web accessible automatiquement par des machines, ils permettent également, la description de propriétés non fonctionnelles telles que la qualité de services, les contraintes de sécurité, et l'intégration effective des services Web dans des applications industrielles, d'une manière uniforme compréhensible par tous. **[20]**

IX. Conclusion:

Les technologies du web sémantique ont été utilisées pour enrichir les services Web sémantique, ce qui permet l'automatisation des divers aspects relatifs aux services Web. Le chapitre qui suit va être consacré à la découverte des services web et aux approches qui ont été déjà proposés pour améliorer ce processus.

Chapitre II

I. Introduction :

Les services Web constituent un moyen pour les entreprises d'offrir leurs services à travers le Web. Avec le nombre grandissant de fournisseurs de services, il est nécessaire de disposer de mécanismes pour la découverte de services Web de manière efficace et flexible.

II. Publication des services web :

UDDI (Universal Description, Discovery and Intégration) est une spécification d'annuaires de services web, cette norme W3C propose un ensemble de structures à publier par les fournisseurs de services afin de publier leurs services. Ces structures sont formalisées en XML, elles offrent 03 types d'informations **[21]** :

- ✓ **Les pages blanches :** qui décrivent les informations de contacts sur les entreprises.
- ✓ **Les pages jaunes :** qui décrivent des informations de classification de services.
- ✓ **Les pages vertes :** qui donnent des informations techniques des services.

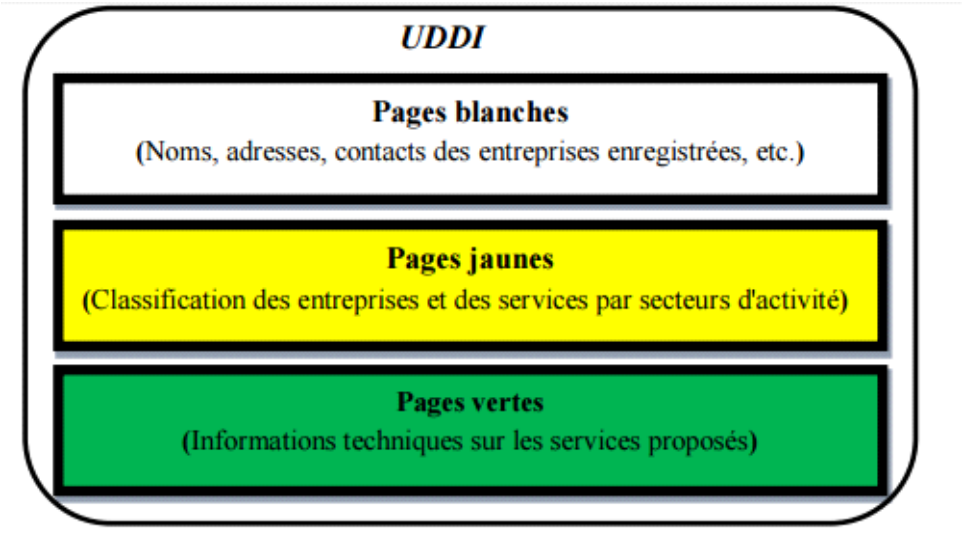

Figure III.1. Le contenu de l'annuaire UDDI

La norme UDDI offre aussi une API aux applications clientes, pour rechercher des services et/ou ses fournisseurs, ajouter ou modifier des services ou des entreprises. Nous distinguons deux types d'annuaires UDDI (publiques et privés) :

- ✓ **L'annuaire UDDI publique** est implémenté sous forme d'un réseau de nœuds UDDI, ces nœuds sont synchronisés, chacun d'eux est possédé par une entreprise donnée (telles que SAP, IBM et Microsoft). La publication d'un service chez une entreprise Propage automatiquement ses informations (pages blanches, jaunes et vertes) aux différents nœuds UDDI. L'accès à l'ensemble des informations des registres peut se faire à n'importe quel nœud UDDI. Ce type d'annuaire est gratuit.

- ✓ **L'annuaire UDDI privé** permet à une entreprise de choisir les partenaires pour lesquels elle autorise la publication et l'invocation de ses services web.

Les structures de données d'un registre UDDI sont illustrées dans la figure suivante :

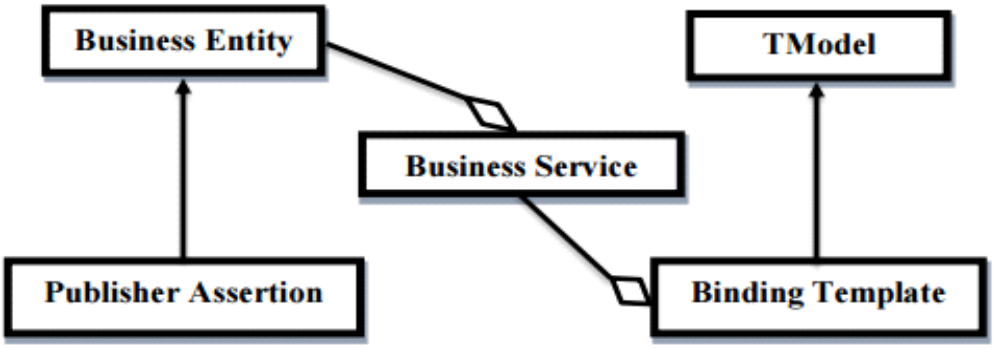

Figure III.2. Les structures de données d'UDDI

- ✓ **La partie « pages blanches »** : est constituée de deux éléments <business Entity > et <Publisher Assertion>.

  - <BusinessEntity> : donne des informations de contact sur l'organisation fournissant les services (les noms des dirigeants, les emails, les numéros de téléphones, le site web, etc.).

- <PublisherAssertion> : spécifie les liens d'affiliation entre deux entreprises (mère et fille). Lorsque deux éléments <BusinessEntity> référencent la même <PublisherAssertion, nous parlons de relation d'affiliation entre ces <BusinessEntity> .

✓ **La partie « pages jaunes »** : est constituée des éléments <BusinessService>.

- <BusinessService> décrit un ensemble de services fournis par une organisation (le nom du service, la description textuelle, les informations de classification). Un <BusinessService> est un sous élément de <BusinessEntity>.

✓ **La partie «pages vertes»**: est constituée des éléments <BindingTemplate>et <tModel>.

- <BindingTemplate> spécifie des informations techniques, telles que l'URI du service. Un <BindingTemplate> est un sous élément de <BusinessService>.

- <tModel> définit des structures d'informations techniques telles que l'interface WSDL, les taxonomies industrielles, les ontologies, etc. Un élément <tModel> peut être réutilisé par plusieurs éléments <BindingTemplate>.

Le protocole d'utilisation de l'UDDI offre trois fonctions de base :

✓ **Publish :** permet de publier un nouveau service.
✓ **find :** permet d'interroger l'annuaire UDDI.
✓ **bind:** permet d'établir une connexion entre l'application cliente et le service.

III. Annotation sémantique de services web :

Il y a eu une prolifération récente de services Web comme la technologie pour l'exécution de processus commercial et l'intégration d'application. Bien que des services Web soient basés sur des standards largement acceptées, le manque d'une description formelle de la signification de leur fonctionnalité et de leurs données échangées a été un barrage significatif dans la réalisation de promesses d'intégration.

Comme le nombre de services Web augmente, il est important d'automatiser des outils pour la découverte des services Web. La mesure de description disponible dans la norme WSDL actuelle quitte la pièce pour les interprétations ambiguës des fonctionnalités et des données d'un service Web. **[22]**

L'ambiguïté dans l'interprétation gêne l'automatisation de tâches comme la découverte de service, la composition, l'invocation etc. Une des façons que les développeurs ont utilisées pour aborder ces questions est en développant une langue de balisage sémantique pour des Services Web.

III.1. Annoter un service web :

Sémantiquement l'annotation d'un service Web implique l'explication de la sémantique exacte des données de service Web et les éléments de fonctionnalité qui sont cruciaux pour l'utilisation du service Web. Ceci est fait en annotant les éléments de service Web avec des concepts dans des modèles de domaine ou des ontologies.

Puisque les ontologies représentent une vue convenue du domaine modelé, donc n'importe quelle ambiguïté dans l'interprétation de fonctionnalité ou les données d'un service Web est éliminée. Le but d'annoter des services Web est de permettre la découverte automatisée de service sans équivoque. Par exemple, deux services Web signifiés pour des fonctionnalités complètement différentes peuvent utiliser les mêmes types de données et noms pour leurs opérations (inputs/outputs), rendant ainsi l'interprétation de leur fonctionnalité ambiguë. Pour comprendre quelles parties d'un service Web doivent être annotées, il est important de comprendre l'interaction de la sémantique dans le cycle de vie ou leur utilisation dans un service Web.

En découvrant un service Web, un demandeur décrit ses exigences en termes de la fonctionnalité c'est-à-dire des opérations d'un service Web et les données utilisées (inputs/outputs). Les spécifications facultatives incluent les conditions

préalables et les effets de l'opération. Les conditions préalables sont des exigences qui doivent être respectées avant qu'une opération de service Web ne soit invoquée et les effets sont les résultats d'invocation d'une opération.

Les annotations sémantiques sont donc associées aux inputs/outputs des conditions préalables et les effets d'un élément d'opération d'un service Web. Cependant Plusieurs mécanismes de découverte avancés considèrent les aspects non fonctionnels de services Web et les exigences grand public comme la métrique de qualité, la fiabilité, la sécurité etc.

L'avantage d'ajouter de la sémantique est omniprésente dans le cycle de vie entier d'un processus Web (voir la **Figure III.3**). Les développeurs peuvent utiliser des annotations sémantiques pour expliquer les capacités de leurs services Web **(1).** Une fois que ces services Web sont publiés dans l'UDDI **(2)**, un demandeur peut formuler ses exigences dans un modèle de service sémantique **(3)** pour découvrir ou composer des Services Web. Un service sémantique ou un modèle de processus est un service abstrait ou une description de processus, où le flux de contrôle est créé manuellement et la fonctionnalité exigée est décrite en utilisant des termes d'un modèle de domaine ou une ontologie.

Le raisonnement technique peut être utilisé pour comparer les exigences dans le modèle de service avec les capacités des services Web disponibles dans l'UDDI **(4)** pour la découverte de ces derniers. Pendant la composition, l'aspect fonctionnel des annotations peut être utilisé pour créer des compositions de service utiles.

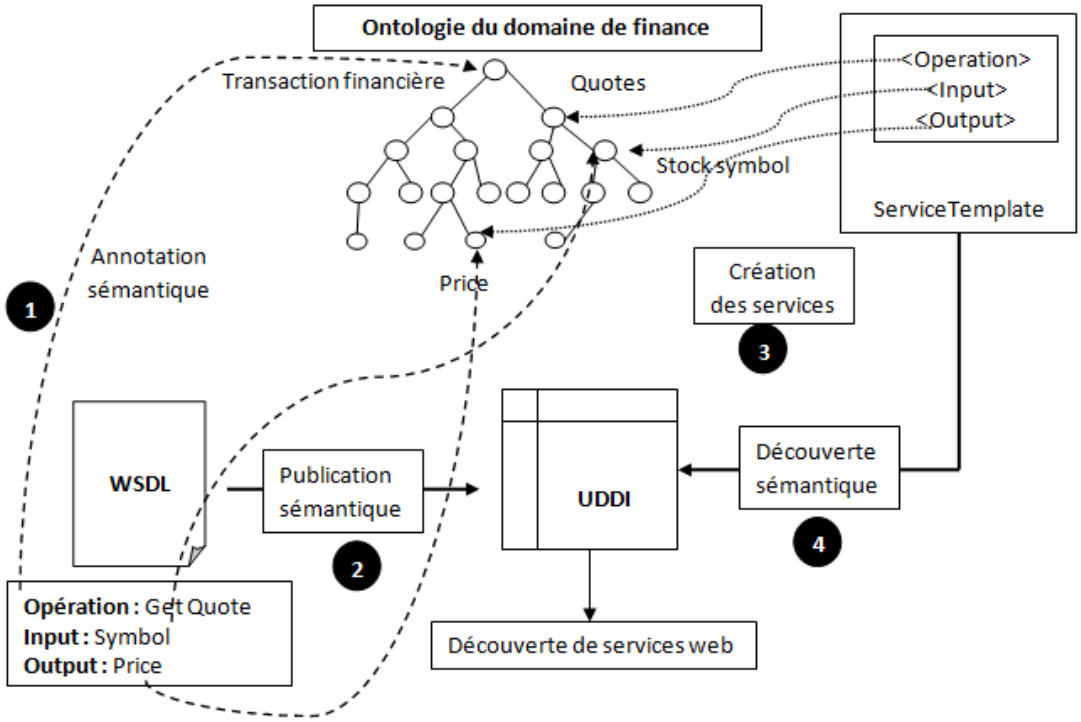

Figure III.3. Sémantique dans cycle de vie d'un service Web

III.2. Quatre Types de Sémantique dans les Services Web :

La table III.2 illustre les quatre types de sémantique : données, fonctionnelles, non-fonctionnelles Et sémantique d'exécution associée aux services Web et comment Ils touchent aux différentes étapes montrées dans la **Figure III.1.**

Types de sémantiques	**Description**	**Utilisation**
Sémantique de données	Définition formelle des données dans les messages inputs/outputs des services web.	Découverte de services et interopérabilité entre les services.
Sémantique fonctionnelle	Définition formelle des capacités d'un service Web.	Découverte et Composition de services Web.

Sémantique non-fonctionnelle	Définition formelle de contraintes quantitatives ou non-quantitatives comme QoS (Qualité de Web Service).	Découverte, Composition Et Interopérabilité de services Web.
Sémantique d'exécution	Définition formelle de l'exécution ou du flux de services dans processus ou d'opérations dans services.	Vérification de processus et traitement d'exception.

Tableau III.1. Types de Sémantique dans les Services Web

La vérification de processus implique la vérification de la justesse (le contrôle et le flux de données) d'une composition de processus. L'objectif de traitement d'exception est d'identifier des points de panne dans un processus Web et définir comment les surmonter.

Maintenant que nous comprenons pourquoi la sémantique est exigée dans des descriptions de service Web et quel genre de la sémantique est utile, nous pouvons continuer à explorer comment ces annotations sémantiques sont créées.

III.3. Création des annotations sémantiques :

L'idée fondamentale derrière l'association de sémantique avec des éléments de service Web est de trouver le concept sémantique le plus approprié dans une ontologie pour un élément WSDL. Ceci est fait en correspondant un WSDL et un schéma de modèle de domaine.

Pour faire simple nous supposons que les modèles de domaine ont été créés en utilisant OWL, bien qu'ils puissent bien être représentés dans RDF, UML, etc.

La correspondance entre un WSDL (essentiellement XML) et le schéma OWL présente le problème de matching entre deux modèles hétérogènes, chacun avec sa propre expressivité, capacités et des restrictions. Le problème de matching entre deux schémas revient au problème d'interopérabilité de données dans le contexte de schémas de base de données. Les mots matching et mapping étaient souvent utilisés de façon interchangeable dans la littérature.

Le schéma de mot matching se réfère au processus de recherche des correspondances sémantiques entre les éléments de deux schémas et mapping traite la représentation physique établis par la correspondance de schéma et les règles pour transformer les éléments d'un schéma à celui de l'autre.

Par exemple dans la **Figure III.4.**qui montre un élément WSDL et un concept d'OWL, le résultat de schéma de metching doit identifier l'objet POAddress dans le WSDL qui est sémantiquement équivalent au concept Adresse dans l'ontologie.

Le mapping montré comme XQuery (XQuery LO : un Langage d'interrogation XML) et les scripts XSLT (XSL des Transformations (XSLT)) font la correspondance opérationnelle en spécifiant des règles pour transformer les éléments d'un schéma à celui de l'autre.

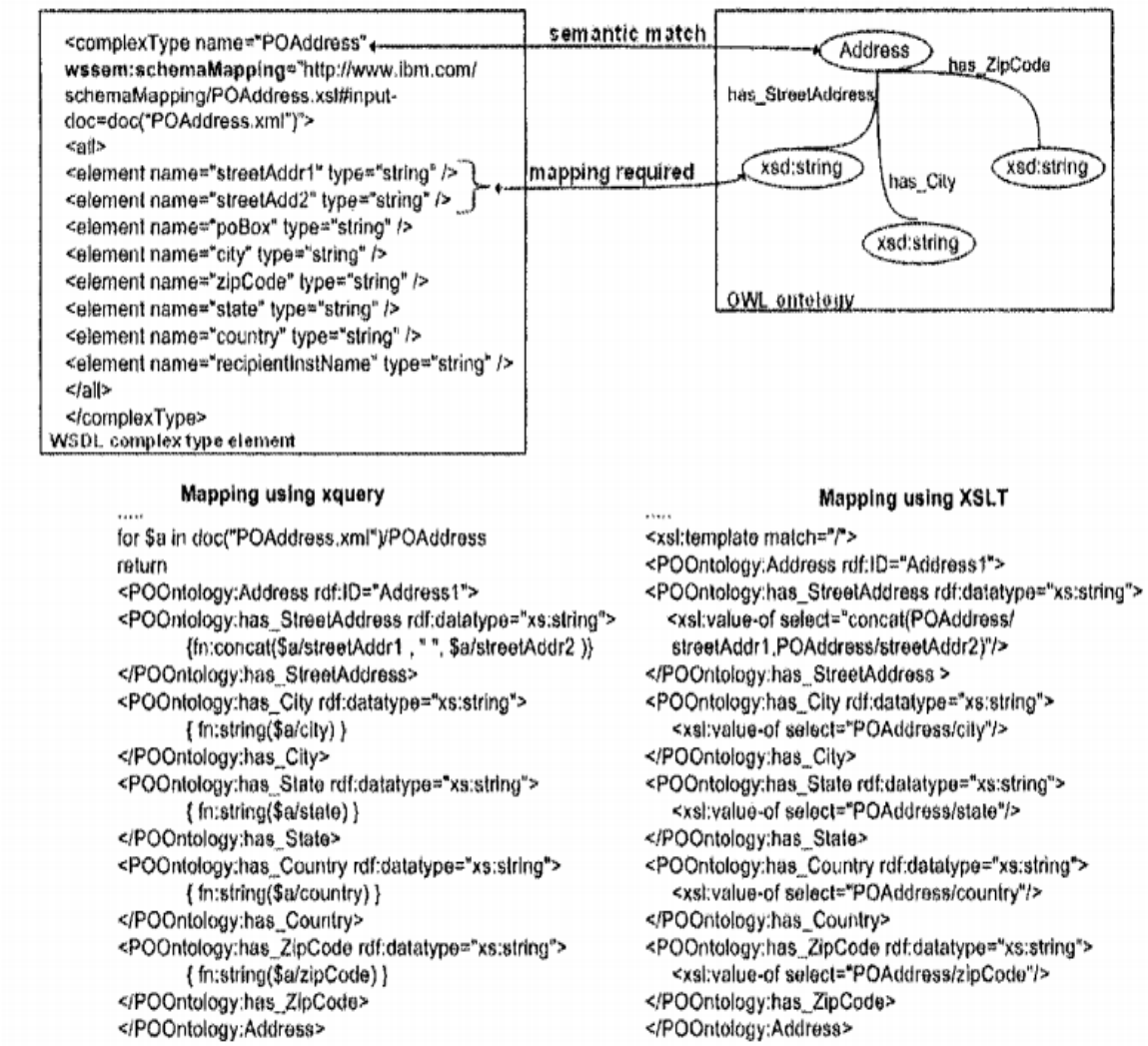

Figure III.4. Elément WSDL et un concept de OWL

III.3.1. Matching (correspondance) :

Il y a eu un travail significatif dans la communauté de base de données pendant les années 1980 et au début des années 1990 sur la reconnaissance du besoin de l'interopérabilité de données, le schéma de mapping/merging/transformations, l'hétérogénéité sémantique et l'utilisation d'ontologie et des logiques de description pour l'intégration schématique et sémantique, etc.

Cependant, une grande partie du travail passé dans l'intégration de base de données s'est concentré sur la correspondance aux modèles homogènes, par exemple, entre deux schémas de base de données. N'importe quelle différence dans la représentation de schéma a été traitée normalisant les schémas disparates avant le matching. Dans le cas de matching d'un WSDL (XML le schéma) et le schéma OWL, nous traitons vraiment deux modèles différents.

La transformation d'un modèle moins expressif (XML) à un modèle plus expressif (OWL) exigerait d'habitude que des humains fournissent la sémantique supplémentaire, tandis que la transformation dans l'autre direction peut être mieux.

Le travail actuel dans le domaine du modèle de management s'est concentré sur le développement d'une infrastructure générique qui résume des opérations sur des modèles (c'est-à-dire, des schémas) et des mapping entre des modèles comme les opérations de haut niveau qui sont génériques et indépendantes du modèle de données et l'application d'intérêt.

Dans le domaine de services Web, **[23]** aborde la différence dans l'expressivité entre le OWL et WSDL (XML) en normalisant tous les deux les représentations pour un format de graphique commun. Le résultat de MATCHING doit établir les correspondances sémantiques qui sont alors représentées comme des annotations.

L'utilisation possible de techniques d'apprentissage automatique pour créer des métadonnées pour des services Web a été explorée dans ASSAM **[24].**

Le composant d'annotateur d'ASSAM **[25]** place le problème de classification des opérations et les données d'un Service Web comme des problèmes de classification de texte. L'outil lit depuis le service web les annotations sémantiques existantes et fournies les données d'apprentissage et les étiquettes sémantique pour des services web invisibles. Une tentative semblable à l'utilisation de techniques d'apprentissage automatique est présentée dans **[26].**

Un système semi-automatisé pour créer des annotations sur des éléments de Service Web devrait donc pouvoir correspondre à un schéma WSDL et un ou plusieurs schémas de modèle de domaine et rendre les correspondances sémantiques avec le degré de certitude dans les matchs.

En cas de l'ambiguïté, la participation d'utilisateur pourrait aider à affiner les matchs produits par le système. Bien que le besoin du schéma correspondant soit tout à fait évident (pour produire des annotations sémantiques), le besoin de fournir des mapping mérite plus d'attention.

III.3.2. Mapping :

Comme nous avons vu, des annotations sémantiques sur des éléments de service Web facilitent la découverte de service sans équivoque et la composition. Dans le contexte de composition de service, l'ordre de services assure une compatibilité sémantique entre leurs inputs/outputs, mais n'assure pas nécessairement d'interopérabilité.

IV. Découverte des services web :

La découverte de services Web consiste à localiser les descriptions des services Web décrivant certains critères fonctionnels. **[22]**

Initialement, les descriptions des services web n'étaient faites qu'au niveau syntaxique, la découverte de services Web était basée sur des techniques issues du domaine de la recherche d'information. Cependant, avec le développement des technologies du web sémantiques et afin d'automatiser la découverte de services Web, de nouvelles approches basées sur la sémantique des descriptions des services ont été proposées.

IV.1. Critères de découverte:

IV.1.1. Le critère d'automatisation:

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

IV.1.2. Le critère d'architecture (Centralisation/distribution):

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

IV.1.3. Le critère de Matchmaking:

Ce critère s'intéresse à l'algorithme de description de la requête de client avec la description du service, il y a deux approches qu'on peut les distinguées, les approches fonctionnelles et les approches non fonctionnelles.

IV.2. Les approches de découverte :

Les approches de découverte dépendent du niveau de représentation, sémantique ou syntaxique, des descriptions des services Web. Cependant, les approches varient selon l'architecture centralisée ou distribuée adoptée.

IV.2.1. Approches syntaxiques:

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

IV.2.1.1. Approche UDDI :

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

IV.2.1.2. Approche basée sur les qualités de service :

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

- ✓ Le fournisseur de service : offre le service en l'enregistrant dans l'annuaire UDDI.

- ✓ Le consommateur de service : c'est lui qui découvre et invoque les services.
- ✓ L'annuaire UDDI : c'est l'annuaire doté d'informations sur la description fonctionnelle du web service et des informations sur les Qds associés à ce service.
- ✓ Et le certificateur: son rôle et de vérifier les revendications de qualité de service d'un web service avant son enregistrement.

IV.2.2. Approche sémantique :

Ces approches fondées sur des langages particuliers d'ontologies du web. Ceci leur permet de décrire les services web de façons non ambiguë et interprétable par des programmes. Cette famille comprend OWLS et WSMO qui représente les modèles sémantiques les plus utilisées dans la littérature. Cette catégorie assure une meilleure interprétation et par la suite garantir une découverte efficace en améliorant la qualité des résultats obtenus, donc ces approches sont plus complexes et plus fiable que les technique syntaxique, cette approche peuvent être regroupées en trois classes : l'approche algébrique, l'approche déductive et l'approche hybride.

L'approche algébrique exploite généralement la théorie des graphes et procède par calcul de distance, tandis que l'approche de découverte déductive est fondée essentiellement sur la logique. L'approche qualifiée d'hybride, quant à elle, est celle qui combine les mécanismes de ces deux classes.

IV.2.2.1. Approche algébrique :

L'approche de découverte algébrique trouve ses origines dans l'algèbre et les mécanismes de recherche d'information. Elle est fondée sur le calcul du degré de similarité textuelle à partir de graphes structurés construits à cet effet, ou encore sur le calcul de distance (du chemin) entre les concepts appariés.

Cette approche que nous qualifions d'algébrique utilise des mécanismes de matching structurel, numérique et syntaxique à travers un appariement de graphes structurés et en calculant des distances numériques pour vérifier la similitude syntaxique.

Pour exploiter la sémantique, ces mécanismes d'appariement utilisent les fréquences des termes et les sous-graphes. iMatcher1 **[27]**, Agent Approach for Service Discovery and Utilization (AASDU) **[28]** et DSD-Matchmaker **[29]** sont des plateformes et travaux qui adoptent cette catégorie d'approches.

IV.2.2.2. Approche déductive :

L'approche déductive est fondée sur la logique. Les travaux optant pour cette catégorie d'approches utilisent des descriptions de services et des requêtes spécifiées en langages issus de formalismes logiques, telles que la logique de description et la logique de premier ordre. Ils utilisent également des règles logiques pour la découverte de services Web et exploitent les ontologies pour couvrir leur aspect sémantique.

Pour calculer le degré d'appariement, ils recourent à différentes façons et ciblent plusieurs éléments de description de services tout en prenant en considération leur sémantique. Ils optent essentiellement pour trois types d'appariement : IO-matching (Inputs and Outputs matching) **[30][31][32]** PE-matching (pré conditions and effects matching) **[27]** et IOPE-matching (Inputs, Outputs, pré conditions and effects matching) **[33][34][35][36]**.

Dans l'IO-matching, les éléments objet d'appariement se limitent aux inputs et outputs. PE-matching prend les préconditions et effets comme éléments d'appariement. Quant au cas du IOPE-matching, les concepts sémantiques décrivant les inputs, outputs, pré conditions et effets font à la fois l'objet de l'appariement des services et des requêtes.

IV.2.3. Approche Hybride :

L'approche hybride utilise des mécanismes déductifs intégrant les méthodes de calcul de distances. L'idée est de remédier aux limites de chacun de ces deux mécanismes en les combinant. Plusieurs travaux **[37][38][39][40][41][42]** optent pour cette approche.

V. Sélection des services web :

La sélection des Web services consiste à choisir parmi un ensemble de Web services découverts ceux qui répondent aux préférences de l'utilisateur sur la base des besoins fonctionnels et non fonctionnels. Les besoins non fonctionnels sont généralement exprimés à l'aide des critères de QoS. Plusieurs approches et méthodes ont été présentées dans la littérature pour la sélection de services selon les propriétés non fonctionnelles. **[43]**

V.1. Propriétés fonctionnelles et non fonctionnelles dans les Web services :

Les propriétés fonctionnelles et non fonctionnelles de services définissent un Web service. En effet, les services se distinguent par les fonctionnalités qu'ils peuvent fournir. Ainsi deux services similaires, c'est à dire qui offrent sémantiquement tous les deux les mêmes fonctionnalités (exemple un service Température et un service Heat qui offrent des fonctions de mesure de température), peuvent se différencier par leurs propriétés non fonctionnelles. Par exemple le service Température a un temps d'exécution inférieur au service Heat alors que ce dernier dispose de propriétés de sécurité telle que la confidentialité.

V.1.1. Les propriétés fonctionnelles :

Les propriétés fonctionnelles d'un service désignent les opérations qu'il peut fournir. Les propriétés fonctionnelles sont décrites dans la description de service en termes d'entrées, sorties, pré-conditions et effets du Web service simple et reflètent le fonctionnement du service.

V.1.2. Les propriétés non fonctionnelles:

Les propriétés non fonctionnelles de service appelées aussi : qualités de service définissent les capacités de service à fonctionner dans de bonnes conditions en termes de disponibilité, performance, coût d'invocation, fiabilité, etc.

V.2. Sélection basée sur les besoins fonctionnels :

Dans ce cas, la sélection de service se base principalement sur les propriétés fonctionnelles d'un service. Un fournisseur de service publie une description de son service avec laquelle un utilisateur peut chercher le service qui répond aux fonctionnalités requises. Les moyens de sélection de services actuels reposent sur l'adéquation syntaxique exacte entre l'interface de service offerte par les fournisseurs et l'interface de service requise par l'utilisateur.

V.2.1. Insuffisance des approches basées sur les besoins fonctionnels :

Cette approche basée sur les besoins fonctionnels présente un manque. En effet, dans l'environnement des Web services, plusieurs services peuvent offrir les

mêmes fonctionnalités. Ce qui force l'utilisateur à choisir manuellement le service ou d'effectuer un choix aléatoire qui ne garantit pas une qualité de service. Par conséquent, l'aspect fonctionnel est certes important, mais non suffisant.

V.3. Sélection de services basée sur les besoins non fonctionnels :

Pour pouvoir réaliser la sélection parmi un ensemble de services dont les fonctionnalités sont équivalentes : Seules les propriétés non fonctionnelles du service font l'objet de raffinement pour sélectionner le meilleur service. Les besoins non fonctionnels des Web services sont généralement exprimés à l'aide des critères de QoS.

V.3.1. Présentation du problème de sélection selon les propriétés de QoS :

Un Web service composite est constitué d'un ensemble de tâches à exécuter, ses tâches composantes peuvent être exécutées à l'aide de Web services. En particulier, pour une même tâche, on peut découvrir plusieurs Web services aptes à l'exécuter, on serait alors face à un problème de sélection de Web services afin de choisir la combinaison de services qui offre la meilleure qualité de service. Cette sélection devra aussi prendre en considération le respect des contraintes de l'utilisateur.

V.4. Stratégie de Sélection :

On distingue principalement deux stratégies de sélection différentes:

- ✓ Selection locale.
- ✓ Selection global.

V.4.1. Sélection Locale :

Elle a pour objectif de choisir le meilleur Web service pour chaque tâche individuelle à part entière en considérant des contraintes de QoS relatives à chaque tâche plutôt qu'en considérant des contraintes de QoS globales exprimées pour l'ensembles des tâches. Il s'agit de sélectionner pour chaque tâche un Web service apte à l'exécuter en tenant en compte les contraintes de l'utilisateur.

V.4.1.1. Avantages :

- Simple à mettre en œuvre.
- Trouve le meilleur Web service adéquat pour chaque tâche.
- Parvient à trouver une bonne solution dans des temps de calcul raisonnables.

V.4.1.2. Inconvénients :

- Il arrive que la composition de Web services peut être cachée à l'utilisateur, c'est-à-dire, il n'arrive pas à distinguer entre les services simples et les services composites.
- L'utilisateur ne peut pas définir des contraintes sur la totalité de la composition.

V.4.2. Sélection globale:

Contrairement à la méthode précédente, on choisit la combinaison de Web services qui garantit la meilleure qualité globale en tenant compte des contraintes de QoS et des préférences globales assignées pour l'ensemble des tâches.

V.4.2.1. Avantages :

- L'utilisation des contraintes globales sur l'ensemble de la composition est plus évidente pour l'utilisateur car il ne peut pas distinguer entre les différents services composants.

V.4.2.2. Inconvénients :

- Le nombre de combinaisons possibles augmente exponentiellement avec le nombre de tâches et de candidats.

VI. Conclusion:

La découverte de services Web présente un axe de recherche émergent. Plusieurs approches ont été présentées dans la littérature. L'objectif commun de ces approches était de découvrir des descriptions de services Web décrivant certains critères fonctionnels.

Au début, les descriptions des services Web étaient principalement syntaxiques, cependant, avec la fusion des deux technologies services Web et Web sémantique

une nouvelle génération de services Web est née connue sous le nom de services Web sémantiques. Avec le développement des services Web sémantiques, de nouvelles approches de découverte ont été proposées. Ces approches se basent essentiellement sur la description des services sans tenir compte du retour d'usage.

Pour résoudre ce problème, nous proposons une approche qui exploite l'annotation par des tags (mots clés) de ces services pour indiquer l'avis des utilisateurs sur les fonctionnalités ou la qualité des services utilisés, ainsi les résultats de la recherche seront classés selon le poids des tags qui leurs sont associés.

Dans le chapitre qui suit, nous allons détailler la conception de notre approche.

Chapitre III

I. Problématique :

Avec l'évolution du web ces dernières années le nombre de services web ne cesse d'accroitre ce qui provoque une immersion pour les utilisateurs qui se retrouvent confrontés à plusieurs questions concernant la façon d'exprimer leurs exigences d'une manière simple, libre et souple, mais efficace. La plupart du temps, les utilisateurs découvrent les services Web via une interface de recherche en utilisant des requêtes simples avec des fonctionnalités avancées ou des requêtes composées. Les deux styles de recherche ne facilitent pas la découverte de service Web dans certains aspects: d'une part, le mécanisme de découverte simple à base de mot-clé offre souvent un résultat de correspondance faible dans les environnements actuels parce qu'ils traitent des services Web mal décrits (description textuelle insuffisante des services Web dans les fichiers WSDL) et d'autre part, les formalismes composés utilisés pour construire les requêtes de découverte sont souvent liés à la description logique, qui exigent des efforts des utilisateurs lors de la construction de leurs requêtes, mais aussi des fournisseurs du service pour enrichir leurs services publiés avec la description logique ou de l'information ontologique. Les algorithmes de recherche, dans ce cas, peuvent avoir recours à l'ontologie d'utilisation et exiger la définition d'ontologie normalisée.

La découverte d'une ressource Web a été très améliorée en utilisant des tags dans le contexte actuel du Web 2.0. Le succès d'une telle approche peut être interprété par le nombre croissant d'utilisateurs et les taux de différentes applications sur le Web. Au sein de ces systèmes de marquage de collaboration, les utilisateurs peuvent marquer ou annoter une ressource (par exemple, image, URL, référence bibliographique, etc.) avec des mots clés afin de le classer, partager avec d'autres utilisateurs et de faciliter la recherche future de celle-ci. Ce genre d'environnement incite les utilisateurs à participer au processus d'enrichissement de toute ressource Web avec des descriptions textuelles (mots-clés ou texte libre). Une ressource Web peut alors être découverte par un nuage de tags qui lui est associé.

Si l'on considère un service Web par le biais de ses caractéristiques physiques, il est représenté par une URL pointant à sa description (en WSDL). Ce dernier ne tient pas de description textuelle suffisante. Associer une description textuelle du nombre actuel de services Web publiés semble fortement être une tâche impossible pour les fournisseurs de services. Toutefois, le marquage de ces services sera juste une tâche participative des

utilisateurs de services; il ne nécessite aucun effort de la part du prestataire de service et ne porte pas atteinte à l'intégrité des descriptions de service. Nous pouvons également assurer par ce mécanisme que les habitudes des utilisateurs sur le Web ne changent pas et la découverte d'un service Web se passera de la même manière que dans d'autres processus de recherche de ressources Web (par exemple, image, URL, etc.).

Convaincus par divers avantages des systèmes de marquage de collaboration dans l'environnement Web 2.0 et les difficultés rencontrées par les utilisateurs lors de la recherche d'un service Web, nous allons proposer dans ce travail de recherche un environnement de découverte de service Web basé sur les systèmes de marquage de collaboration. Nous croyons qu'un tel système aidera ceux qui ont besoin de chercher un service Web, dans la construction de la requête, mais aussi dans la conservation de leurs comportements communs à ceux qui sont utilisés pour toute recherche d'une ressource Web taguée. Dans cette proposition nous allons discuter de la façon dont un service Web peut être marqué et trouvé plus tard en utilisant les tags.

II. Les acteurs intervenants du système :

Les acteurs de notre système sont classés en deux catégories.

- ✓ **Le fournisseur de services:** c'est le propriétaire du service Web. Son rôle et la publication du service web à l'aide de formats standardisés (XML, WSDL-web service description language).

- ✓ **L'utilisateur de services :** c'est celui qui va demander un service Web. Il est composé par l'application qui va rechercher et sélectionner un service Web. L'application cliente peut être elle-même un service Web.

III. Diagramme de cas d'utilisation :

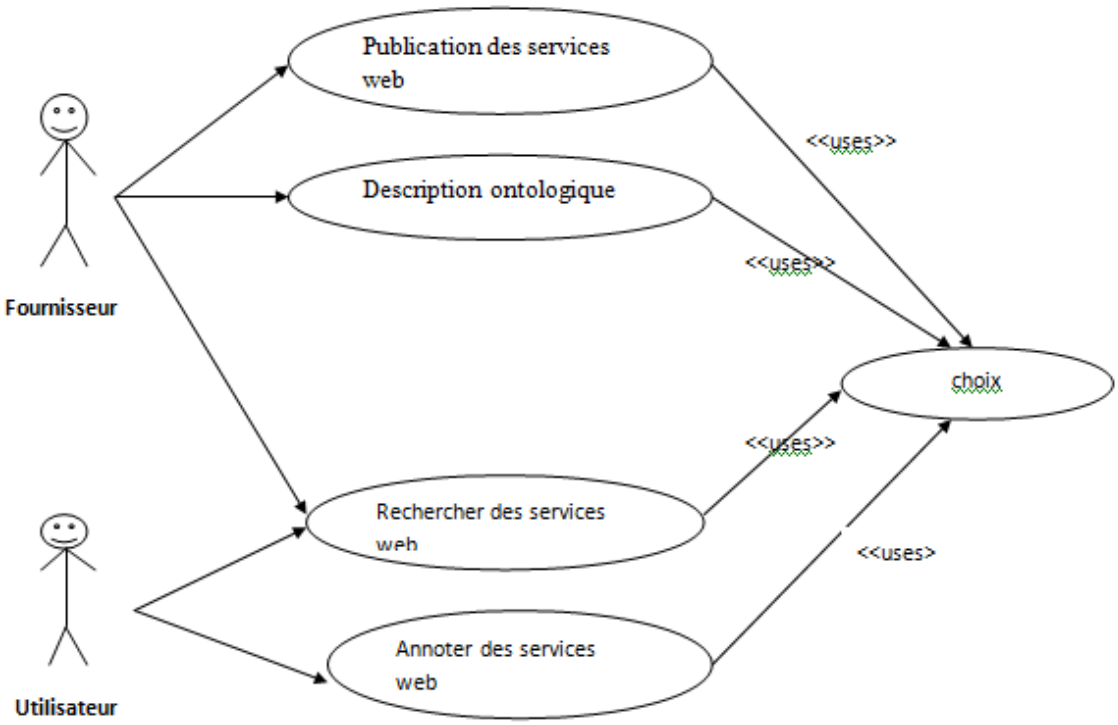

Figure IV.1. Diagramme de cas d'utilisation

IV. Description et fonctionnalités du système :

Le système basé sur cette approche proposée est un système de recherche de services web qui se base sur l'annotation par des tags. Les fonctionnalités de ce système sont :

- ✓ La découverte de services web adaptés à l'utilisateur en utilisant l'annotation des services par des tags.
- ✓ Annotation des services web par les utilisateurs en utilisant les tags.

Ces fonctionnalités permettent d'exploiter l'activité d'annotation des services par des tags pour améliorer le processus de découverte et offrent aux utilisateurs l'accès le plus pertinent aux ressources.

Dans ce mémoire on suppose que les services ont étés déjà publiés et à défaut d'avoir un registre de services web on utilise une base de données contenant les noms des services, leurs descriptions et leurs URI qu'on a retiré des fichiers WSDL de chaque service.

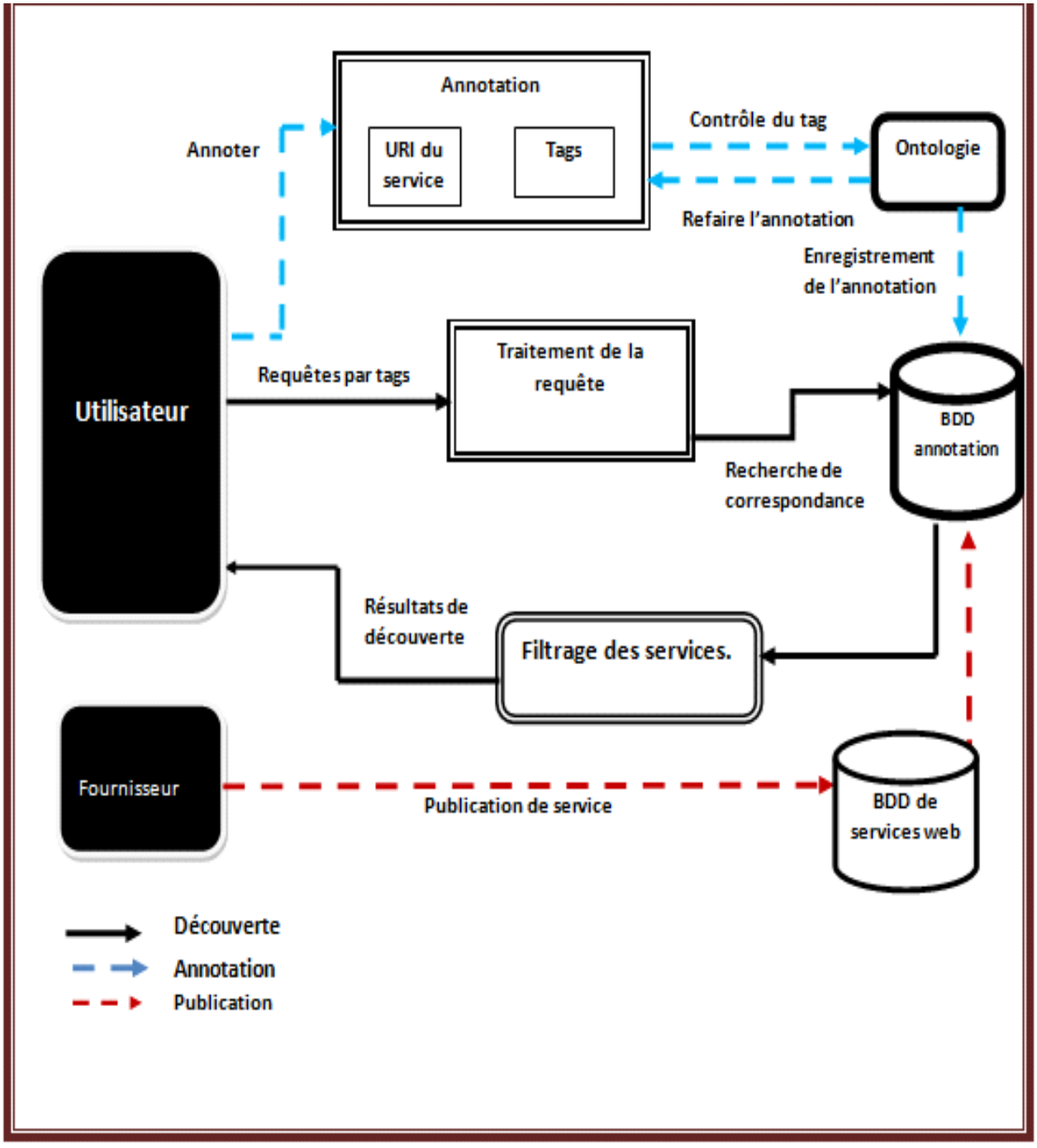

Figure IV.2. Description du système

IV.1. Phase publication :

La phase publication se fait par le fournisseur qui diffuse les descriptions de ses services Web dans l'annuaire de services web.

IV.2. Phase d'annotation sémantique :

Cette phase comporte plusieurs étapes, comme le montre la **figure IV.3**:

- (1) L'utilisateur clique sur le bouton annoter. L'utilisateur sera renvoyé vers une autre page qui est annotation.php.

- (2) L'utilisateur aura deux champs à remplir, le champ annotation ou il va écrire son annotation et un champ url ou il mettra l'adresse du service. Lorsque l'utilisateur a fini d'annoter toutes les ressources, il envoie son annotation au système.

- (3) le système effectuera un contrôle par rapport à l'ontologie my_ontology.owl.

- (4) Si l'annotation et valide elle sera insérée dans la table annotation de la base de donné.

- (5) Si l'annotation n'est pas valide un message d'erreur sera retourné à l'utilisateur.

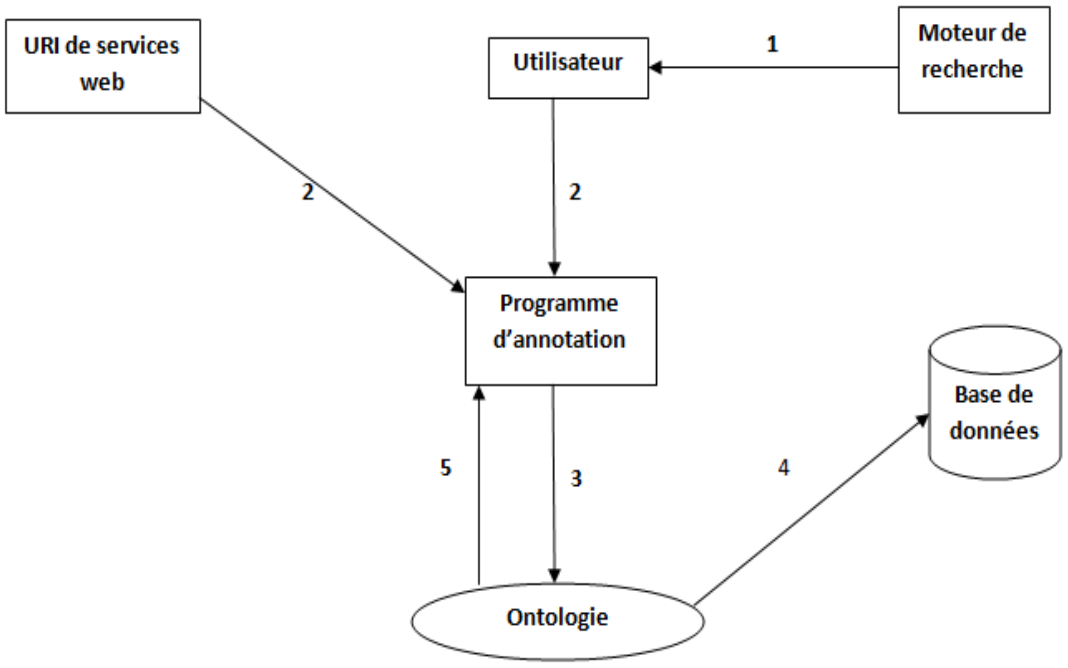

Figure IV.3. Phase d'annotation

IV.3. Diagramme de séquence du module d'annotation:

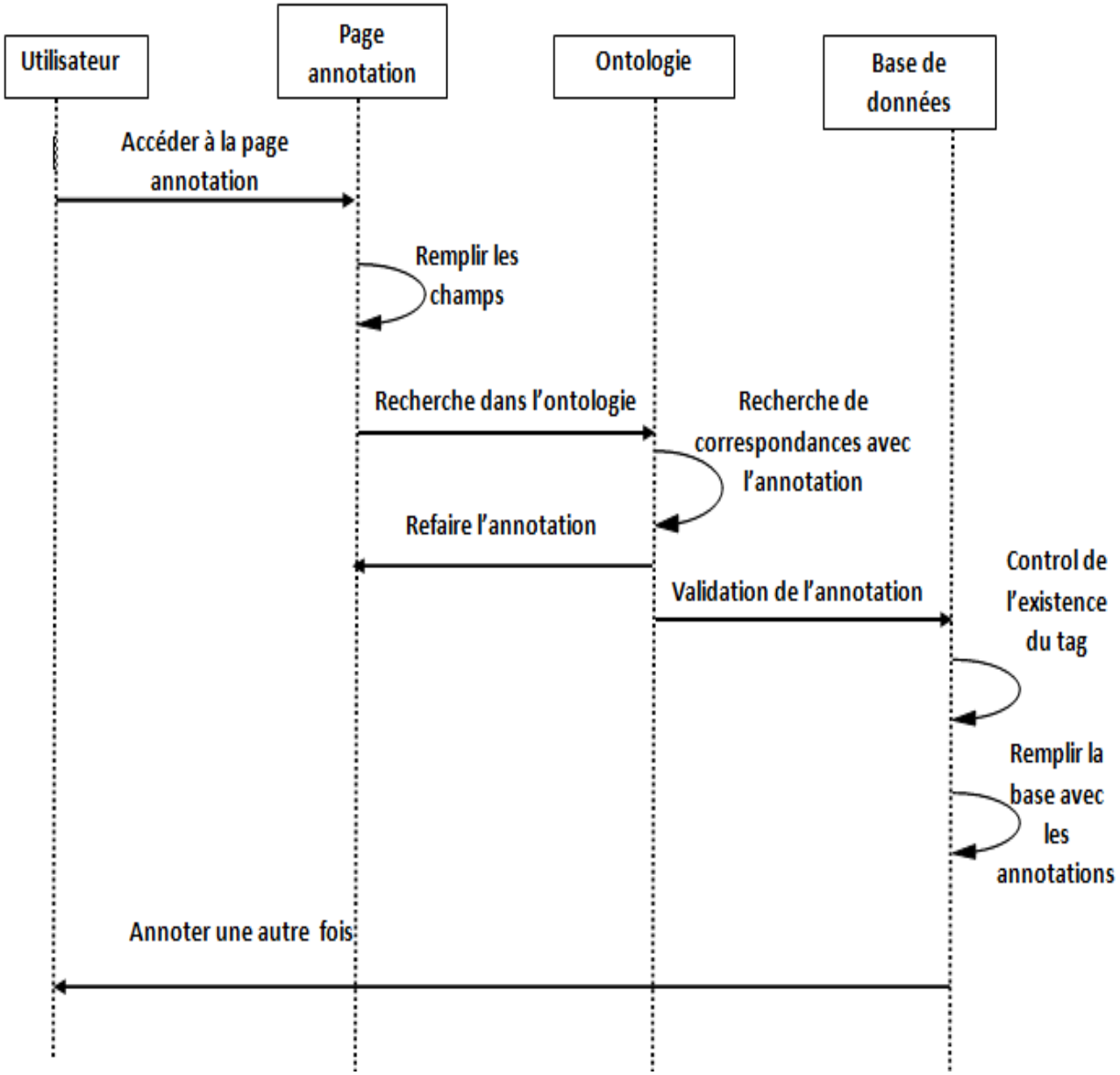

Figure IV.4. Diagramme de séquence du module d'annotation

IV.4. Découverte de Services Web:

Cette fonctionnalité a pour but de découvrir les services web selon des tags qui ont étés désambiguïsé avec une ontologie des tags à fin de renforcer la similitude entre une requête utilisateur et un service web annoté.

-(1) l'utilisateur tape sa requête dans le champ de recherche.

-(2) le moteur fait une connexion a une base de données afin de chercher les services annotés correspondant à la requête.

-(3) un filtrage est fait par rapport au poids des tags.

-(4) le résultat de la recherche sera sous forme de tableau où seront classés les services selon les poids des tags du plus fort au plus faible.

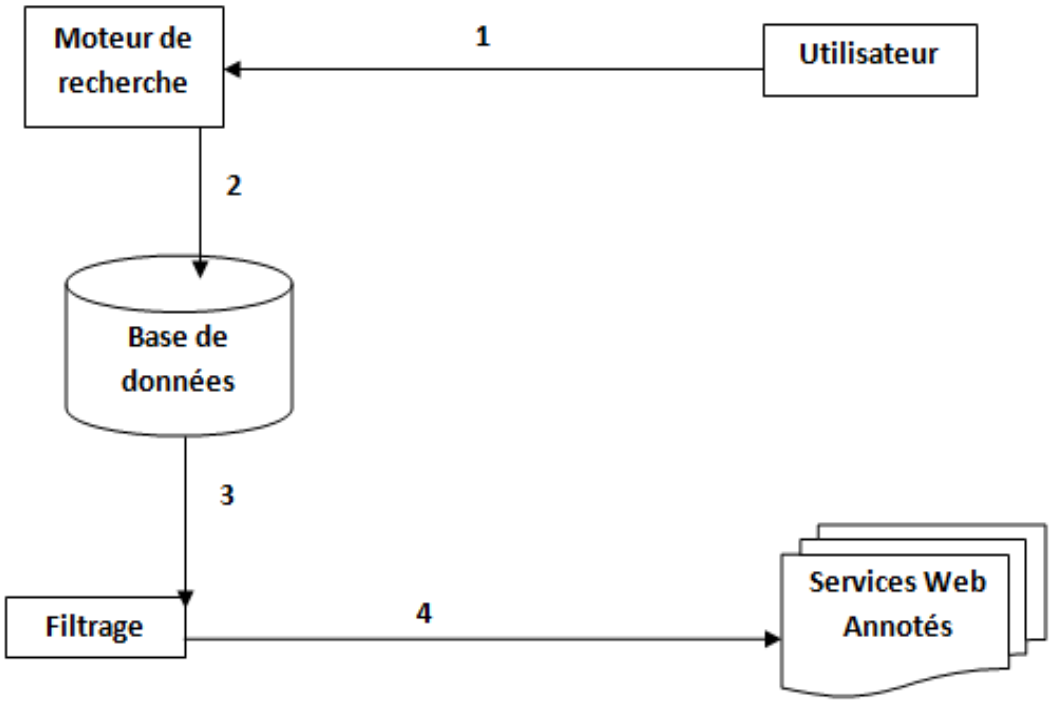

Figure IV.5. Phase découverte

IV.5. Diagramme de séquence du module de découverte :

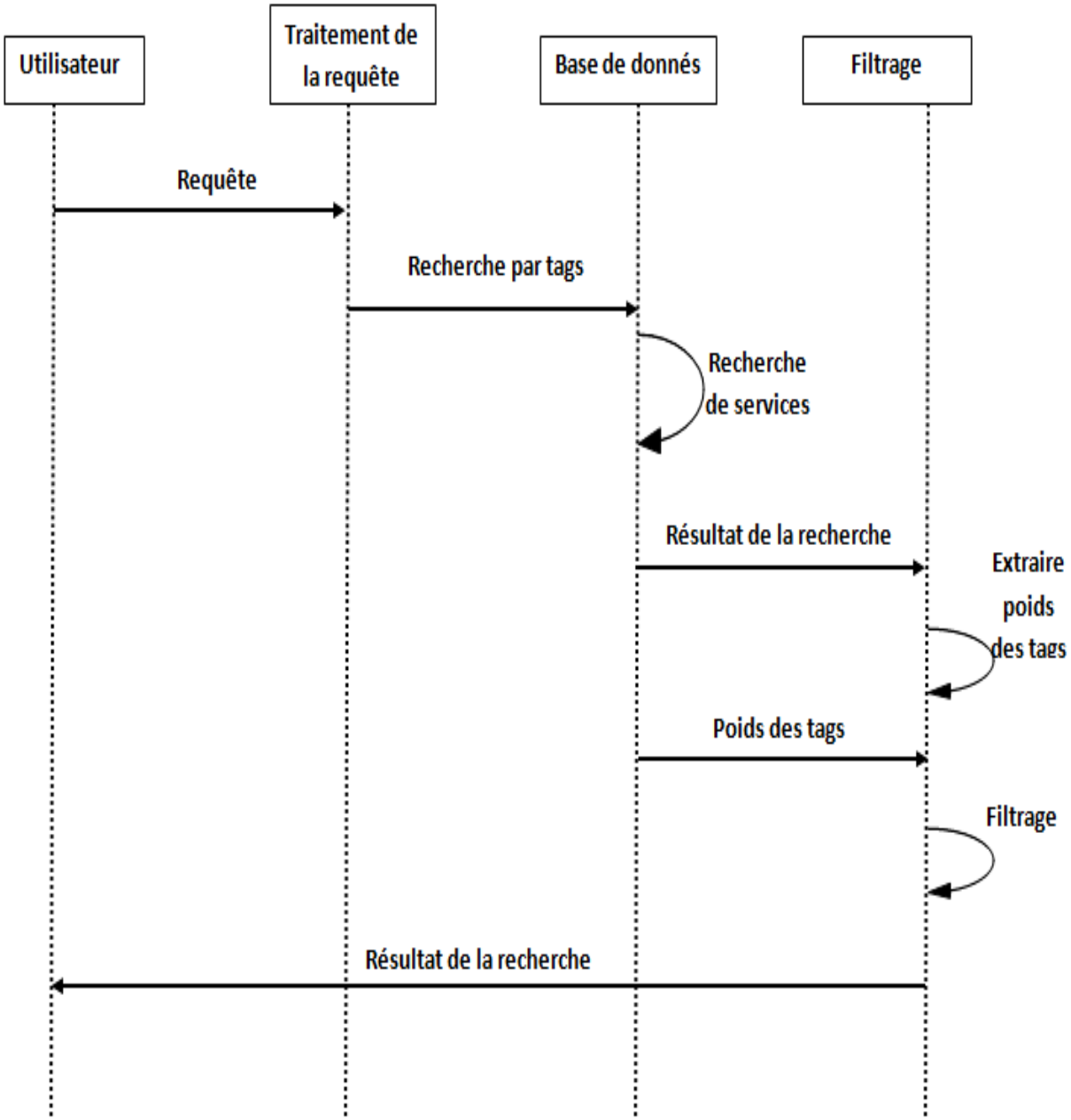

Figure IV.6. Diagramme de séquence du module de découverte

IV.6. Modèle relationnel :

Le modèle relationnel représente la base de données comme un ensemble de tables, sans préjuger de la façon dont les informations sont stockées dans la machine. Les tables constituent donc la structure logique du modèle relationnel. Au niveau physique, le système est libre d'utiliser n'importe quelle technique de stockage (fichiers séquentiels, indexage, adressage dispersé, séries de pointeurs, compression…) dès lors qu'il est possible de relier ces structures à des tables au niveau logique. Les tables ne représentent donc qu'une abstraction de l'enregistrement physique des données en mémoire. De façon informelle, le modèle relationnel peut être défini de la manière suivante :

- ✓ les données sont organisées sous forme de tables à deux dimensions, encore appelées relations, dont les lignes sont appelées n-uplet ou tuple en anglais ;
- ✓ les données sont manipulées par des opérateurs de l'algèbre relationnelle ;
- ✓ l'état cohérent de la base est défini par un ensemble de contraintes d'intégrité.

Dans notre application nous avons utilisé deux tables Service et Tags.

- ✓ **Services** (**id,** Nom_service, description, URI).
- ✓ **Tags** (**Id**, tags, poids).

Les attributs Nom_service, description et URI ont étés retiré des fichiers WSDL de chaque service. Ces fichiers WSDL ont étés eux même retiré de la collection Owls_tc4 qui sera détaillé dans le chapitre suivant.

IV.7. Les tables de la base de données:

Les informations manipulées sur les tags doivent être stockées dans différentes tables d'une base de données, pour notre cas la base porte le nom de « Moteur », les noms attribués à ses tables et à leurs différents champs doivent respecter certaines règles.

IV.7.1. Table Service :

Les données de cette table sont introduites manuellement.

Colonne	**Type**	***Description***	***Clef***	***Observation***
Id	Int (10)	Identifiant du service	Primaire	/
Service	Varchar(30)	Nom du service	/	/
Description	Varchar(90)	Description du service	/	/
URI	Varchar(50)	URI ou localisation du service	/	/

Tableau IV.1. Table Service

IV.7.2. Table Tags :

Cette table sera remplie automatiquement par les tags validés par le système.

- ✓ L'attribut **Id** correspond à id de la table Service.
- ✓ L'attribut tags correspond à l'annotation de l'utilisateur.

✓ L'attribut Poids correspond au nombre de fois que le tag se répète.

Colonne	***Type***	***Description***	***Clef***	***Observation***
Id	int(10)	Identifiant du service	Primaire	/
Tags	Text (25)	Annotation du service	/	/
Poids	Int (3)	Poids du tag	/	/

Tableau IV.1. Table Tags

IV.8. Ontologie :

L'ontologie utilisée dans notre approche est une ontologie nommée my_ontology.owl comme le montre la **Figure IV.7** qui fais elle-même appel à d'autres ontologie comme book.owl, concept.owl, portal.owl et simplified_sumo.owl issues de la collection de test owls-tc4 qui donne des informations sur les fournisseurs de services web avec un ensemble de classes décrivant les propriétés et les caractéristiques de ces services. OWL-S facilite l'automatisation de l'utilisation des services web, y compris la découverte et la sélection d'un service, l'invocation de ce service, l'interopération et la composition de services.

OWL-S s'appuie sur le langage OWL qui est une recommandation du Web Ontology Working Group du World Wilde Web Consortium.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rdf:RDF [
<!ENTITY books.owl "http://127.0.0.1/ontology/books.owl">
<!ENTITY concept.owl "http://127.0.0.1/ontology/concept.owl">
<!ENTITY my_ontology "http://127.0.0.1/ontology/my_ontology.owl#">
<!ENTITY my_ontology.owl "http://127.0.0.1/ontology/my_ontology.owl">
<!ENTITY owl "http://www.w3.org/2002/07/owl#">
<!ENTITY portal.owl "http://127.0.0.1/ontology/portal.owl">
<!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#">
<!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#">
<!ENTITY simplified_sumo.owl "http://127.0.0.1/ontology/simplified_sumo.owl">
<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">
]>
<rdf:RDF xml:base="&my_ontology.owl;"
xmlns:my_ontology="&my_ontology;"
xmlns:owl="&owl;"
xmlns:rdf="&rdf;"
xmlns:rdfs="&rdfs;">

<!-- Ontology Information -->
<owl:Ontology rdf:about="">
<owl:imports>
<owl:Ontology rdf:about="&books.owl;"/>
</owl:imports>
<owl:imports>
<owl:Ontology rdf:about="&concept.owl;"/>
</owl:imports>
<owl:imports>
<owl:Ontology rdf:about="&portal.owl;"/>
</owl:imports>
<owl:imports>
<owl:Ontology rdf:about="&simplified_sumo.owl;"/>
</owl:imports>
</owl:Ontology>

<!-- Classes -->
<owl:Class rdf:about="#1PersonBicycle">
<rdfs:subClassOf rdf:resource="#Bicycle"/>
<rdfs:subClassOf>
<owl:Restriction>
<owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:minCardinality>
<owl:onProperty rdf:resource="#Person"/>
</owl:Restriction>
</rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:about="#2PersonBicycle">
<rdfs:subClassOf rdf:resource="#Bicycle"/>
<rdfs:subClassOf>
<owl:Restriction>
<owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">2</owl:minCardinality>
```

Figure IV.7. Extrait de my_ontology.owl

A partir de cette ontologie nous avons créé un fichier XML nommé base_index qui contient tous les concepts et propriétés contenus dans cette dernière comme le montre la figure suivante :

```
<?xml version="1.0" ?>
- <base_index>
  - <concept ID="#4WheeledCar" ontology="my_ontology.owl">
    - <concepts_asc>
        <asc ID="#Car" ontology="my_ontology.owl" />
        <asc ID="#Auto" ontology="my_ontology.owl" />
        <asc ID="#Machine" ontology="my_ontology.owl" />
        <asc ID="#PeopleTransporter" ontology="my_ontology.owl" />
        <asc ID="#WheeledVehicle" ontology="my_ontology.owl" />
        <asc ID="#Object" ontology="simplified_sumo.owl" />
        <asc ID="#Transporter" ontology="my_ontology.owl" />
        <asc ID="#Vehicle" ontology="my_ontology.owl" />
        <asc ID="#Thing" ontology="owl" />
      </concepts_asc>
    - <proprietes_asc>
        <ascprop ID="#Rigid" />
        <ascprop ID="#Wheel" />
        <ascprop ID="#Shape" />
        <ascprop ID="#Person" />
        <ascprop ID="#Color" />
        <ascprop ID="#LifeTime" />
        <ascprop ID="#Engine" />
        <ascprop ID="#Power" />
        <ascprop ID="#Speed" />
        <ascprop ID="#madeBy" />
        <ascprop ID="#belongsTo" />
        <ascprop ID="#Model" />
        <ascprop ID="#hasValue" />
        <ascprop ID="#Profitable" />
      </proprietes_asc>
    - <chemins>
      - <chemin>
          <arete ID="#Car" />
          <arete ID="#Auto" />
          <arete ID="#Machine" />
          <arete ID="#Object" />
        </chemin>
      </chemins>
    </concept>
  - <concept ID="#1PersonBicycle" ontology="my_ontology.owl">
    - <concepts_asc>
        <asc ID="#Bicycle" ontology="my_ontology.owl" />
```

Figure IV.8. Extrait de base_index.xml

Le but de ce fichier étant de faire un control sur les concepts pour permettre la désambigüisation des annotations saisie par les utilisateurs pour ensuite pouvoir les valider.

L'utilisateur qui voudra annoter un service devra entrer l'url du service et une annotation pour ce dernier. Cette annotation sera alors validée ou non selon le résultat du control effectué avec le fichier base_index.xml.

V. Annotation et découverte par mots-clés :

Un service Web peut être marqué par différents mots-clés fournis par les différents utilisateurs. Un poids du tag se réfère au nombre d'occurrence de ce tag associé à un service Web. Plus ce poids est important, plus le service Web devient important, compte tenu de l'étiquette correspondante. Tous les tags inscrits au système forment la tag-collection. Chaque tag de la tag-collection est associé à un certain nombre de ressources, formant un vecteur de ressources.

Dans le cas de l'annotation par un seul mot clé, chaque tag est considéré comme indépendant de l'autre. Cependant, nous proposons 3 combinateurs pour aider les utilisateurs à formuler des requêtes avancées: AND, OR et NOT. L'algorithme de base utilisé pour découvrir les services Web balisés est présenté dans la **Figure IV.9.**

V.1. Algorithme de recherche par un seul mot-clé :

```
Let Q: a query tag.
TagC: a tag collection.
t: a tag in the tag collection

WSvect: the vector of resources associated to a tag.

Step 1: Exact_matching (Q)
Foreach t in TagC do {
If (IsSameString (Q, t)) {
      return WSvect;
}
}
```

Figure IV.9. Algorithme de recherche par un seul mot-clé

Dans notre cas la recherche se fera à partir de la base de données décrite précédemment où la collection de tags représente la table annotation où sont enregistrés les tags des utilisateurs, les requêtes utilisées c'est des requêtes SQL qui cherche dans la table annotation le tag correspondant au tag introduit dans la barre de recherche.

Le résultat d'une découverte basée sur mot clé unique est classé en fonction de l'importance du poids de l'étiquette associée aux ressources.

Le combinateur AND est utilisé pour filtrer le résultat de la découverte en croisant différents ensembles de résultats liés à chaque étiquette ensemble. Le combinateur OR est utilisé pour calculer l'union des ensembles de résultats et le combineur NOT permet de calculer la différence entre les jeux de résultats.

Par exemple, considérons séparément le résultat de deux découvertes par des mots-clés "Tagi" et "Tagj" comme suit :

Tag_i	$WebService_1$	$Tag\text{-}weight_1$
	$WebService_2$	$Tag\text{-}weight_2$
	$WebService_3$	$Tag\text{-}weight_3$

Tag_j	$WebService_2$	$Tag\text{-}weight_A$
	$WebService_3$	$Tag\text{-}weight_B$
	$WebService_5$	$Tag\text{-}weight_C$

Le résultat d'une requête " Tag_i ET Tagj" peut être représenté de cette façon :

Tag_i	$WebService_2$	$Tag\text{-}weight_2$
	$WebService_3$	$Tag\text{-}weight_3$
Tag_j	$WebService_2$	$Tag\text{-}weight_A$
	$WebService_3$	$Tag\text{-}weight_B$

Le résultat d'une requête " Tag_i OU Tag_j " peut être représenté de cette façon:

tag_i	$WebService_1$	$Tag\text{-}weight_1$

	$WebService_2$	$Tag\text{-}weight_2$
	$WebService_3$	$Tag\text{-}weight_3$
Tag_j	$WebService_2$	$Tag\text{-}weight_A$
	$WebService_3$	$Tag\text{-}weight_B$
	$WebService_5$	$Tag\text{-}weight_C$

Le résultat d'une requête "NOT Tagi" est l'ensemble des services Web qui ne contiennent pas les services qui sont présents dans l'ensemble du "tag" de la requête de résultat.

Les utilisateurs peuvent annoter et découvrir une ressource en utilisant les balises de mots clés composés. Ce type de balise peut être reconnu comme une seule chaîne (concaténation de mots-clés). Le processus de découverte respecte presque le même algorithme trouvé dans la **Figure IV.9**.

Dans notre approche nous nous contentons des annotations et découverte par un seul mot-clé.

VI. Conclusion :

Ce chapitre a été consacré à la description de notre contribution qui a pour but d'exploiter l'activité de l'annotation lors de la découverte des services web adaptés aux besoins et spécificités des usagers.

L'architecture que nous proposons est basée sur deux modules, un module d'annotation reposant sur une ontologie qui permet de valider et de contrôler les tags des utilisateurs et un autre module qui permet la découverte de services web fondée sur les annotations précédentes des utilisateurs. Enfin, les résultats seront affichés selon le poids des tags du plus fort au plus faible. Le chapitre suivant est consacré à l'implémentation du système.

Chapitre IV

I. Introduction :

La découverte des services web est une étape primordiale, dans le travail réalisé dans le cadre de notre projet vise à implémenter une approches de découverte des services web, nous décrivons en premier lieu les outils utilisés ensuite nous présenterons quelques interfaces de notre application.

II. Outils de développement :

Pour développer notre architecture, nous avons choisi Netbeans comme environnement de développement dans sa version 8.1 téléchargeable à partir de : http://www.netbeans.org.

II.1.Langages de programmation :

II.1.1.HTML :

L'HyperText Markup Language, généralement abrégé HTML ,ce qui signifie en français langage de balisage d'hypertexte», est le format de données conçu pour représenter les pages web. C'est un langage de balisage permettant d'écrire de l'hypertexte, d'où son nom. HTML permet également de structurer sémantiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie, et des programmes informatiques. Il permet de créer des documents interopérables avec des équipements très variés de manière conforme aux exigences de l'accessibilité du web**. [44]**

II.1.2.PHP :

PHP (officiellement "PHP: HypertextPreprocessor") est un langage de script qui est principalement utilisé pour être exécuté par un serveur HTTP, mais il peut fonctionner comme n'importe quel langage interprété en utilisant les scripts et son interpréteur sur un ordinateur. Il permet aux développeurs Web d'écrire des pages dynamiques rapidement. **[45]**

PHP n'est pas un langage compilé, c'est un langage interprété par le serveur : le serveur lit le code PHP, le transforme et génère la page HTML. Pour fonctionner, il a donc besoin d'un serveur Web. De ce fait une plateforme minimale de base pour l'exécution d'un site Web développé en PHP comprend :

- ✓ l'interpréteur PHP (serveur PHP) ;
- ✓ un serveur Web (Apache, IIS, …).

Le langage PHP possède les mêmes fonctionnalités que les autres langages permettant d'écrire des scripts CGI, comme collecter des données, générer dynamiquement des pages Web ou bien envoyer et recevoir des cookies, … .

Il a les avantages suivants :

- la plus grande qualité et le plus important avantage du langage PHP est le support d'un grand nombre de bases de données et la simplicité d'interfaçage avec eux.
- PHP est utilisable sur la majorité des systèmes d'exploitation, comme Linux, de nombreuses variantes Unix (incluant HP-UX, Solaris et OpenBSD), Microsoft Windows, Mac OS X, RISC OS et d'autres encore.
- PHP supporte aussi la plupart des serveurs Web actuels : Apache, Microsoft Internet Information Server, Personal Web Server, Netscape et iPlanet servers, OreillyWebsite Pro server, Caudium, Xitami, OmniHTTPd et beaucoup d'autres encore.
- la gratuité et la disponibilité du code source.
- la simplicité d'écriture de scripts (apprentissage rapide)

Malgré ses avantages il a les limitations suivantes :

- Langage interprété ;
- L'orienté objet reste limité ;
- Pas adéquat en termes de rapidité et de maintenabilité pour des projets de grandes envergures.

III.1.3. XML :

Le langage XML est un langage qui permet de décrire des données à l'aide de balises et de règles que l'on peut personnaliser. La force de XML réside dans sa capacité à pouvoir décrire n'importe quel domaine de données grâce à son extensibilité. Il va permettre de structurer, poser le vocabulaire et la syntaxe des données qu'il va contenir. En réalité les balises XML décrivent le contenu plutôt que la présentation (contrairement À HTML). Ainsi, XML permet de séparer le contenu de la présentation. Ce qui permet par exemple d'afficher un même document sur des applications ou des périphériques différents sans pour autant nécessiter de créer autant de versions du document que l'on nécessite de représentations. **[46]**

II.1.4.Le langage de requête SQL :

SQL (Structured Query Language, ce qui signifie en français Langage de Requêtes Structuré) est un langage de définition de données (LDD, ou en anglais DDL Data Définition Language), c'est-à-dire qu'il permet de créer des tables dans une base de données relationnelle, ainsi que d'en modifier ou en supprimer. C'est aussi un langage de manipulation de données (LMD, ou en anglais DML, Data Manipulation Language), et un langage de contrôle de données (LCD, ou en anglais DCL, Data Control Language), pour les bases de données relationnelles, cela signifie qu'il permet de sélectionner, insérer, modifier ou supprimer des données dans une table d'une base de données relationnelle. **[47]**

II.2. Les serveurs :

II.2.1. XAMPP:

XAMPP est un ensemble de logiciels permettant de mettre en place facilement un serveur Web et un serveur FTP. Il s'agit d'une distribution de logiciels libres (X Apache MySQL Perl PHP) offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide. Ainsi, il est à la portée d'un grand nombre de personnes puisqu'il ne requiert pas de connaissances particulières. **[48]**

II.2.2. Serveur web apache :

Le serveur web Apache, apparu en 1995, est le plus populaire des serveurs web sur Internet. Il a été créé dans la volonté de développer et de maintenir un serveur HTTP sécurisé, efficace et évolutif pour les systèmes d'exploitation modernes.

Apache est multi plate-forme et gratuit, son installation est facile, rapide et son utilisation n'est pas compliqué.

Grace à une association avec PHP, apache devient un serveur web dynamique.

II.2.3.Serveur base de donnéesMySQL:

MySQL est un système de gestion de bases de données relationnelles (SGBDR) libre, fonctionnant sous diverses plates-formes telles que UNIX, Linux et Windows, et permettant de manipuler des instructions adressées à la base de données Sous forme de requêtes SQL.Il fait partie des logiciels de gestion de base de données les plus utilisés au

monde, autant par le grand public (applications web principalement) que par des professionnels**. [49]**

II.3.Ontologie:

II.3.1.OWL:

OWL (Web Ontology Language) est le langage standard pour la représentation d'ontologies, proposé par le consortium W3C. OWL utilise les primitives de base modélisées par les schémas RDF. OWL permet de définir des classes qui représentent des ensembles d'individus, et des propriétés qui représentent les caractéristiques des individus ou les relations possibles entre ces derniers. Les connaissances sont prises en compte selon deux niveaux d'abstraction : la représentation des concepts et des relations relève du niveau terminologique, tandis que la représentation des individus et les assertions dans lesquelles ils interviennent relèvent du niveau assertionnel.

Pour répondre à des besoins de représentation et d'inférence variés, exprimés à travers différentes applications, le consortium W3C propose trois sous langages : OWL Lite, OWL DL et OWL Full, énumérés ici par ordre décroissant d'expressivité comme suit :

- ✓ **OWL Full**: est la version la plus complexe d'OWL, celle qui offre le plus haut niveau d'expressivité. Elle autorise l'utilisation de toutes les primitives OWL combinées d'une façon arbitraire aux structures RDF et RDF Schema.
- ✓ **OWL DL:** il s'adresse aux utilisateurs qui souhaitent une expressivité maximale sans perdre la complétude et la décidabilité du calcul de subsomption.
- ✓ **OWL Lite**: est le langage le plus simple (il est moins expressif que les deux autres), il est décidable et complet. Il s'adresse aux utilisateurs qui ont principalement besoin d'hiérarchies de classification et de constructeurs élémentaires.

De nombreux outils pour la construction des ontologies ont été développés ces dernières années. Les plus largement utilisés par la communauté ingénierie des ontologies sont d'OILEd, KAON, Jena et Protégé. Pour l'inférence, il existe plusieurs moteurs d'inférence, la plupart conçus pour raisonner sur les Logiques de Description, mais qui acceptent en entrée des fichiers OWL/RDF(S). Parmi ceux-ci, nous citons RacerPro, Pellet, Fact, Fact++, Surnia, F-OWL, Hoolet, etc. Certains moteurs d'inférence ne peuvent raisonner qu'au niveau terminologique, alors que des moteurs comme Pellet et RacerPro permettent de raisonner aussi sur les instances de concepts. **[50]**

II.3.2.Collections OWLS-TC4 :

Les données utilisées dans nos expérimentations sont issues d'un extrait du corpus open source OWLS-TC version 2.2.1, Ce dernier est développé par le centre allemand pour la recherche en intelligence artificielle (http://www.dfki.de/scallops). Cette base de données est constituée de 1007 descriptions de service définis sous forme OWL-S. Elles sont classées en sept domaines différents : le domaine d'éducation, le domaine médical, le domaine de nourriture, le domaine militaire, le domaine de voyages (tourisme), le domaine de communication, et le domaine d'économie.

La plupart de ces services sont extraits de l'annuaire UDDI d'IBM, et sont traduits du format WSDL en format OWLS de façon semi-automatique. La base contient aussi un ensemble de requêtes reparties sur les 07 classes, ces requêtes sont décrites en OWLS. Chaque service web (i.e. document OWLS) est classé manuellement par des experts humains comme étant pertinent ou non par rapport à une requête donnée. La base offre aussi un ensemble d'ontologies pour décrire les services et les requêtes.

III. Interfaces de notre prototype :

III.1.Interface de Recherche :

Figure V.1.Interface de recherche de services web.

III.2.Interface Résultats de la recherche par tags:

Figure V.2.Interface Résultats de la recherche par tags.

III.3. Interface Résultats de la recherche par noms de services:

Figure V.3.Interface Résultats de la recherche par nom de services.

III.4. Interface Annotation :

Figure V.4. Interface Annotation.

III.5. Interface Annotation validée :

Figure V.5. Interface Annotation validée.

III.6. Interface Annotation Non valide :

Figure V.6. Interface Annotation Non valide.

III.7. Interface Annotation non introduite:

Figure V.7. Interface Annotation Non introduite.

IV. Conclusion :

Ce chapitre a été consacré à la mise en œuvre et la validation de notre approche de découverte qui a pour but d'exploiter l'activité d'annotation des services par des tags. Nous avons présenté les détails des travaux expérimentaux de notre approche ce qui nous a confirmé qu'elle est plus efficace en termes de précision que le système de recherche classique que nous avons illustré afin de montré la différence.

Conclusion générale

Conclusion générale :

Les services Web sont la plus récente technologie adoptée pour l'intégration et l'interopérabilité des systèmes répartis. Ils sont caractérisés par leurs indépendances aux plateformes et aux systèmes d'exploitation, ce qui a impliqué leur adoption par les différentes entreprises et organisations commerciales et industrielles, ces organisations offrant leurs services à distance via le Web, ce qui augmente le nombre de services offerts (publies). Et par conséquence la tache de découverte de services web est devenue très difficile.

Dans notre travail on a proposé une découverte qui se base principalement sur les annotations par des tags, exprimées par les utilisateurs. Notre proposition prend quelques paramètres au cours du processus de découverte comme les annotations des utilisateurs et leurs validités.

Le premier chapitre est consacré à l'état de l'art en général et plus particulièrement aux principaux standards liés à la technologie des services web sémantiques, ainsi que leurs avantages et leurs vulnérabilités.

Dans le deuxième chapitre, nous avons présenté, particulièrement deux problématiques importantes dans le domaine des services web sémantiques : la découverte de services, et la sélection de services.

Les deux derniers chapitres sont consacrés à notre contribution, nous avons proposé d'orienter la découverte de service web selon les annotations exprimées par les utilisateurs de ces services.

Perspectives :

La majorité des travaux portant sur la découverte ne prennent généralement pas en considération la qualité de service et l'avis des utilisateurs**.** Ces données peuvent être utilisées intensivement pour personnaliser la recherche du service web adéquat.

Plusieurs améliorations et extensions peuvent être envisagées pour enrichir l'approche de découverte.

- ✓ L'utilisation de langages complet et riche en outils est primordiale afin d'avoir de meilleurs résultats.
- ✓ Appliquer cette approche sur le terrain avec des données réelles, ce que va améliorer les ontologies proposées.
- ✓ L'attribution des rôles : proposer et implémenter une fonction propre qui permet l'attribution des rôles d'une façon efficace et optimale.
- ✓ Automatiser toutes les étapes de l'approche proposée qui ne sont pas déjà automatisées
- ✓ Accepter des annotations avec mot composés.

BIBLIOGRAPHIE

[1] Les Services Web disponible sur :http://www-igm.univ-mlv.fr/~dr/XPOSE2004/woollams/definition.html

[2] Fabrice Rossi, Services Web. Université Paris-IX Dauphine disponible sur : http://apiacoa.org/publications/teaching/webservices/Introduction.pdf

[3] Le XML disponible sur :http://glossaire.infowebmaster.fr/xml/

[4] XML (eXtensible Markup Language) par Anis Chelly disponible sur **:**

https://prezi.com/essguz6brgmw/xml-extensible-markup-language

[5] Le XML disponible sur : http://glossaire.infowebmaster.fr/xml/

[6] SOAP Simple Object Access Protocol page 3 disponible sur :

http://www-inf.it-sudparis.eu/cours/middleware/enseignement/projetsEtudiant/0102/rapportSOAP.pdf

[7] Ricardo DE LA ROSA-ROSERO Découverte et Sélection de Services WebMaster pour une application Mélusine, Mémoire de fin d'étude page 18 disponible sur : http://www-adele.imag.fr/Les.Publications/reports/DEA2004Del.pdf

[8] WSDL disponible sur : https://openclassrooms.com/courses/les-services-web

[9] Avantages et inconvénients des services web disponible sur : http://dspace.univ-tlemcen.dz/bitstream/112/1244/1/DEHANE-Aicha-Djihad-KEBIR-Zohra.pdf

[10] Web sémantique disponible sur : http://jplu.developpez.com/tutoriels/web-semantique/introduction/

[11] Les métadonnées disponible sur : https://www.w3.org/People/Berners-Lee/Publications.html

[12] Composants du web sémantique disponible sur : http://www-sop.inria.fr/edelweiss/people/khaledKhelif/docs/these_KHELIF.pdf

[13] Architecture du web sémantique disponible sur : http://bu.univ-ouargla.dz/Chafik_Berdjouh.pdf?idthese=721

[14] Les Services web sémantique disponible sur : http://bu.umc.edu.dz/theses/informatique/GHA6030.pdf

[15] Services web sémantique disponible sur : http://dspace.univ-bouira.dz:8080/jspui/bitstream/123456789/1566/1/s%C3%A9lection%20des%20web%20services%20semantique%20par%20boudjetaba%20hakim.pdf

[16] RDF disponible sur : http://www-igm.univ-mlv.fr/~dr/XPOSE2009/Le%20Web%203.0/technologies.html

[17] OWL-S disponible sur : https://www.w3.org/Submission/OWL-S/

[18] Semantic Annotations for WSDL disponible sur: https://www.w3.org/TR/sawsdl/

[19] WSMO disponible sur: http://dspace.univ-tlemcen.dz/bitstream/112/8406/1/Mediation_semantique_de_donnees_basee_sur_les_services_web%20_Application_dans_le_domaine_medical.pdf

[20] Les avantages de l'utilisation du Web sémantique pour la description des services Web disponible sur**:** http://www.memoireonline.com/12/13/8382/m_Mashup-semantique50.html

[21] thèse de doctorat en informatique, Une approche de composition de services Web à l'aide des Réseaux de Petri orientés objet, Présentée par : Sofiane Chemaa dispo sur http://www.univ-constantine2.dz/files/Theses/Informatique/Doctorat/Chemaa-Sofiane.pdf

[22] Article SEMANTIC ANNOTATIONS IN WEB SERVICES Meenakshi Nagarajan Large Scale Distributed Information Systems (LSDIS).

[23] A-A. Patil, S.A.Oundhakar, A.P.Sheth, and K.Verma. Meteor-s web service annotation framework. In WWW, 2004.

[24] [Hess et al. 2004a], Machine Learning for Annotating Semantic Web Services. 2004a.

[25] [Hess et al. 2004b], ASSAM: A Tool for Semi-Automatically Annotating Semantic Web Services. 2004b.

[26] Oldham N et al, METEOR-S Web Service Annotation Framework with Machine Learning Classification. Proceedings of the 1st International Workshop on Semantic Web Services and Web Process Composition (SWSWPC'04), In Conjunction with the 2nd International Conference on Web Services (ICWS'04). 2004.

[27] [Schumacher et al., 2008] M. Schumacher, H. Helin, and H. Schuldt. Semantic Web Service Coordination. CASCOM: Intelligent Service Coordination in the Semantic Web, Birkhäuser Basel, 2008.

[28] Palathingal et Chandra, System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on

[29] Klein et König-ries, Coupled signature and specification matching for automatic service binding. In Proceedings of European Conference on Web Services (ECOWS 2004), LNCS. 3250. Springer, Erfurt, Germany, September. 2004.

[30] N.Srinivasan, M. Paolucci, and K. Sycara. Semantic Web Service Discovery in the OWL-S IDE. In System Sciences, 2006. Proceedings of the HICSS '06. Hawaii. USA. 2006.

[31] J. Fan, B. Ren, and L.-R. Xiong. An Approach to Web Service Discovery Based on the Semantics. In FSKD (2), 2005.

[32] Paolucci, M., Srinivasan, N. Sycara, K., Nishimura, T. Toward a Semantic Choreography of Web Services: From WSDL to DAML-S. In: Procs of the International Conference of Web Services (ICWS03), June 2003, Las Vegas, Nevada, USA. CSREA Press 2003.

[33] M.C.Jaeger, G.Rojec-Goldmann, G.Muhl, C.Liebetruth, and K.Geihs. Ranked Matching for Service Descriptions using OWL-S, 2005. .In Paul Muller, Reinhard Gotzhein, and Jens B, editors, Kommunikation in verteilten Systemen (KiVS 2005),Kaiserslautern, Germany, February 2005.

[34] U.Keller, R.Lara, H.Lausen, A.Pollers, and D.Fensel.Automatic location of services. In Proc.of the 2nd European Semantic Web conference (ESWC), Heraklion, Crete, 2005.LNCS 3532.

[35] M.Stollberg, U.Keller, H.Lausen, and S.Heymans. Two-Phase Web Service Discovery Based on Rich Functional Descriptions. In ESWC'07:Proc.of the 4th European conference on the Semantic Web, Berlin, Heidelberg, 2007.

[36] U. Küster and B. Konig-Ries. Evaluating semantic web service matchmaking effectiveness based on graded relevance. In Proc. of the 2nd International Workshop SMR on Service Matchmaking and Resource Retrieval in the Semantic Web at the 7th International Semantic Web Conference (ISWC08), Karlsruhe, Germany, October 2008.

[37] C. Rey and M. Schneider, Mediation in a Dynamic Context: Arguing for a Request-Oriented Approach and Structuring It. . In A. Dahanayake and W. Gerhardt, Web-enabled Systems Integration: Challenges and Practices 2002. Idea Group Publishing.

[38] Kiefer, C., Bernstein, A., Lee, H. J., Klein, M. et Stocker, M. (2007). Semantic process retrieval with iSPARQL. In Proceedings of the 4th European Semantic Web Conference (ESWC '07).

[39] C.Kiefer, and A. Bernstein, The Creation and Evaluation of iSPARQL Strategies for Matchmaking, in The Semantic Web: Research and Applications. 2008.

[40] P. Bertoli, A. Cimatti, and P. Traverso. Interleaving execution and planning for nondeterministic, partially observable domains. In ECAI, 2004.

[41] F.Kaufer and M.Klusch. WSMO-MX: A Logic Programming Based Hybrid Service Matchmaker. In European Conference on Web Services, Los Alamitos, CA, USA, 2006. IEEE Computer Society.

[42] M.Klusch, P.Kapahnke, and F.Kaufer. Evaluation of WSML Service Retrieval with WSMOMX. In IEEE International Conference on Web Services, 2008. ICWS '08, Sept. 2008.

[43] BOUDJELABA Hakim, Sélection des Web Services Sémantiques, Mémoire de Magister disponible sur : http://dspace.univ-bouira.dz:8080/jspui/bitstream/123456789/1566/1/s%C3%A9lection%20des%20web%20services%20semantique%20par%20boudjetaba%20hakim.pdf

[44] Le HTML disponible sur :http://info.massena.free.fr/HTML-expose.pdf

[45] Le langage PHP disponible sur :https://fr.wikipedia.org/wiki/PHP

[46] Langage XML disponible sur : http://www.commentcamarche.net/contents/1332-xml-introduction-a-xml

[47] Langage SQL disponible sur: http://www.commentcamarche.net/contents/1062-le-langage-sql

[48]Le serveur XAMPP disponible sur : https://desgeeksetdeslettres.com/programmation-java/xampp-plateforme-pour-heberger-son-propre-site-web

[49] Jean-Claude CABIANCA, SGBD MySQL disponible sur http://www.jcc40.fr/wp-content/uploads/2012/03/mysql.pdf

[50] OWL Web Ontology Language Guide disponible sur https://www.w3.org/TR/owl-guide/