



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université Mohamed Khider – BISKRA  
Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie  
**Département d'informatique**

N° d'ordre : RTIC 5 /M2/2018

## Mémoire

Présenté pour obtenir le diplôme de master académique en

# Informatique

Parcours : Réseaux et TIC

---

# Une approche basée agent pour la sélection des Services Web

---

Par :

**MEGHERBI SAMMY**

Soutenu le 24/06/2018, devant le jury composé de :

BENDAHMANE Asma	M.A.A	Président
BELOUAAR Houcine	M.A.A	Rapporteur
RAHMANI Salima	M.A.A	Examineur

## Remerciements

Ce mémoire est le résultat d'un travail d'une année de recherche menée à l'Université de Biskra, Département d'Informatique.

Tout d'abord, je tiens à remercier Dieu, le tout puissant et miséricordieux, qui m'a donné la force et la patience pour accomplir ce travail.

Ce mémoire n'aurait pas pu avoir lieu sans un encadrement de grande qualité, celui de Mr. BELOUAAR Hocine pour avoir proposé et dirigé ce sujet de mémoire et pour son accueil bienveillant et ses conseils avisés, il a bien voulu répondre patiemment et aimablement à mes nombreuses questions. Ainsi je le remercie pour son aide précieuse et continue pendant la rédaction de ce mémoire.

Je remercie les honorables membres de jury d'avoir accepté d'être membre de mon jury de thèse, d'évaluer mon travail et pour nous avoir honorés de leur présence.

Je tenais également à remercier tous mes enseignants de la faculté d'informatique à l'université Mohammed kheider Biskra et plus précisément de la branche RTIC.

Enfin j'adresse mes gratitude à tous mes collègues et mes amies.

## Dédicace

Je remercie Dieu de m'avoir donné le courage pour accomplir ce modeste travail que je dédie à :

Ma mère, qui a œuvré pour ma réussite, de par son amour, son soutien, tous les sacrifices consentis et ses précieux conseils, pour toute son assistance et sa présence dans ma vie.

Mon père, qui peut être fier et trouver ici le résultat de longues années de sacrifices et de privations pour m'aider à avancer dans la vie.

À ma sœur et mes cousins Tiko, Souhil, Doudou, Mehdi, Oussama, Houssein . À mes chers amis Dilaw, Dhia, Foufou, Anis .

Aux personnes que j'aime et qui ont été toujours là pour moi.

# Table des matières

<b>Remerciements</b>	1
<b>Dédicace</b>	2
<b>Résumé</b>	vii
<b>Abstract</b>	viii
<b>Introduction Générale</b>	1
<b>1 Service Web</b>	4
<b>1.1 Introduction</b> . . . . .	5
<b>1.2 Architecture Orientée Service (SOA)</b> . . . . .	5
<b>1.2.1 Définition de SOA</b> . . . . .	6
<b>1.2.2 Objectifs</b> . . . . .	6
<b>1.2.3 Avantages d'une architecture orientée service :</b> . . . . .	7
<b>1.3 Services web</b> . . . . .	7
<b>1.3.1 Définition des services web</b> . . . . .	7
<b>1.3.2 Architecture des services web</b> . . . . .	8
<b>1.3.3 Standards de base de services Web</b> . . . . .	11
<b>1.3.3.1 SOAP</b> . . . . .	12
<b>1.3.3.2 SOAP RPC</b> . . . . .	15
<b>1.3.4 La technologie WSDL</b> . . . . .	15
<b>1.3.5 UDDI - Universal Description Discovery and Integration</b> . .	17

1.3.6	Types de structures de données d'un modèle UDDI	18
1.3.7	Types d'annuaires UDDI	19
1.3.8	Les avantages et inconvénient des services Web	20
1.3.8.1	Les avantages	20
1.3.8.2	Les inconvénients	20
1.4	<b>Conclusion</b>	21
<b>2</b>	<b>SMA et Sélection des Services Web</b>	<b>22</b>
2.1	<b>Introduction</b>	23
2.2	<b>Qualité de service (QoS)</b>	23
2.2.1	Définition	23
2.2.2	Critères de qualité de service	24
2.3	<b>La sélection des Services Web</b>	25
2.3.1	Critères pour les méthodes de sélection des services	26
2.4	<b>Méthodes d'aides à la décision multicritère (MCDM)</b>	27
2.4.1	<b>Description de quelques méthodes MCDM</b>	27
2.4.1.1	<b>La Méthode ELECTRE</b>	28
2.4.1.2	<b>La Méthode AHP</b>	29
2.4.1.3	<b>La Méthode TOPSIS</b>	31
2.5	<b>Agents et Systèmes multi-agents</b>	33
2.5.1	<b>La notion d'agent</b>	33
2.5.1.1	Définition d'un agent	33
2.5.1.2	Propriétés des agents	34
2.5.1.3	Types d'agents	34
2.5.2	<b>Les systèmes multi-agents(SMA)</b>	36
2.5.2.1	<b>Définition d'un SMA</b>	36
2.5.2.2	<b>Interactions dans un système multi-agents</b>	36
2.5.2.3	<b>La communication dans un système multi-agents</b>	37

---

<b>2.6 Les agents et les services web</b>	38
<b>2.7 Les travaux connexes</b>	40
<b>2.7.1 Travail N 1</b>	40
<b>2.7.2 Travail N 2</b>	41
<b>2.8 Conclusion</b>	42
<b>3 Conception du système</b>	<b>43</b>
<b>3.1 Introduction</b>	44
<b>3.2 Architecture générale</b>	44
<b>3.2.1 Description des agents</b>	45
<b>3.3 Description</b>	48
<b>3.4 Fonctionnement général du système</b>	49
<b>3.5 Conclusion</b>	51
<b>4 Implémentation et étude de cas</b>	<b>53</b>
<b>4.1 Introduction</b>	54
<b>4.2 Environnement de développement</b>	54
<b>4.2.1 Environnement matériel et logiciel</b>	54
<b>4.2.2 Langage de programmation</b>	54
<b>4.2.2.1 Le langage Java</b>	55
<b>4.2.3 Outils et technologies</b>	56
<b>4.2.3.1 Eclipse</b>	56
<b>4.2.3.2 La plateforme JADE</b>	57
<b>4.2.3.3 Gestion de base de données</b>	59
<b>4.2.4 La base de données</b>	60
<b>4.2.5 Communication entre les agents</b>	62
<b>4.3 Présentation des interfaces graphiques</b>	62
<b>4.3.1 Interface d'accueil :</b>	62
<b>4.3.2 Les interfaces principales de notre prototype :</b>	63

---

4.3.2.1 Interface d'Inscription du Fournisseur :	63
4.3.2.2 Interface d'Inscription du Client :	64
4.3.2.3 Interface Login :	65
4.3.2.4 Interface d'Espace Client :	66
4.3.2.5 Interface d'Espace Fournisseur :	69
<b>4.4 Conclusion</b>	<b>71</b>
<b>Conclusion Générale</b>	<b>72</b>

# Table des figures

1.1 Structure des services web[15]. . . . .	9
1.2 Architecture en Pile des servicesWeb[4]. . . . .	10
1.3 Traitement d'un message SOAP[2].. . . .	13
1.4 Structure de message SOAP[6]. . . . .	14
1.5 Structure et description WSDL[7]. . . . .	16
1.6 Le contenu de l'annuaire UDDI[4]. . . . .	17
1.7 Les structures de données d'UDDI. . . . .	19
2.1 Prise de décision multicritères (MCDM) [26]. . . . .	28
2.2 Table de préférences standard . . . . .	30
2.3 Matrice pour évaluer les poids des critères . . . . .	31
2.4 Architecture d'un agent réactif. . . . .	35
2.5 Architecture d'un agent cognitif. . . . .	35
2.6 Système multi-agents. . . . .	36
2.7 Communication par Partage d'Informations . . . . .	38
2.8 Processus de l'approche proposée. . . . .	41
3.1 Architecture proposée. . . . .	45
3.2 Architecture de l'agent Requestor. . . . .	46
3.3 Architecture de l'agent Broker. . . . .	46
3.4 Architecture de l'agent Discovery. . . . .	47
3.5 Architecture de l'agent Ranking. . . . .	48
3.6 Diagramme de séquence de la phase de publication des services . . . . .	50



---

3.7 Diagramme de séquence du fonctionnement générale de l'appli- cation. . . . .	51
4.1 Environnement logiciel utilisé. . . . .	55
4.2 Interface principale de eclipse. . . . .	57
4.3 Logo de Jade . . . . .	57
4.4 Interface graphique de la plateforme JADE. . . . .	58
4.5 Logo MySQL . . . . .	59
4.6 Logo PhpMyAdmin. . . . .	60
4.7 Logo XAMPP . . . . .	60
4.8 Table Base de donnée1. . . . .	61
4.9 Visualisation des échanges de messages entre les agents du système. . . . .	62
4.10 Interface principale de l'application. . . . .	63
4.11 interface d'inscription du fournisseur. . . . .	64
4.12 interface d'inscription du client. . . . .	64
4.13 Interface Login. . . . .	65
4.14 Interface Client. . . . .	66
4.15 Interface d'affichage des services similaires (Service SMS). . . . .	66
4.16 Interface d'affichage des services similaires (Service MSG). . . . .	67
4.17 Interface d'affichage des services similaires (Service MAIL). . . . .	67
4.18 Interface recherche d'un service. . . . .	68
4.19 Résultat calculé par TOPSIS. . . . .	68
4.20 Interface Fournisseur. . . . .	69
4.21 Interface ajout d'un nouveau service. . . . .	69
4.22 Interface affichage des services. . . . .	70
4.23 Interface Recherche d'un nouveau service. . . . .	70

## Résumé

Avec la croissance importante des nombres services sur le net, la sélection d'un service web parmi plusieurs services similaires est devenue une tâche très délicate pour l'utilisateur de services web.

Une sélection doit être effectuée afin de déterminer quels services Web pertinents répondraient aux besoins d'un utilisateur. La qualité des services Web qui se mesure e à l'aide de plusieurs métriques peut répondre au mieux à cette préoccupation.

C'est par l'apparition des systèmes multi-agent que la dimension coopérative à des entités pour la réalisation de tâches complexe fut exploitée avec succès.

Dans ce travail, nous proposons une approche basée agent qui prend en charge les critères non fonctionnels de qualité de services (QoS) dans le processus de découverte et de sélection des services web. L'introduction de l'aspect agent dans notre architecture est dans le but de doter les services web des capacités d'agents.

Afin de montrer la faisabilité de notre approche, nous avons développé un prototype de sélection des services web.

**Mots clés :** Service Web, Qualité de service (QoS), SOA, Sélection des services, AHP, TOPSIS, Agent, Java.

---

## Abstract

With the significant progress of the number of services on the net, the selection of a web service among several similar services has become a very delicate task for the user of web services.

A selection should be made to determine which relevant web services would respond the needs of a user. The quality of web services that can be measured using multiple metrics can best address this concern.

It was through the emergence of multi-agent systems that the cooperative dimension to entities for performing complex tasks was successfully exploited.

In this work, we propose an agent-based approach that supports non-functional quality of service (QoS) criteria in the process of discovering and selecting web services. The introduction of the agent aspect in our architecture is intended to provide web services with agent capabilities.

In order to show the feasibility of our approach, we have developed a prototype web services selection.

**Keywords :** Web Service, Quality of service (QoS), SOA, Selection of service, AHP, TOPSIS, Agent, Java.

## Introduction Générale

### I. Contexte :

Le Web a évolué pour englober diverses sources d'information accessible mondialement. Les organisations de tous les spectres ont déjà déplacé leurs opérations principales au Web, ce qui a entraîné une croissance rapide des différentes applications Web et en particulier les services web. Ces derniers ont l'avantage de faciliter l'intégration d'applications ou de systèmes d'information.

Les organisations commerciales exploitent actuellement de plus en plus le web pour l'adaptation aux marchés à travers les applications B2B (intégration de business-to-Business) et les applications B2C (Business-to-consumer).

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.

## II. Problématique et objectifs :

Le nombre de services Web offrant la même fonctionnalité sur le Web est en croissance sans cesse, de sorte que la tâche de sélection de services Web parmi plusieurs services similaires est devenue une tâche très délicate pour le consommateur du service.

La capacité d'accéder efficacement aux services Web est nécessaire. En outre, comme les services Web avec des fonctionnalités similaires sont censés être fournis par des fournisseurs concurrents, le défi majeur est de concevoir des stratégies d'optimisation pour trouver les "meilleurs" services Web .

L'objectif du projet consiste à exploiter le paradigme agent et les méthodes MCDM (AHP,TOPSIS) pour sélectionner le meilleur service.

## III. Solution :

On propose dans ce travail un modèle basé agents pour la sélection des services web prise en compte de la qualité de service QoS de telle façon la phase de la recherche des services similaire serait donc simple et facile et pour faire l'évaluation de ses critères qui focalise sur l'utilisation de deux algorithmes le premier qui est l'algorithme AHP pour calculer le poids des critères et le second est l'algorithme TOPSIS pour classer les services web a base des critères de qualité QoS, en fonction des exigences non-fonctionnelles. Nous nous inspirons cette idée après l'étude de quelques travaux qui utilisent les méthodes qui aident à la décision multicritères comme une axe principale pour la résolution de certains problème dans le domaine des services Web pour avoir une sélection des meilleurs services Web parmi les services similaires.

## IV. Plan du mémoire :

Le travail développé dans le cadre de ce projet est présenté en quatre chapitres et une conclusion générale, qui sont organisés comme suit :

- **Le premier chapitre « Services Web »** : Il présente la technologie des services Web et les principaux standards qu'elle supporte, parmi les quels on cite les protocoles soap, wsdl, uddi, il montre aussi leurs avantages et leurs inconvénient.
- **Le deuxième chapitre « Sélection des services Web »** : Est consacré à la sélection et aux Systèmes multi agents, et Concerne notre propre cas d'application dont le cadre de la sélection des services web en basant sur les deux algorithmes AHP et TOPSIS.
- **Le troisième chapitre « Conception du système »** : Nous commencerons notre première contribution sous la forme d'une conception de l'architecture générale de notre système basé agent, nous présenterons aussi la descriptions détaillée de ses composants.
- **Le quatrième chapitre « Implémentation du système »** : Dans ce dernier chapitre, nous allons présenter l'environnement logiciel et matériel sur lequel le système sera réalisé et validé, les détails d'implémentation de notre application, propose une mise en œuvre pratique et une présentation des résultats expérimentaux.

Ces quatre chapitres sont clôturés par une conclusion générale qui résume les résultats de notre travail, et présente les perspectives que nous souhaitons réaliser dans le futur.

Chapitre

1

# **Service Web**

## 1.1 Introduction

De nos jours, les entreprises et les systèmes d'information expriment un grand besoin pour échanger des informations et des services. Ceci nécessite des langages communs de communication. Les efforts de standardisation de ces langages ont donné lieu à un nouveau domaine de recherche connu sous le nom de « protocoles B2B ». Une technologie émergente dans ce domaine a permis de tracer quelques pistes intéressantes pour la communication entre entreprises. Cette technologie est celle de services Web. Les services Web sont un paradigme naissant qui vise à la transposition des architectures par composant dans le cadre du Web. Un service Web est un composant logiciel qui offre des services à travers une interface standardisé. La particularité des services Web est l'utilisation de la technologie Internet comme infrastructure pour la communication entre eux.

Les fonctionnalités offertes par les services Web sont relativement simples, alors que les besoins des utilisateurs sont de plus en plus complexes, à tel point qu'un seul service Web ne peut pas les satisfaire. L'implication de la composition des services Web, dans le but de créer un nouveau service Web dit composite ainsi que la sélection automatique de ces dernières offres des nouvelles perspectives qui répondent aux exigences complexes des utilisateurs.

## 1.2 Architecture Orientée Service (SOA)

Le terme *SOA*<sup>1</sup> est dans le domaine de la technologie de l'information au cours des dernières années est devenu un sujet d'hypo. SOA n'est pas une nouvelle technologie mais beaucoup plus un concept à distribué IT-

---

1. Service Oriented Architecture



Services impliquer dans des applications commerciales. L'idée centrale des architectures orientées services réside dans la perte de différents systèmes sur les services coupler entre eux et donc de nouvelles applications au moyen pour développer la composition[12].

### 1.2.1 Définition de SOA

Une architecture orienté service est une architecture logicielle s'appuyant sur un ensemble de services simples. Ils sont développés en s'inspirant des processus métier de l'entreprise. Un service est un composant fonctionnant de manière autonome et offrant des fonctionnalités métiers à d'autres applications ou d'autres services. Ces services représentent les fonctions basiques des fonctionnalités des entreprises. Ils dialoguent entre eux au travers de bus ou par Internet, on parle alors de Webservice (WSOA). Les échanges peuvent se faire de manière synchrone ou asynchrone. L'entreprise s'enrichit de services mutualisables permettant de répondre rapidement et avec souplesse aux demandes du marché. En effet, ils correspondent à un processus métier mutualisable au niveau de l'entreprise. Cela permet les changements au niveau informatique des décisions stratégiques et tactiques de l'entreprise[13].

Une architecture orientée services (SOA) est une architecture logicielle, basé sur les concepts clés suivants : Application frontend, service, référentiel de service et serviceBus. Un service consiste un contrat, un ou plusieurs Interfaces et implémentation[12].

### 1.2.2 Objectifs

L'architecture SOA vise trois objectifs importants :

- Identification des composants fonctionnels.

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

### 1.2.3 Avantages d'une architecture orientée service :

Une architecture orientée services permet d'obtenir tous les avantages d'une architecture client-serveur et notamment [14] :

- Une modularité permettant de remplacer facilement un composant (service) par un autre.
- Une réutilisabilité possible des composants (par opposition à une système tout-en-un fait sur mesure pour une organisation).
- De meilleures possibilités d'évolution (il suffit de faire évoluer un service ou d'ajouter un nouveau service).
- Une plus grande tolérance aux pannes..
- Une maintenance facilitée.

## 1.3 Services web

Le sujet des services web est un sujet ardu et rebutant, ils constituent une concrétisation des principes du SOA (Architecture orientée service), la plus utilisée. En plus de ces derniers, les services Web présentent plusieurs avantages tels que l'adaptation aux caractéristiques du Web, et le bénéfice d'un large support de la part des industriels. Nous allons voir dans ce chapitre la définition des services Web, les architectures associée.

### 1.3.1 Définition des services web

Un service web est un programme informatique permettant la communication et l'échange de données entre applications et systèmes hétéro-

gènes dans des environnements distribués. Il s'agit donc d'un ensemble de fonctionnalités exposées sur internet ou sur un intranet, par et pour des applications ou machines, sans intervention humaine, et en temps réel.

Le groupe de World Wide Web Consortium (W3C ) dans le cadre du service web a émis la définition suivante : « Un service web est un système logiciel conçu pour permettre l'interopérabilité de machine-à-machine sur un réseau. Dispose d'une interface décrite dans un format exploitable par machine (en particulier WSDL). D'autres systèmes exerce une interaction avec le service web d'une manière prescrite par sa description en utilisant des messages SOAP, typiquement transmis avec le protocole HTTP avec une sérialisation XML , en conjonction avec d'autres standards relatifs au web»[1].

Les différentes définitions proposées s'accordent au moins sur l'idée qu'un service Web est un nouveau type de composant logiciel ayant la capacité de publier ses fonctions sur Internet sous forme de services, de rendre ces services facilement invocables et de les mettre à disposition par l'intermédiaire de protocoles Internet standardisés (HTTP, XML) [2].

### **1.3.2 Architecture des services web**

L'exposition et l'utilisation des services se fait dans un contexte particulier qui définit clairement les interactions entre le service et ses utilisateurs. Les services Web communiquent via un ensemble de technologies fondamentales qui partagent une architecture commune. Ils ont été conçus pour être réalisés sur de nombreux systèmes développés et déployés de façon indépendante. Les technologies utilisées par les services Web sont HTTP, WSDL, XML-RPC, SOAP et UDDI. 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 (Annuaire UDDI) : 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[3].

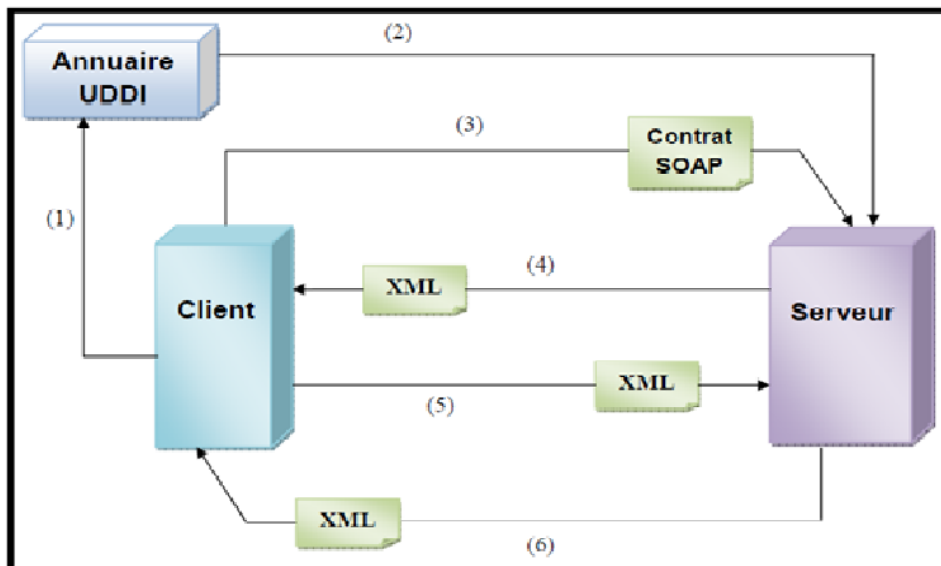


FIGURE 1.1 – Structure des services web[15].

1. Le client envoie une requête à l'annuaire de Service pour trouver le service Web dont il a besoin.
2. L'annuaire a trouvé le service approprié, il envoie l'information du serveur qui l'héberge.
3. Le client demande quel est le contrat du service web que tu proposes ?
4. Le serveur envoie sa réponse sous la forme établie par WSDL en langage XML.

5. Le client appelle le service web sous la forme établie par SOAP en langage XML.
6. Le serveur envoie le résultat du service web sous la même forme normalisée.

L'architecture standard d'un service Web est organisée en plusieurs couches. Chacune d'elles répond à des préoccupations fonctionnelles différentes telles que la publication, la description, la messagerie et le transport. Comme illustré sur la Figure I.2. Les différentes couches de l'architecture d'un service Web s'interfaçent avec des standards, comme suit[4] :

- **La couche Business Process** : Cette couche supérieure permet l'intégration de services Web, elle établit la représentation d'un « Business-Process » comme un ensemble de service Web. De plus, la description de l'utilisation de différents services composant ce service est disponible par l'intermédiaire de cette couche.
- **La couche de publication** : repose sur l'annuaire UDDI<sup>[2]</sup>, qui assure le regroupement, le stockage et la diffusion des descriptions des services Web.

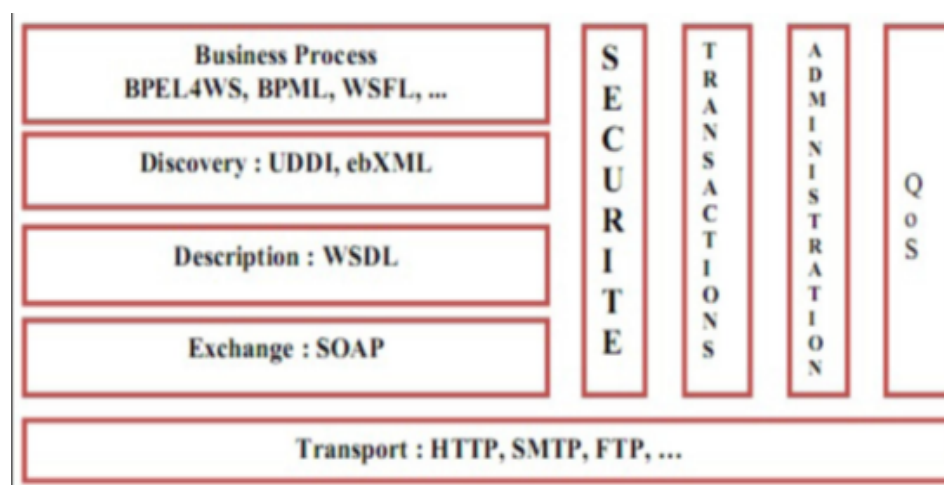


FIGURE 1.2 – Architecture en Pile des servicesWeb[4].

- **La couche description** est prise en charge par le langage WSDL (Web Service Description Language) , qui décrit les fonctionnalités fournies par le service Web, les messages reçus et envoyés pour chaque fonctionnalité, ainsi que le protocole utilisé pour la communication.
- **La couche communication** : la couche de communication des messages propose différents mécanismes liés à l'acheminement des messages (format de communication des messages, adressage, routage...etc.). Cette couche utilise des protocoles reposants sur le langage XML<sup>3</sup>, car sa syntaxe unique résout les conflits syntaxiques lors de l'encodage des données. Actuellement, SOAP<sup>4</sup> est le protocole le plus utilisé pour cette couche.
- **La couche transport** le protocole le plus utilisé dans cette couche est l'HTTP<sup>5</sup>. Cependant, d'autres protocoles peuvent être utilisés, tels que le SMTP<sup>6</sup> ou le FTP<sup>7</sup>, permettant ainsi aux services Web de rester indépendants du mode de transport utilisé.
- **Les couches transversales** Security, Transactions, Administration, QoS, Ce sont des couches qui rendent viable l'utilisation effective des services Web dans le monde industriel.

### 1.3.3 Standards de base de services Web

Une caractéristique qui a permis un grand succès de la technologie des services web est qu'elle est construite sur des technologies standards, nous présentons dans cette section les trois standards de base, à savoir, le protocole SOAP, le langage WSDL et l'annuaire UDDI. Ces trois standards sont basés sur le langage XML.

---

3. Extensible Markup Language  
4. Simple Object Access Protocol  
5. Hyper Text Transfer Protocol  
6. Simple Mail Transfer Protocol  
7. File Transfer Protocol

### 1.3.3.1 SOAP

**SOAP (Simple Object Access Protocol) :** SOAP est un protocole léger bâti sur XML. Il permet la transmission des informations structurées (message) entre objet distant dans un environnement distribué et décentralisé (SOA). Le transfert se fait à l'aide des protocoles HTTP et SMTP. Le protocole SOAP est particulièrement utile pour exécuter des dialogues requête-réponse RPC (RemoteProcedure Call).

SOAP [5] est un protocole adopté par le Consortium W3C. Le Consortium W3C crée des standards pour le Web : son but est donc de créer des standards pour favoriser l'échange d'information.

SOAP veut dire Simple Object Access Protocol. Si l'on voulait traduire cette définition en français cela donnerait Protocole Simple d'Accès aux Objets. En effet, le protocole SOAP consiste à faire circuler du XML généralement, via du http sur le port 80 (ou en utilisant un autre protocole). Cela facilite grandement les communications, car le XML est un langage standard et le port utilisé est le port 80, qui ne pose donc pas de problème pour les firewalls (pare-feu) de l'entreprise, contrairement à d'autres protocoles. Tout comme la technologie des services Web, le protocole SAOP est très jeune. Le protocole SOAP a été créé en septembre 98, avec la version 0.9, par trois grand entreprise[5].

#### **Ses caractéristiques**

- SOAP permet une normalisation des échanges de données. Les données sont encodées en XML et échangées par des appels de procédures à distance [6] en utilisant HTTP/SMTP/POP comme protocole de communication.
- Simple, extensible et permet le diagnostic des erreurs.
- Message unidirectionnel (Requête-Réponse).

- Fonctionne de manière synchrone et asynchrone.
- Indépendant de la plate-forme et du langage.
- N'est pas perturbé par un pare-feu.

### Fonctionnement

La requête SOAP transmise par le client à l'aide des protocoles de trans-

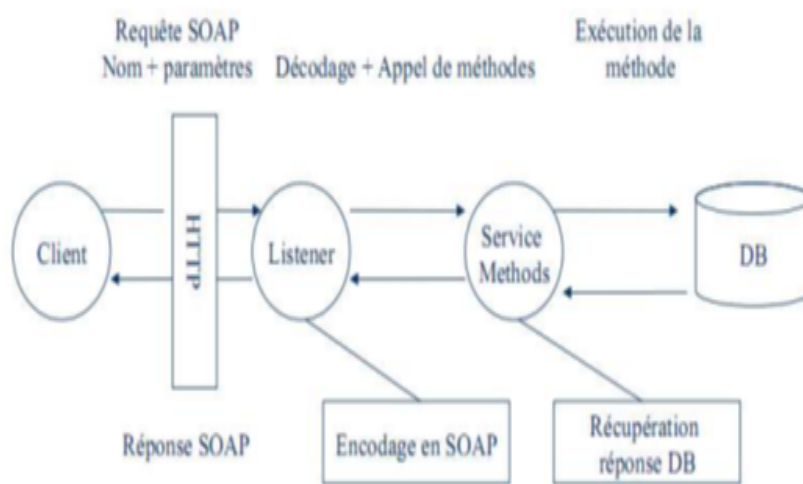


FIGURE 1.3 – Traitement d'un message SOAP[2]..

missions (exemple HTTP) passe par un Listener() qui permet de adresser (cas de java) aux bibliothèques, ces bibliothèques données au client l'accède aux méthodes des services. Si l'exécution des méthodes a besoin de informations de la base de données alors il doit connecter avec la base de données.

Tout d'abord un message SOAP est un document XML qui doit avoir la forme suivante [6] :

- La structure de l'enveloppe SOAP qui définit une structure générale décrivant le contenu, le destinataire, et la nature du message
- Les règles d'encodage SOAP qui définissent le mécanisme de sérialisation utilisé pour échanger des objets



- SOAP RPC qui définit, pour les utilisations synchrones, une convention de représentation des appels et des réponses des procédures distantes.

### Les composant d'un message SOAP

Un message SOAP est composé des parties suivantes comme la montre la figure 1.4



FIGURE 1.4 – Structure de message SOAP[6].

- **Le HTTP Header :**

Le protocole HTTP envoie une requête POST. L'entête HTTP se trouve juste avant le message SOAP, et définit le destinataire du message, les règles d'encodage HTTP, etc. Le champ SOAP Action peut être utilisé pour indiquer l'intention de la requête SOAP. Cette information peut être utilisée par un firewall pour filtrer les messages. Ce champ est obligatoire mais peut être vide si on n'indique pas l'intention de la requête.

- **L'enveloppe SOAP :**

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

- **Le header SOAP :**

Est optionnelle, extensible. Les balises XML qui permettent de sym-

boliser cette partie sont : `<env :Header>` et `</env :Header>` qui peuvent être complétées par des attributs permettant de définir le domaine de noms du service Web. En fait, l'en-tête permet principalement d'ajouter des informations sur le comportement des différents nœuds intermédiaires, lors de traitement du message.

- **Le Body SOAP :**

L'élément corps de message SOAP contient des données spécifiques à l'application. Les balises symbolisant cette partie sont : `<env :Body>` et `</env :Body>`. Les données doivent donc être sérialisées selon l'encodage XML. En plus, des données de cette partie peuvent transporter un type spécial comme : les messages d'erreurs.

### 1.3.3.2 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.

### 1.3.4 La technologie WSDL

WSDL <sup>8</sup>[2] a été développé conjointement par IBM, Microsoft et Ariba et a été présenté pour analyse au W3C qui l'a accepté comme une notice et publié sur leur site. L'utilité de WSDL est de décrire et publier le format et les protocoles d'un Web service de manière homogène par l'utilisation du format XML. Cela permettra au requérant et à l'émetteur d'un service de comprendre les données qui seront échangées.

- **Structure et description WSDL :** WSDL est divisé en trois éléments majeurs pouvant être séparés et utilisés indépendamment ou être combinés pour former un seul document XML. Ces éléments se subdivisent

---

8. Web Service Description Language

en 7 types de composantes descriptives dans la définition des services réseaux. Ces éléments sont montrés dans la figure 1.5.

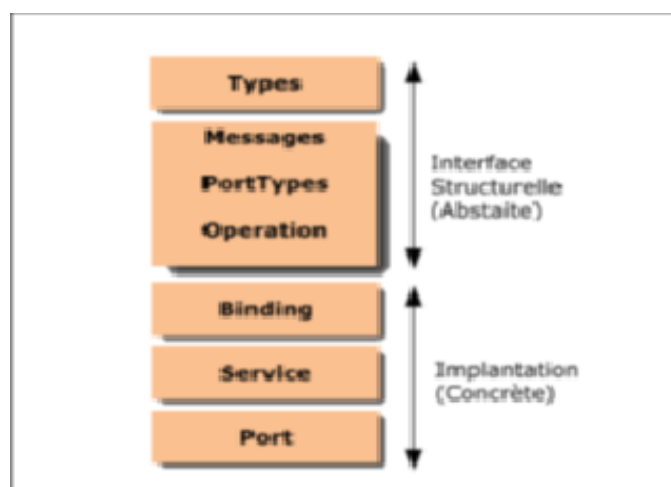


FIGURE 1.5 – Structure et description WSDL[7].

1. **Data Type Définition** : Identification du contenu et du type de données qui sont dans le message.
2. **Data types** : Une enveloppe pour les « data type définitions » utilisant des systèmes Tel que XSD (XML Schéma Définition)
3. **Abstract Operations** : Définition de la manière dont les données seront échangées.
4. **Operation** La description abstraite d'une action supportée par le service.
5. **Port Types** : Un ensemble d'opérations supporté par un ou plusieurs point d'accès (port).
6. **Binding** : Un protocole spécifique et une spécification du format de données pour un point d'accès particulier (port).
7. **Service Binding** : Définition de la couche transport qui servira au message.
8. **Port** : Un point d'accès unique définit comme une combinaison de l'adresse Réseau et du point d'accès particulier (port).

9. **Service** : Un ensemble de terminaisons reliées.

### 1.3.5 UDDI - Universal Description Discovery and Integration

UDDI est une spécification définissant la manière de publier et de découvrir les services Web sur un réseau [8]. 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. UDDI peut être vu comme un annuaire tel que présenter dans la figure 1.6

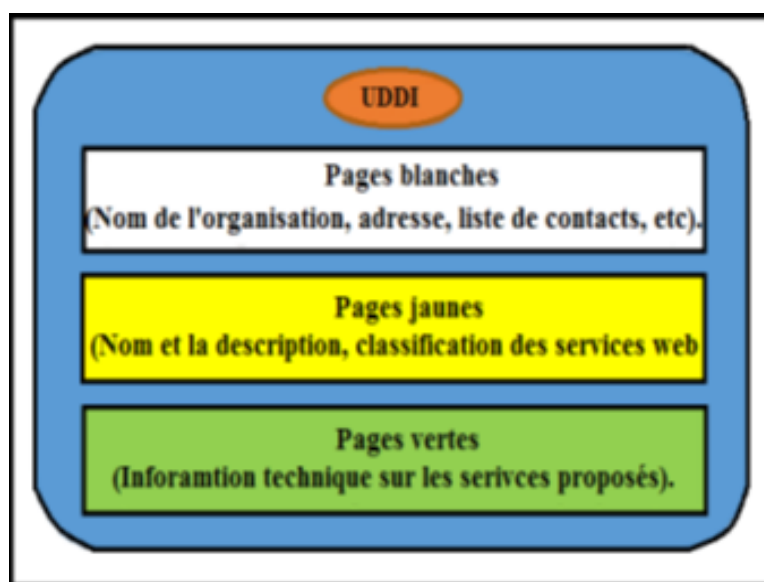


FIGURE 1.6 – Le contenu de l'annuaire UDDI[4].

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

- 1- Les pages blanches** : Contiennent le nom, l'adresse de l'entreprise et l'information de contact.
- 2- Les pages jaunes** : contiennent les données du genre : type d'activité commerciale, emplacement géographique, types de produits offerts et type d'industrie.
- 3- Les pages vertes** : contiennent des informations techniques du service offert, la manière d'interagir avec le service, une définition du

business process et aussi un pointeur vers le fichier WSDL et une clé unique identifiant le service [9].

Un annuaire UDDI est conçu pour être interrogé par des messages SOAP, afin de pouvoir stocker et fournir des informations permettant l'accès aux documents WSDL (Web Services Description Language) décrivant les protocoles et les formats de messages nécessaires pour interagir avec les services Web répertoriés dans l'annuaire. Les outils de recherche disponibles sont basés sur des mots-clés et ne prennent pas en considération les relations entre les services Web et les caractéristiques sémantiques de chaque service Web, forçant l'utilisateur à reformuler sa requête en utilisant de nouveaux termes clés [4].

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 répliqués de DNS). Une fois ceci est fait, le service Web peut alors être connu de tous ceux qui le recherchent [10].

### 1.3.6 Types de structures de données d'un modèle UDDI

Le modèle UDDI comporte 5 types de structures de données décrites sous forme de schéma XML.

**1- 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.

**2- Business service :** Ensemble d'informations sur les services publiés par l'entreprise (nom du service, les catégories. . .).

**3- Binding Template :** Ensemble d'informations sur le lieu d'hébergement du service (c.à.d. l'adresse du point d'accès du service).

**4-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.

**5-Publisher Assertion :** Ensemble d'informations contractuelles entre les partenaires.

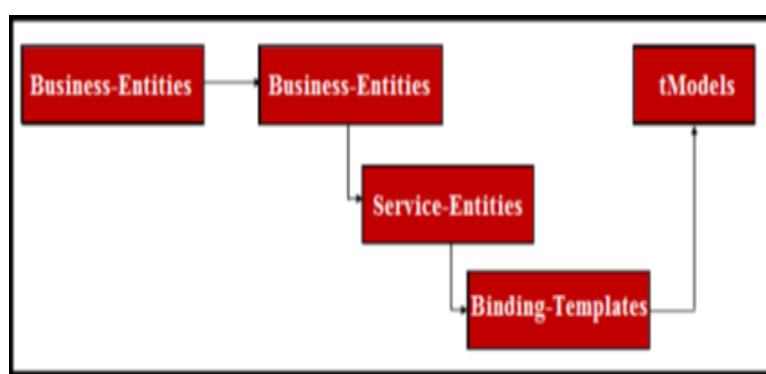


FIGURE 1.7 – Les structures de données d'UDDI.

### 1.3.7 Types d'annuaires UDDI

Cette section s'articule autour de deux types importants d'annuaires UDDI (publiques et privés), nous allons les discuter par la suite.

- **L'annuaire UDDI public :** Il est implémenté sous forme d'un réseau de noeuds UDDI, ces noeuds sont synchronisés, chacun d'eux est possédé par une entreprise donnée (telles que SAP (Systems, Applications and Products for data processing), IBM et Microsoft). La publication d'un service chez une entreprise Propage automatiquement ses informations (pages blanches, jaunes et vertes) aux différents noeuds UDDI. L'accès à l'ensemble des informations des registres peut se faire à n'importe quel noeud UDDI. Ce type d'annuaire est gratuit.

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

### 1.3.8 Les avantages et inconvénient des services Web

#### 1.3.8.1 Les avantages

- Les services Web fournissent l'interopérabilité entre divers logiciels fonctionnant sur diverses plates-formes.
- Les services Web utilisent des standards et protocoles ouverts.
- Les protocoles et les formats de données sont au format texte dans la mesure du possible, facilitant ainsi la compréhension du fonctionnement global des échanges.
- Basés sur le protocole HTTP, les services Web peuvent fonctionner au travers de nombreux pare-feux sans nécessiter des changements sur les règles de filtrage.
- Les outils de développement, s'appuyant sur ces standards, permettent la création automatique de programmes utilisant les services Web existants. [10]

#### 1.3.8.2 Les inconvénients

- Les normes de services Web dans certains domaines sont actuellement récentes.
- Les services Web ont de faibles performances par rapport à d'autres approches de l'informatique répartie telles que le RMI, CORBA, ou DCOM.
- En l'utilisation du protocole HTTP, les services Web peuvent contourner les mesures de sécurité mises en place à travers les firewalls. [11]

## 1.4 Conclusion

Nous avons présenté dans ce chapitre, une vue d'ensemble des services web tout en relatant leurs technologies et leur fonctionnement. Ces technologies améliorent l'intégration et l'interopérabilité entre services à travers l'infrastructure Web en utilisant différents standards XML : SOAP pour l'échange de messages, WSDL pour la description de services et UDDI pour la publication et la découverte de services Web. Nous avons présenté aussi le principe de l'architecture SOA.

Dans le prochain chapitre nous allons présenter les approches de sélection des services web qui se basent sur des critères de qualité de service (fonctionnels et non fonctionnels) et la notion d'agents aussi les différentes méthodes multicritères qui aide à la décision (MCDM). Puis, nous détaillerons deux algorithmes de sélection multi critères AHP (Analytic Hierarchy Process) et TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution) que nous allons utiliser dans notre système.



Chapitre

2

# **SMA et Sélection des Services Web**

## 2.1 Introduction

La sélection des services Web a attiré de plus en plus d'attention. Le service qui satisfait un agent parmi un ensemble de services doit être sélectionné sur les préférences de l'agent sur les attributs de qualité de service (QoS) non fonctionnels des services. Il est toutefois souhaitable d'évaluer les préférences dans une manière qualitative plutôt que quantitative. En outre, une décision doit être prise pour un groupe d'agents avec différentes préférences sur les attributs (QoS) des services Web[16]. Dans ce travail nous proposons une approche de sélection des services web. Nous utilisons un modèle à base de poids pour un ensemble de paramètres de QoS (Qualité de Service), pour la sélection du meilleur service web.

## 2.2 Qualité de service (QoS)

### 2.2.1 Définition

La qualité de service d'un service Web est définie comme étant une combinaison de plusieurs critères qui peuvent être quantitatifs tels que le temps de réponse, la disponibilité, ... ou qualitatifs tels que la sécurité, la disponibilité, l'accessibilité, la fiabilité,..., etc. Ces paramètres peuvent être alors considérés comme un critère de choix lorsqu'on à sélectionner parmi plusieurs services web découverts ceux qui respectent les contraintes imposées[17].

Chaque service est identifié par une interface et cette interface comporte des propriétés fonctionnelles et éventuellement des propriétés non-fonctionnelles permettant à un consommateur de trouver le service qui convient à ses besoins[17].

### 2.2.2 Critères de qualité de service

Dans ce qui suit nous nous focalisons sur les propriétés et les critères les plus représentatives de QoS :

- 1) **Disponibilité (Availability)** : Représente le pourcentage de requêtes réussites par le fournisseur. Les réponses échouées correspondent aux exceptions reçues du côté client. La disponibilité est aussi la probabilité que le service est accessible[21].
- 2) **Fiabilité (Reliability)** : Ce paramètre désigne la capacité du service à accomplir sa fonction correctement pendant une période de temps spécifiée. Il peut être mesuré par le temps moyen entre pannes[19].
- 3) **Temps de réponse (Response time)** : C'est la durée pour traiter une requête dès l'instant de son envoi jusqu'au moment qui reçoit une réponse[17].
- 4) **Débit (Throughput)** : C'est le nombre de demandes de service complétées sur une période de temps [22].
- 5) **Compatibilité (Interoperability)** : La compatibilité d'un service Web pour intégrer un ensemble de normes données.
- 6) **Accessibilité (Accessibility)** : L'accessibilité est l'aspect qualité d'un service qui représente le degré auquel il est capable de répondre à une demande de service Web [22].
- 7) **Prix d'exécution (Execution price)** : Le prix d'exécution est la somme d'argent qu'un demandeur de service doit payer pour exécuter une opération d'un service [23].
- 8) **Réputation** : C'est une mesure de la crédibilité du service. Elle dépend principalement des expériences d'utilisateurs finals.

## 2.3 La sélection des Services Web

Le domaine informatique a beaucoup évolué depuis ces dernières années. Le nombre de services publiés sur Internet a grandi à une vitesse explosive, il est difficile pour les demandeurs de services de sélectionner des services Web satisfaisants, qui fournissent des fonctionnalités similaires. La Qualité de service (QoS) est considérée comme le plus important critère non fonctionnel pour le filtrage de service supplémentaire.

Dans cette phase, le contexte est intégré entre l'utilisateur et l'annuaire. Lors de la recherche des Services Web élémentaires, certains travaux intègrent le contexte d'utilisation afin de trouver les Services Web les mieux adaptés à la demande de l'utilisateur. Le système peut avoir à sa disposition un ensemble de Services Web qui peuvent répondre de manière fonctionnelle à la demande. La technique de sélection mise alors en jeu qui consiste à sélectionner le service web qui correspond le mieux au contexte.

De manière simplifiée, le processus de sélection de services Web repose sur trois étapes : l'expression de la requête de l'utilisateur, la recherche de services et la validation du résultat de la recherche par l'utilisateur[18].

- **L'expression de la requête** : L'utilisateur lance sa requête. Afin de permettre à l'utilisateur de découvrir les Services Web pertinents par rapport à son contexte.
- **La recherche de Services Web** : La recherche est effectuée par un processus de correspondance entre les différents critères de qualité et les services représentant la requête.
- **La validation de la recherche par l'utilisateur** : Le résultat de l'étape précédente est retourné à l'utilisateur qui déclare s'il est satisfait de ce

résultat. Si l'utilisateur n'est pas satisfait, le processus de recherche est renouvelé.

### 2.3.1 Critères pour les méthodes de sélection des services

La sélection de service Web utilise des approches qui sont étudiées par rapport à un ensemble de caractéristiques[20] ,ils sont décrites comme suit :

- **Qualité de service** : Ceci se rapporte à des approches qui tiennent compte de la qualité de service que le critère de prise de décision. Les QoS sont répandues Durée (temps d'exécution), la disponibilité, fiabilité et de coût. Ceci est un sujet de la plus haute importance, qui doit être pris en considération[20].
- **La préférence de l'utilisateur** : Ceci fait référence aux approches qui traitent de préférences de l'utilisateur a fin de rendre compte de la priorité des consommateurs de services. Par exemple, l'importance relative des critères dans une matrice de décision peut être obtenue par la préférence du demandeur de service[20].
- **Automatique** : L'élément essentiel dans la sélection automatique de service réside dans l'étape finale . Quand un service est disponible , le concepteur de service spécifie les données pour le service et l'utilisateur spécifie les exigences .Cependant, l'intervention humaine est absente lors de la sélection de service[20].
- **Evolutivité** : Ceci se rapporte à des approches qui tiennent compte de nombreuses propriétés et les processus de classement qui se produisent en même temps, tout en conservant la précision des résultats. Dans certains procédés, la précision du procédé est influencée par le nombre de services de remplacement ou l'augmentation du nombre de critères[20].

## 2.4 Méthodes d'aides à la décision multicritère (MCDM)

Les méthodes de classification multicritère font partie du domaine d'aide multicritère à la décision. Elles consistent à élaborer des procédures d'affectation qui permettent d'affecter chaque individu à une classe prédéfinie et ceci à travers l'examen de sa valeur intrinsèque en se référant à des normes préétablies. Ces méthodes utilisent uniquement des comparaisons entre l'individu à affecter et les individus de référence par le biais d'un modèle de préférence[24].

Le but des méthodes en MCDM est donc moins de proposer automatiquement une unique solution considérée comme optimale, que d'accompagner le décideur dans sa démarche de décision. La phase de définition du problème, des alternatives et des critères est généralement considérée comme plus importante que la phase purement calculatoire mettant en œuvre une méthode particulière d'agrégation de préférence. En particulier, l'interaction avec le décideur tout au long du processus fait que le résultat d'une méthode d'aide à la décision ne semble pas arriver de nulle part, mais permet bien au décideur de prendre sa décision de manière éclairée[25].

### 2.4.1 Description de quelques méthodes MCDM

Il existe trois étapes dans l'utilisation de toute technique de prise de décision impliquant numérique analyse des alternatives [26] :

- 1) Détermination des critères et des solutions de rechange pertinents.
- 2) Joindre des mesures numériques à l'importance relative des critères et à l'impact des alternatives sur ces critères.

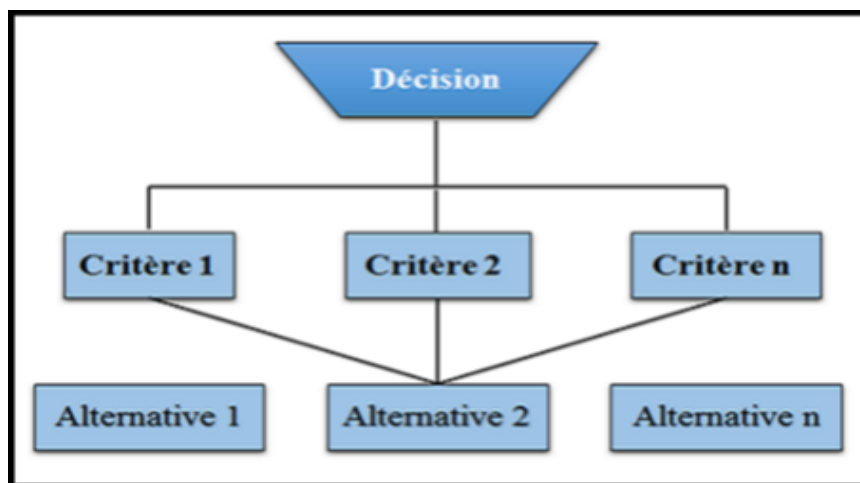


FIGURE 2.1 – Prise de décision multicritères (MCDM) [26].

3) Traiter les valeurs numériques pour déterminer un classement de chaque alternative.

Le modèle de somme pondérée (WSM : Weighted Sum Model) est la méthode la plus ancienne et probablement la plus utilisée. Le modèle de produit pondéré (WPM : Weighted Product Model) peut être considéré comme une modification du WSM, et a été proposé afin de surmonter une partie de sa faiblesse. Le processus de hiérarchie analytique (AHP : Analytic Hierarchy Process).

Récemment, la modification du AHP est considérée comme plus cohérente que l'approche originale. D'autres méthodes largement utilisées sont les méthodes ELECTRE et TOPSIS.

#### 2.4.1.1 La Méthode ELECTRE

La méthode ELECTRE introduit en 1966. Le concept de base de la méthode ELECTRE est de traiter "Relations de surclassement" en utilisant des comparaisons par paire entre des alternatives sous chacune des critères séparément. La relation de surclassement des deux alternatives  $A_i$  et  $A_j$  décrit que même si la  $i$ -ième alternative ne détermine pas la  $j$ -ième alternative quantitativement, alors le décideur peut encore prendre le risque

de considérer  $A_i$  comme presque mieux que  $A_j$ . Les alternatives seraient dominées, s'il existe une autre alternative les excelle dans un ou plusieurs critères et égale dans les critères restants.

La méthode ELECTRE commence par des comparaisons par paires d'alternatives sous chaque critère. En utilisant des valeurs physiques ou monétaires, notées  $g_i$  ( $A_i$ ) et  $g_j$  ( $A_j$ ) des alternatives  $A_i$  et  $A_j$  respectivement, et en introduisant des seuils pour les différents  $g_i$  ( $A_i$ ) et  $g_j$  ( $A_j$ ), le décideur peut déclarer qu'il / Elle est indifférente entre les alternatives considérées, qu'il a une préférence faible ou stricte pour l'une des deux, ou qu'il est incapable d'exprimer aucune de ces relations de préférence. Par conséquent, un ensemble de relations binaires d'alternatives, les relations dites de surenchère peuvent être complètes ou incomplètes [26].

Ensuite, le décideur est prié d'attribuer des poids ou des facteurs d'importance afin d'exprimer leur importance relative. A travers les évaluations consécutives des relations de dépassement des alternatives, la méthode ELECTRE provoque l'indice de concordance défini comme la quantité de preuves à l'appui de la conclusion que l'alternative  $A_j$  dépasse ou domine, les alternatives  $A_i$ , ainsi que l'indice de discordance le « counter-part » de l'indice de concordance [26].

#### 2.4.1.2 La Méthode AHP

La méthode AHP est simple dans son principe et dans son application. Cette simplicité explique le nombre important de travaux scientifiques où la méthode AHP a été utilisée avec succès. Elle procède par combinaisons deux à deux des éléments de chaque niveau hiérarchique par rapport aux éléments du niveau supérieur [27].



La méthode AHP est utilisée avec succès dans de nombreux domaines, dont celui de la décision multicritère. La méthode commence par la définition de l'objectif principal à atteindre ou la décision à prendre, à partir de là, elle décompose cet objectif en une structure hiérarchique de critères et de sous critères d'évaluation. Dans le dernier niveau hiérarchique nous retrouvons les candidats à évaluer (les alternatifs). Parmi les avantages de la méthode AHP, sa capacité de structurer un problème complexe ; multi-critère ; permet la comparaison des éléments (alternatifs, critères et sous critères) et facile à implémenter [27]. AHP est une méthode de comparaison par paire que chaque critère se compare entre eux et obtient le score par rapport a la figure 2.2[31] :

PREFERENCE LEVEL	NUMERICAL VALUE
Equally preferred	1
Equally to moderately preferred	2
Moderately preferred	3
Moderately to strongly preferred	4
Strongly preferred	5
Strongly to very strongly preferred	6
Very strongly preferred	7
Very strongly to extremely preferred	8
Extremely preferred	9

FIGURE 2.2 – Table de préférences standard

Le critère qui a un meilleur niveau obtiendra le numéro numérique mentionné dans le tableau et l'autre obtiendra la valeur réciproque de la valeur. Pour évaluer les poids des critères, une matrice doit être créée en fonction de la définition l'échantillon de matrice pour trois critères est présenté ci-dessous[31] :

Dans ce qui suit, on décrit les étapes de l'AHP[31] :

	A	B	C
A	1	x	1/y
B	1/x	1	1/z
C	y	z	1

FIGURE 2.3 – Matrice pour évaluer les poids des critères

- **Étape 1** : résume toutes les valeurs dans chaque colonne.
- **Étape 2** : Les valeurs dans chaque colonne sont divisées par les sommes de colonne correspondantes.
- **Étape 3** : Convertir les fractions en décimales et trouver la moyenne de chaque rangée. Cette somme correspond au poids du critère de la rangée.

#### 2.4.1.3 La Méthode TOPSIS

TOPSIS (technique de préférence de commande par rapport à la solution idéale) était développé par Hwang et Yoon en 1980 comme une alternative à la méthode ELECTRE et peut être considérée comme l'une de ses variantes les plus largement acceptées. Le concept de base de cette méthode est que l'alternative choisie doit avoir la plus courte distance de l'idéal solution et la distance la plus éloignée de la solution négative-idéal dans certains géométrique sens[26].

La méthode TOPSIS suppose que chaque critère a une tendance à l'utilité monotone croissante ou décroissante. Il est donc facile de définir les solutions idéales et négatives idéales. L'approche distance euclidienne a été proposée pour évaluer la proximité relative des alternatives à la solution idéale. Ainsi, l'ordre de préférence des alternatives peut être obtenu par une série de comparaisons de ces distances relatives [26].

TOPSIS peut être appliqué à la fois pour maximiser (avantage) et minimiser (coût) les critères [32].

La méthode TOPSIS est exprimée en une succession de six étapes comme suit[32] :

- **Étape 1** : Calculez la matrice de décision normalisée. La valeur normalisée est calculé comme suit :

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{n=1}^m x_{ij}^2}}, \text{telque : } i = 1, 2, \dots, m; j = 1, 2, \dots, n. \quad (2.1)$$

- **Étape 2** : Calculez la matrice de décision normalisée pondérée. La valeur normalisée et pondérée est calculée comme suit :

$$v_{ij} = r_{ij} * w_j, \text{telque : } i = 1, 2, \dots, m; j = 1, 2, \dots, n. \quad (2.2)$$

- **Étape 3** : Déterminer les solutions idéale positive ( $A^+$ ) et négative ( $A^-$ ).

$$A^+ = \{ \max_i v_{ij} (i \in J^+) \mid \min_i v_{ij} (i \in J^-) \} = \{ v_j^* \mid j = 1, 2, \dots, m \} \quad (2.3)$$

$$A^- = \{ \min_i v_{ij} (i \in J^+) \mid \max_i v_{ij} (i \in J^-) \} = \{ v_j^- \mid j = 1, 2, \dots, m \} \quad (2.4)$$

- **Étape 4** : Calculer les mesures de séparation en utilisant la distance euclidienne. Les mesures de séparation de chaque alternative de la solution idéale positive et de la solution idéale négative, respectivement, sont les suivantes :

$$S_i^+ = \sqrt{\sum_{n=1}^m (v_{ij} - v_j^+)^2}, j = 1, 2, \dots, m. \quad (2.5)$$

$$S_i^- = \sqrt{\sum_{n=1}^m (v_{ij} - v_j^-)^2}, j = 1, 2, \dots, m. \quad (2.6)$$

- **Étape 5** : Calculer la proximité relative à la solution idéale (coefficient).  
La proximité relative de l'alternative ( $A_i$ ) par rapport à ( $A^*$ ) est définie comme suit :

$$RC_i^* = \frac{S_i^-}{S_i^+ + S_i^-}, i = 1, 2, \dots, m. \quad (2.7)$$

- **Étape 6** : Classez l'ordre de préférence.

## 2.5 Agents et Systèmes multi-agents

La notion d'agent et de système multi-agents (SMA) est relativement récente en Informatique. Le domaine des SMA permet de réaliser des entités artificielles (des programmes, des robots) qui interagissent entre elles et avec leur environnement.

### 2.5.1 La notion d'agent

Il n'y a pas une notion unique d'agent, car le terme agent est utilisé dans de différentes applications par des communautés venant d'horizons divers. Nous présentons ici les concepts les plus adoptés.

#### 2.5.1.1 Définition d'un agent

Un agent est un système informatique situé dans un environnement, capable d'actions autonomes dans cet environnement afin d'atteindre des objectifs prédéfinis.

Jacques Ferber définit l'agent comme étant : "une entité autonome physique ou virtuelle possédant des ressources propres qui est capable d'agir sur elle-même et sur son environnement, qui peut communiquer avec d'autres agents et dont le comportement est la conséquence de ses observations, de ses connaissances, et des interactions avec les autres agents" [29].

#### 2.5.1.2 Propriétés des agents

Les agents peuvent posséder plusieurs propriétés qui sont [30] :

- **Autonomie** : Un agent peut agir sans l'intervention directe d'un tiers (agent ou humain) et contrôler ses actions ainsi que son état interne.
- **Sociabilité** : Il est capable d'interagir avec d'autres agents quand la situation l'exige afin de compléter ses tâches ou aider ses agents à accomplir leurs tâches.
- **Réactivité** : Il est capable de réagir et percevoir les changements de son environnement et élaborer une réponse dans le temps requis.
- **Communication** : Un agent peut communiquer avec d'autres agents ainsi qu'avec des utilisateurs humains.
- **Aptitude sociale** : Un agent peut interagir avec d'autres agents de façon coopérative ou compétitive pour atteindre ses objectifs.
- **Pro-activité** : l'agent est capable, sur sa propre initiative, de se fixer des buts pour atteindre ses objectifs (opportuniste).

#### 2.5.1.3 Types d'agents

Selon leurs modes de fonctionnement et la représentation de leurs environnements, les agents peuvent être classés en trois catégories essentielles à savoir, les agents réactifs, cognitifs et hybrides.

- **Agent réactif** : Comme son nom l'indique, un agent réactif ne fait que réagir aux changements qui surviennent dans l'environnement. Autrement dit, un tel agent se contente simplement d'acquérir des perceptions et de réagir à celles-ci en appliquant certaines règles prédéfinies.

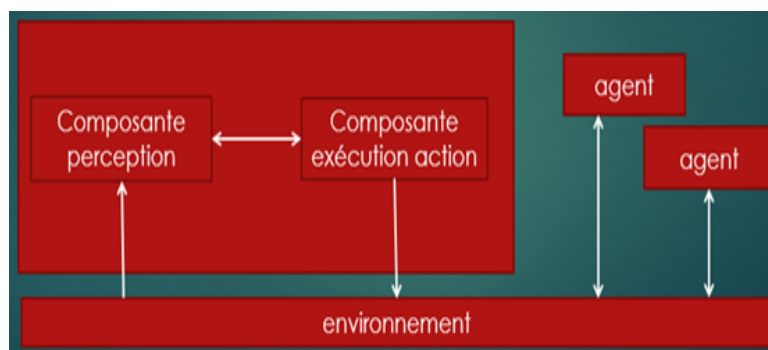


FIGURE 2.4 – Architecture d'un agent réactif.

- **Agent cognitif** : Il est intelligent par lui-même c'est-à-dire qu'il effectue un certain raisonnement pour choisir ses actions. Un tel raisonnement peut se faire soit en se basant sur les buts de l'agent, soit sur une certaine fonction d'utilité.

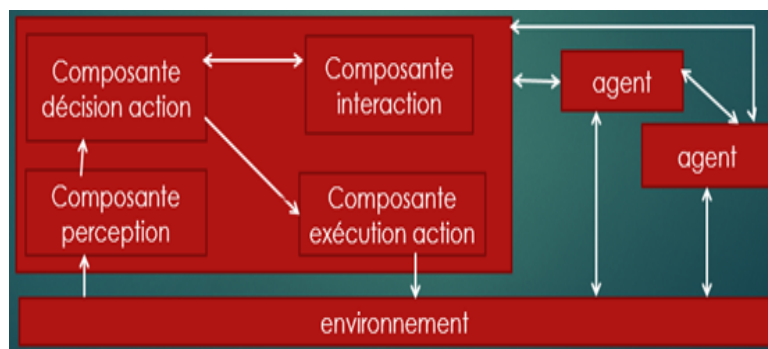


FIGURE 2.5 – Architecture d'un agent cognitif.

- **Agent hybride** : Chaque agent hybride est caractérisé par la notion de couches et chaque couche représente soit les agents cognitifs, soit les agents réactifs. Donc l'agent hybride combine entre les deux comportements (comportement réactif et comportement cognitif).

## 2.5.2 Les systèmes multi-agents(SMA)

Un système multi-agents peut être considéré comme une population d'agents autonomes en interaction partageant un environnement commun.

SMA = Agents + Environnement + Interactions + Organisations (AEIO)

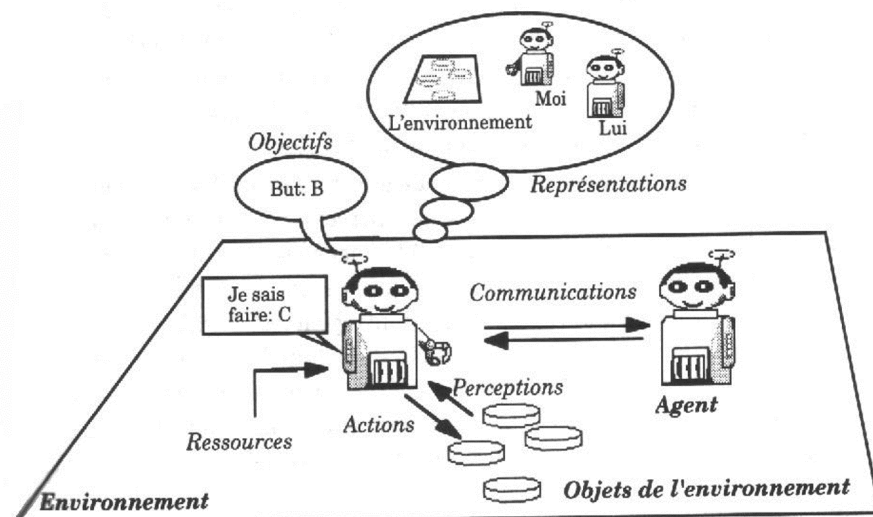


FIGURE 2.6 – Système multi-agents.

### 2.5.2.1 Définition d'un SMA

Un système multi-agents est une communauté d'agents autonomes travaillant en commun, selon des modes parfois complexes de coopération, conflit, concurrence, pour aboutir à un objectif global :

- la résolution d'un problème.
- accomplir des tâches fonctionnelles.
- simuler des systèmes existants.

### 2.5.2.2 Interactions dans un système multi-agents

Les systèmes multi-agents possèdent les avantages traditionnels de la résolution distribuée comme la modularité, la vitesse et la fiabilité.

Jacques Ferber définit l'interaction comme : « un ensemble de comportements résultant du regroupement d'agents qui doivent agir pour satisfaire leurs objectifs en tenant compte des contraintes provenant des ressources plus ou moins limitées dont ils disposent et de leurs compétences individuelles » [29].

L'interaction entre les agents apparaît sous plusieurs modes, qui sont la coopération, la coordination et la négociation.

- **La coopération** : La coopération entre les agents consiste à décomposer les tâches en sous-tâches puis à les répartir entre les différents agents, il existe plusieurs décompositions possibles, le processus de décomposition doit donc tenir compte des ressources disponibles et des compétences des agents.
- **La coordination** : la coordination d'actions est nécessaire pour trois raisons principales :
  - 1- Les agents ont besoin d'informations et de résultats fournies par d'autres agents.
  - 2- Les ressources sont limitées.
  - 3- On cherche à optimiser les coûts.
- **La négociation** : La négociation intervient lorsque des agents interagissent pour prendre des décisions communes, alors qu'ils poursuivent des buts différents. Elle est caractérisée en général par un protocole minimal d'actions qui est de proposer, d'évaluer, d'accepter ou de refuser une solution.

### 2.5.2.3 La communication dans un système multi-agents

Un agent doit être capable de communiquer avec les autres agents, Les agents doivent avoir des capacités à manipuler un langage commun. Cette



communication une forme d'action entre les agents, elle permet la mise en œuvre des interactions entre ces derniers et elle peut être sélective ou diffusée.

Il existe principalement deux modes de communication :

- **Communication par tableau noir (Partage d'informations)**

Dans les systèmes fonctionnant par partage de ressources, les différents composants ne sont pas directement liés entre eux. Ils communiquent au travers d'une zone de données commune appelée tableau noir (Blackboard), dans laquelle sont stockées les connaissances du système.

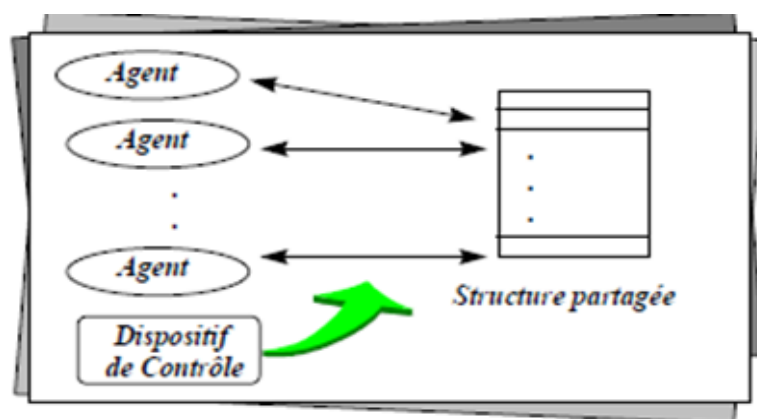


FIGURE 2.7 – Communication par Partage d'Informations

- **Communication par envoi de messages**

Dans les systèmes où la communication se fait par envoi de messages, les connaissances sont distribuées entre les différents agents. Chacun d'eux communique directement avec les autres par envoi de messages.

## 2.6 Les agents et les services web

L'utilisation de la notion d'agent pour découvrir de manière autonome, sélectionner, composer, invoquer et exécuter les services web est

ainsi un enjeu de taille pour doter les services web des capacités intéressante des agents logiciels.

Les agents étendent les services Web de plusieurs manières importantes [38] :

- Un service Web ne se connaît que lui-même, mais pas ses utilisateurs, ses clients. Les agents sont souvent agent sont conscients de soi et des autres agents et de leurs capacités à mesure que des interactions entre agents se produisent.
- Les agents sont intrinsèquement communicatifs, tandis que les services Web sont passifs jusqu'à ce qu'ils soient invoqués. Les agents peuvent fournir des alertes et des mises à jour lorsque de nouvelles informations sont disponibles.
- Un service Web, tel qu'il est actuellement défini et utilisé, n'est pas autonome. L'autonomie est une caractéristique des agents. Parmi les agents, l'autonomie se réfère généralement à l'autonomie sociale, où un agent est conscient de ses collègues et est sociable, mais exerce son indépendance dans certaines conditions.
- Les agents sont coopératifs et, en formant des équipes, les alliances peuvent fournir des services plus complets .
- Les agents peuvent fournir des services à d'autres agents, mais aussi aux consommateurs de services Web, qu'ils soient commerciaux ou individuels.
- Les agents peuvent obtenir des informations du grand référentiel sur le Web sous la forme de services Web.

## 2.7 Les travaux connexes

### 2.7.1 Travail N 1

Dans le travail effectué par « Guobing Zou, Yang Xiang, Yanglan Gan, Dong Wang and Zengbao Liu, "An agent-based web service selection and ranking framework with QoS ", 2009 IEEE » les auteurs ont utiliser un nouveau composant qui s'appelle WSDM-Q (modèle de description de service Web) nous qui considère des informations sur la qualité de service et qui contient deux parties de définitions : service web et demande de service.

Ils ont utilisé Une sélection de service et un classement avec QoS WSSR-Q( selection ranking web service) basé sur le modèle de description de service précédent. Ils ont proposé :

- Un algorithme de sélection de service avec QoS (SSA-Q) satisfaisant pour les exigences de QoS de base de l'utilisateur.
- Un algorithme de classement de service avec QoS (SRA-Q) pour normaliser et calculer des valeurs de QoS complètes de tous les services candidats.
- Un mécanisme de mise à jour de qualité concernant la valeur des attributs QoS.

Ils ont confirmé que, pour répondre aux exigences non-fonctionnelles des demandeurs de service et atteindre une meilleure efficacité de sélection de service Web il est mieux d'utilisé (WSSR-Q).

### 2.7.2 Travail N 2

Dans le travail proposé «Mojtaba Khezrian, Wan M. N. Wan Kadir, Suhaimi Ibrahim, and Alaeddin Kalantari, " A Hybrid Approach for Web Service Selection ", International Journal Of Computational Engineering Research, 2012» les auteurs ont proposé une approche hybride pour résoudre le problème de la sélection de services Web.

Tout d'abord, ils ont appliqué le processus de hiérarchie analytique (AHP) pour évaluer les poids des critères au lieu de collecter les poids. Directement du consommateur de service.

Ensuite, ils ont utilisé VIKOR (VIšekriterijumsko KOmpromisno Rangiranje) pour identifier et classer les services candidats appropriés.

Ils ont présenté une approche pour la sélection des service web comme le montre la figure, premièrement, la pondération des critères sera évaluée par l'AHP, puis, pour la prise de décision, ils ont utilisé la méthode VIKOR. Les étapes de l'approche sont présentées ci-dessous :

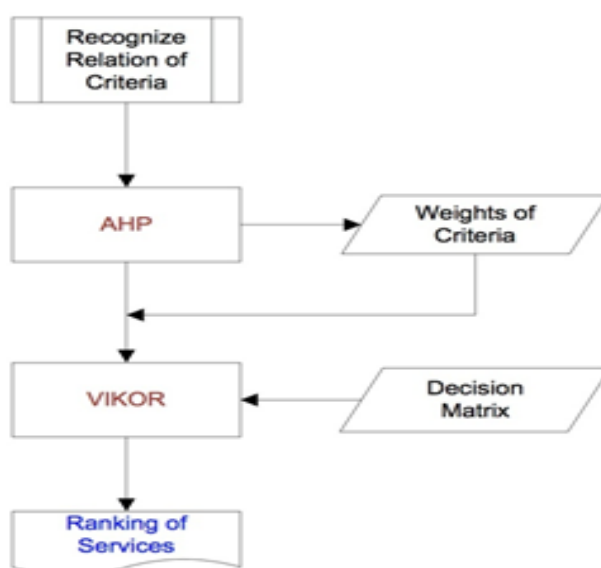


FIGURE 2.8 – Processus de l'approche proposée.

Ils ont appliqué la méthode VIKOR étape par étape pour surmonter le problème de sélection de service dans la vue de MCDM. A la fin ils ont fourni un exemple en utilisant des critères de QoS et d'autres services alternatifs. Le résultat montre que leur approche peut sélectionner les meilleurs et les plus proches des candidats.

## **2.8 Conclusion**

Nous avons présenté dans ce chapitre, le problème de sélection des services web qui est considéré comme étant une tâche importante vue sa complexité exponentielle. Et il nous a permis d'étudier quelques travaux basés sur la sélection, Nous avons consacré ce chapitre pour présenter un état de l'art qui regroupe les différentes approches proposées pour résoudre ce problème.

Dans le chapitre suivant, nous présenterons notre architecture basée agents pour la sélection des services web en tenant en compte les préférences de QoS définies par l'utilisateur.

Chapitre

3

# Conception du système

### 3.1 Introduction

Dans les chapitres précédents, nous avons présenté un état de l'art sur l'architecture orientée services, services web, les qualités des services web, les agents et les systèmes multi agents, la sélection à base des critères de QoS, les méthodes d'aides à la décision multicritère, explication détaillée de l'algorithme TOPSIS. Ce que nous a permis de comprendre et de situer clairement les notions de base pour présenter notre travail.

Pour résoudre ces problèmes, nous présentons dans ce chapitre un aperçu de l'approche proposée dans ce travail. Il s'agit d'une méthode de sélection qui utilise deux algorithmes AHP et TOPSIS pour classer les services web à base des critères de qualité QoS, en fonction des exigences non-fonctionnelles des agents métier (l'agent métier représentant l'utilisateur final). Nous avons représenté notre conception système avec le langage UML (Unified Modeling language) qui est un standard ouvert propre par l'OMG (Object Management Group).

### 3.2 Architecture générale

L'architecture globale de notre approche, qui exprime l'extension de l'architecture de base SOA, en ajoutant une nouvelle composante entre le client et l'annuaire UDDI de telle façon la phase de la recherche des services similaires sera donc simple et facile et pour faire l'évaluation des critères de qualité de service QoS en calculons leurs poids à l'aide de l'algorithme AHP, aussi pour la classification de ces services selon l'algorithme TOPSIS.

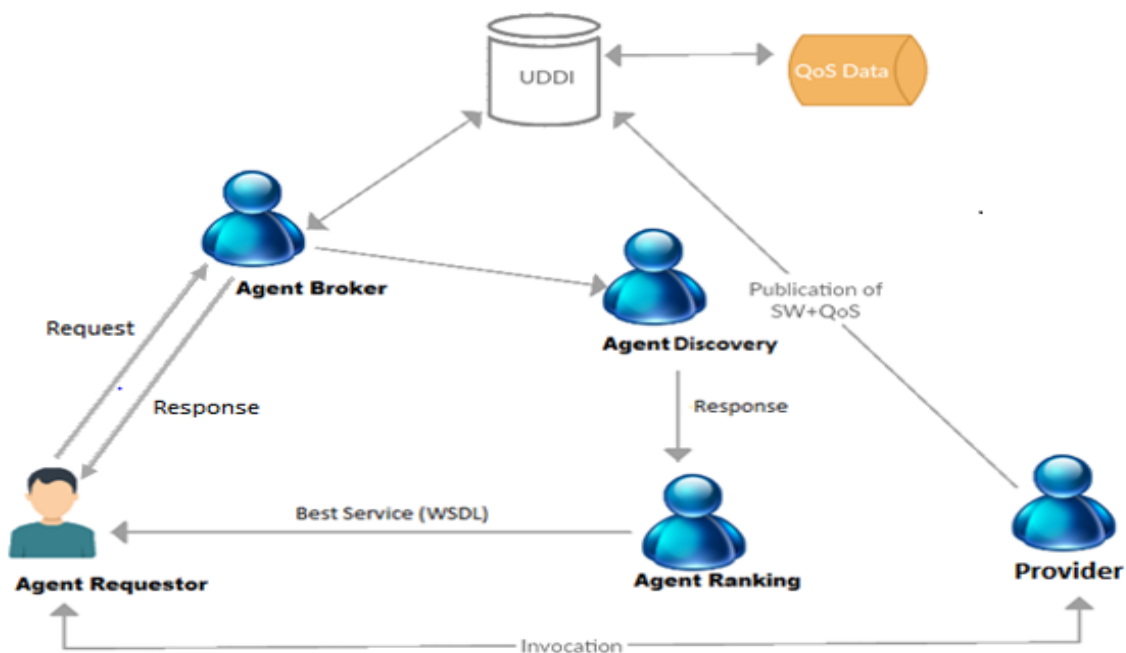


FIGURE 3.1 – Architecture proposée.

### 3.2.1 Description des agents

Dans cette section nous allons décrire en détail l'aspect structurel et fonctionnel des agents présents dans notre système.

- **Agent Requestor (client)** : c'est un agent qui souhaite avoir un service en lançant une requête à l'agent Broker. Il contient deux modules :
  - **Module interface** : Ce module permet aux agents d'interagir entre eux.
  - **Module de communication** : c'est un module responsable de l'échange de message entre l'agent courant et le reste des agents.



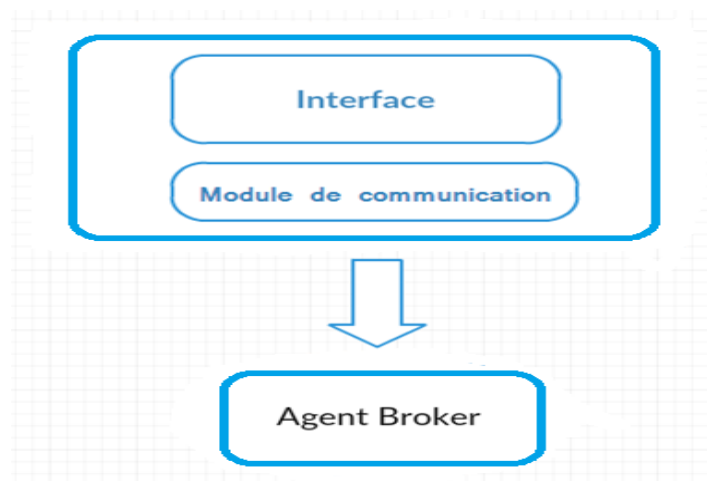


FIGURE 3.2 – Architecture de l'agent Requestor.

- **Agent Broker** : C'est un agent mobile qui s'occupe de la requête émise par le client , en cherchant les services souhaitées dans le registre UDDI il va les transmettre à l'agent discovery. Il contient un module de communication, il possède aussi une base de données contenant une liste des services web similaires.

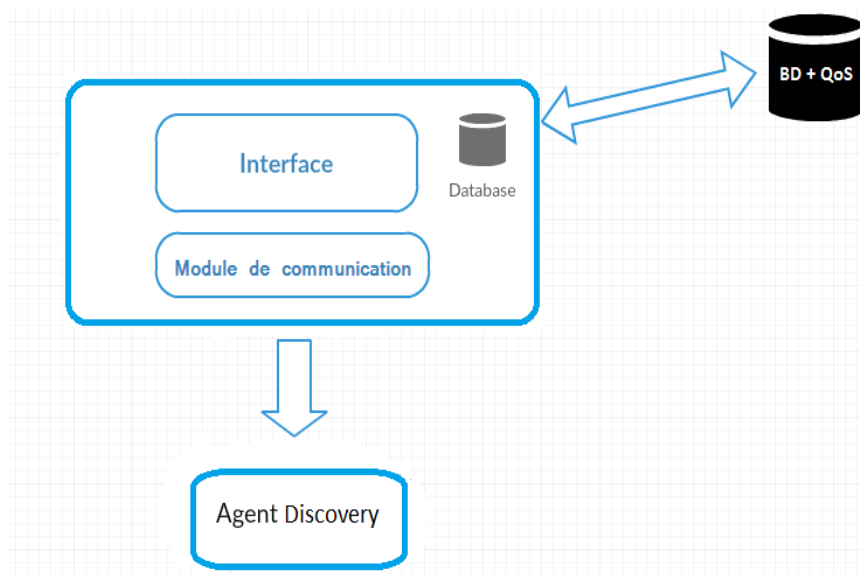
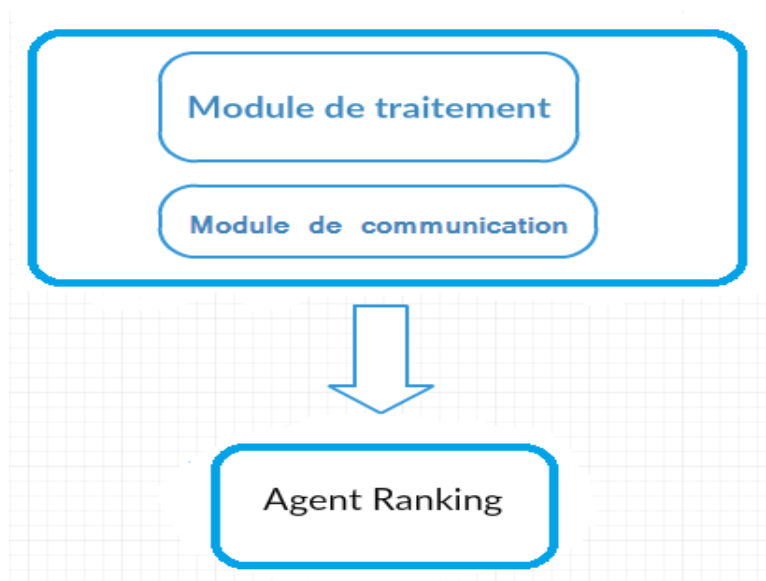


FIGURE 3.3 – Architecture de l'agent Broker.

- **Agent Discovery** : C'est un agent qui s'occupe de la découverte des services. Il possède un module de traitement qui est chargé de calculer le poids des critères de ses services provenant de l'agent Broker à

l'aide de l'algorithme « AHP ».



**FIGURE 3.4** – Architecture de l'agent Discovery.

- **Agent Ranking** : C'est un agent qui s'occupe de la sélection et le classement des services. Il possède un module de classement qui est chargé de sélectionner le meilleur service en classant les services similaires provenant de l'agent discovery en utilisant l'algorithme « TOPSIS ».

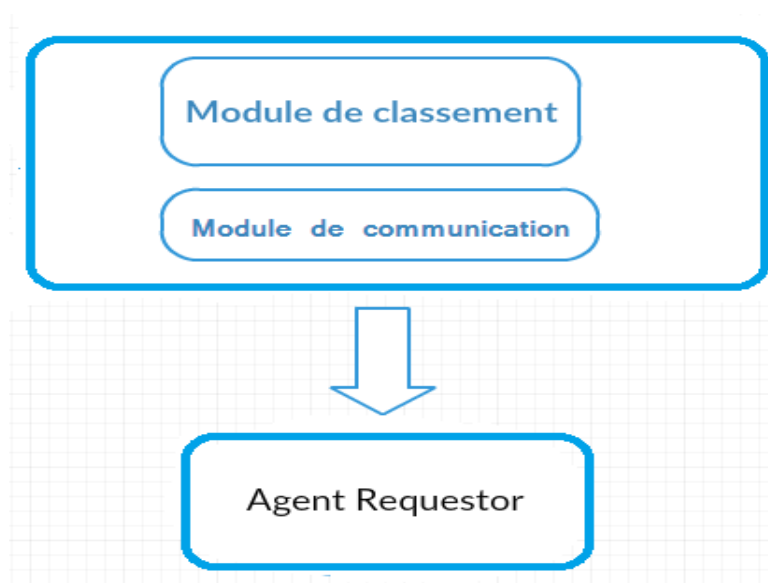


FIGURE 3.5 – Architecture de l'agent Ranking.

### 3.3 Description

Notre architecture démontre le cycle de vie d'un internaute qui demande un service selon plusieurs critères. Avant l'utilisation de n'importe quel service par le client, il doit être publié par le fournisseur.

Au début le client « Agent requestor » lance une requête de demande de service web à « Agent Broker » qui est , en liaison directe avec le registre, et il va procéder la recherche dans ce registre les services souhaités compris dans la requête. Cette requête comporte des critères non-fonctionnels.

Une fois « Agent Broker » trouve les services il va lui répondre par une liste des services similaires, Et il va transmettre cette liste aussi à « Agent Discovery » qui va découvrir ses services tout en commençant par le calcul du poids de leurs critères à l'aide de l'algorithme qui s'appelle « AHP », ensuite il va envoyer la liste des services et le poids calculé à « Agent Ranking ». Ce dernier procède au classement de cette liste selon un algorithme de classement qui s'appelle « TOPSIS » afin de fournir au client le service adéquat.

Ces deux algorithmes sont considérés comme une méthode d'analyse de décisions multicritères, qui va nous donner une liste de services web ordonnée selon les critères de QoS, de tel façon chaque service, prend un poids qui confirme son rang, ce poids nous aide pour faire une meilleure sélection. Pratiquement, le poids le plus grand correspond au meilleur service à remettre au client.

### 3.4 Fonctionnement général du système

Les figures [3.6](#) et [3.7](#) présentent les diagrammes de séquence qui décrivent le scénario nominal de fonctionnement de notre prototype.

- **Diagramme de séquence de la phase de publication des services.**

Au début le fournisseur doit publier les services web disponibles au niveau de l'annuaire et bien sûr avec l'évaluation de ses critères de qualités QoS.

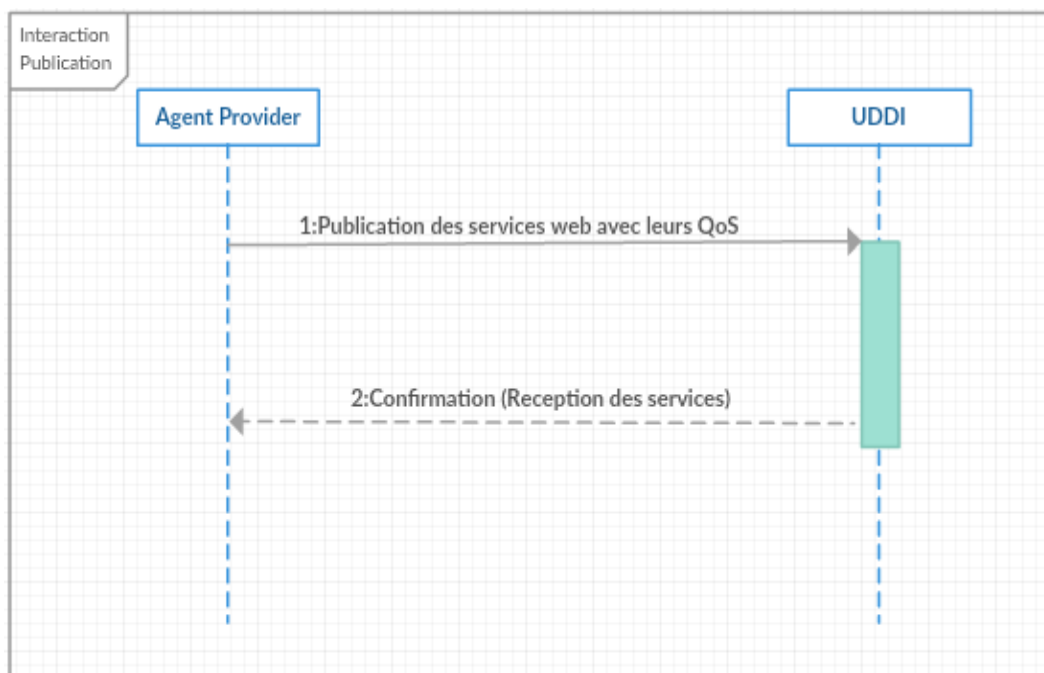


FIGURE 3.6 – Diagramme de séquence de la phase de publication des services

- **Diagramme de séquence de de fonctionnement générale (sélection)**

Après la publication des services web par le fournisseur.

- L'agent Requestor lance une requête fonctionnelle à l'agent Broker pour demander un service parmi les services disponibles avec des critères de qualité de service (critères non fonctionnelles).
- L'agent Broker va chercher les services souhaités dans l'annuaire UDDI et il va lui (agent requestor) envoyer la liste des services similaires
- L'agent Broker va envoyer aussi cette liste à l'agent Discovrey, ce dernier va passer au traitement (calculer le poids des critères) à l'aide de l'algorithme AHP.

Ensuite il va envoyer la liste des services et le poids calculer à l'agent Ranking ce dernier procède à faire une classification des services web selon notre algorithme TOPSIS afin de fournir au client le service adéquat (meilleur service).

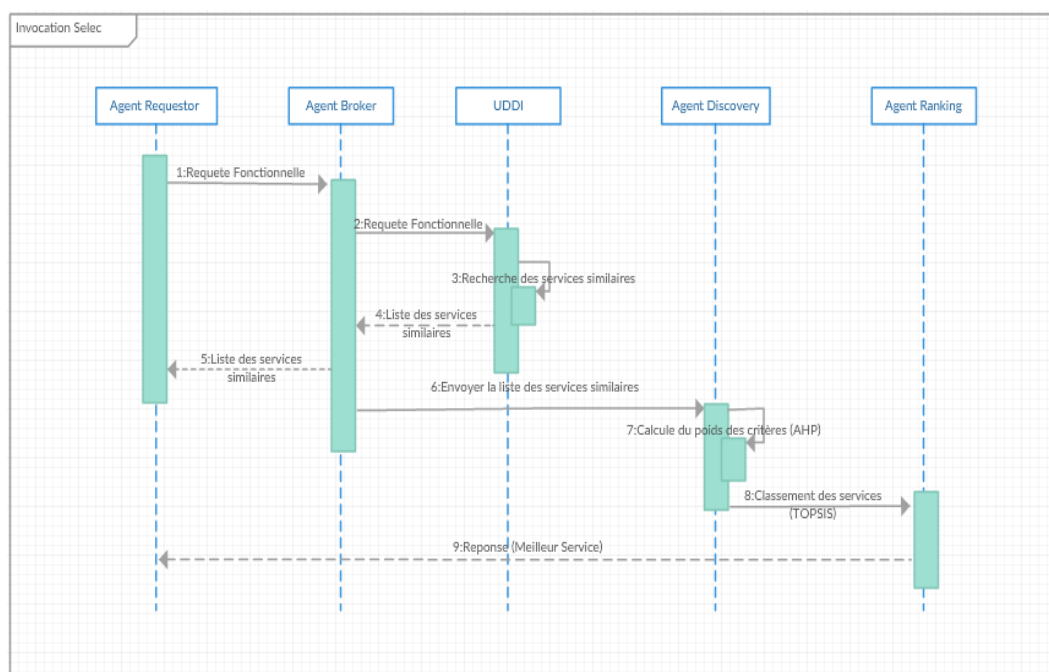


FIGURE 3.7 – Diagramme de séquence du fonctionnement générale de l’application.

- **Classement des services similaires**

Pour fournir aux clients le meilleur service, notre approche utilise deux algorithmes : le premier est l’algorithme AHP qui s’occupe de calculer le poids de chaque critère alors que le second algorithme TOPSIS qui se charge de classer les services similaires.

### 3.5 Conclusion

Dans ce chapitre nous avons présenté notre modèle proposé pour la sélection des services web en se basant sur les agents. Nous avons intégré deux algorithmes de décision le premier pour le traitement des services et le deuxième pour faire le classement des services par ordre selon des critères non fonctionnels. Nous avons exprimé également le fonctionnement de notre système en se basant principalement sur les différents diagrammes UML et sur les scénarios de description pour chaque phase.

Cette étude conceptuelle présente l'architecture générale de notre modèle. Dans le prochain chapitre nous allons présenter les techniques utilisées pour implémenter l'application. Ainsi que, une étude de cas sur un exemple concret.

Chapitre



# **Implémentation et étude de cas**



## 4.1 Introduction

Après avoir présenté en détails notre approche de la sélection des services Web en prenant en compte les critères de qualité de service avec l'utilisation des agents et des deux algorithmes qui aide à la décision multicritère AHP et TOPSIS dans le chapitre précédent, ce chapitre sera consacré à la phase d'implémentation. Dans la quelle nous commençons par la présentation de l'environnement logiciel utilisé, à travers la présentation des outils et des langages de programmation. Nous présenterons ensuite la réalisation de notre système à travers des interfaces.

## 4.2 Environnement de développement

Avant de commencer l'implémentation de notre application, nous allons tout d'abord spécifier les langages de programmation et les outils utilisés qui nous ont semblé être un bon choix vu les avantages qu'ils offrent.

### 4.2.1 Environnement matériel et logiciel

Le matériel utilisé est un ordinateur portable décrit dans la figure suivante :

### 4.2.2 Langage de programmation

De nos jours il existe de nombreux langages de programmation, plus au moins dédiés à tel ou tel type d'applications particulières. Parmi eux, notre choix s'est focalisé sur le langage JAVA.

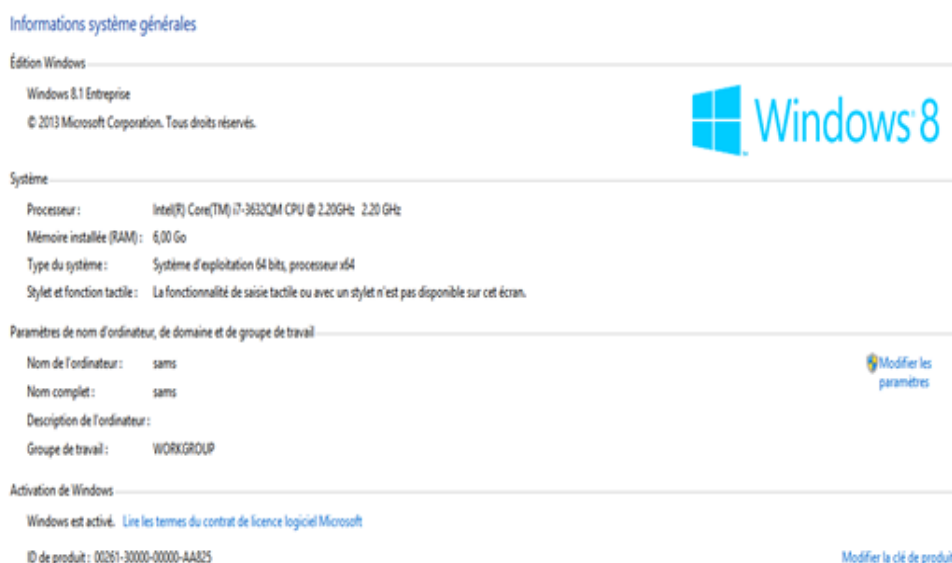


FIGURE 4.1 – Environnement logiciel utilisé.

#### 4.2.2.1 Le langage Java

Apparu fin 1995 début 1996 et développé par Sun Microsystems Java s'est très rapidement taillé une place importante en particulier dans le domaine de l'internet et des applications client-serveur. Les objectifs de Java sont d'être multiplateformes et d'assurer la sécurité aussi bien pendant le développement que pendant l'utilisation d'un programme Java. Il est en passe de détrôner le langage C++ dont il hérite partiellement la syntaxe mais non ses défauts. Comme C++ et Delphi, Java est algorithmique et orienté objet à ce titre il peut effectuer comme ses compagnons, tous les tâches d'un tel langage (bureautiques, graphiques, multimédias, base de données, environnement de développement, etc. . . .) .son point de fort qui le démarque des autres est sa portabilité due(en théorie) à ses bibliothèques de classes indépendantes de la plate-forme ,ce qui est le point essentiel de la programmation sur internet ou plusieurs machines dissemblables sont interconnectées[33].

### 4.2.3 Outils et technologies

#### 4.2.3.1 Eclipse

Eclipse est un projet, décliné et organisé en un ensemble de sous-projets de développements logiciels, de la fondation Eclipse visant à développer un environnement de production de logiciels libre qui soit extensible, universel et polyvalent, en s'appuyant principalement sur Java[33].

Son objectif est de produire et fournir des outils pour la réalisation de logiciels, englobant les activités de programmation (notamment environnement de développement intégré et frameworks) mais aussi d'AGL recouvrant modélisation, conception, test, gestion de configuration, reporting... Son EDI, partie intégrante du projet, vise notamment à supporter tout langage de programmation à l'instar de Microsoft Visual Studio.

Bien qu'Eclipse ait d'abord été conçu uniquement pour produire des environnements de développement, les utilisateurs et contributeurs se sont rapidement mis à réutiliser ses briques logicielles pour des applications clientes classiques. Cela a conduit à une extension du périmètre initial d'Eclipse à toute production de logiciel : c'est l'apparition du framework Eclipse RCP en 2004.

Figurant parmi les grandes réussites de l'open source, Eclipse est devenu un standard du marché des logiciels de développement, intégré par de grands éditeurs logiciels et sociétés de services. Les logiciels commerciaux Lotus Notes 8, IBM Lotus Symphony ou WebSphere Studio Application Developer sont notamment basés sur Eclipse.

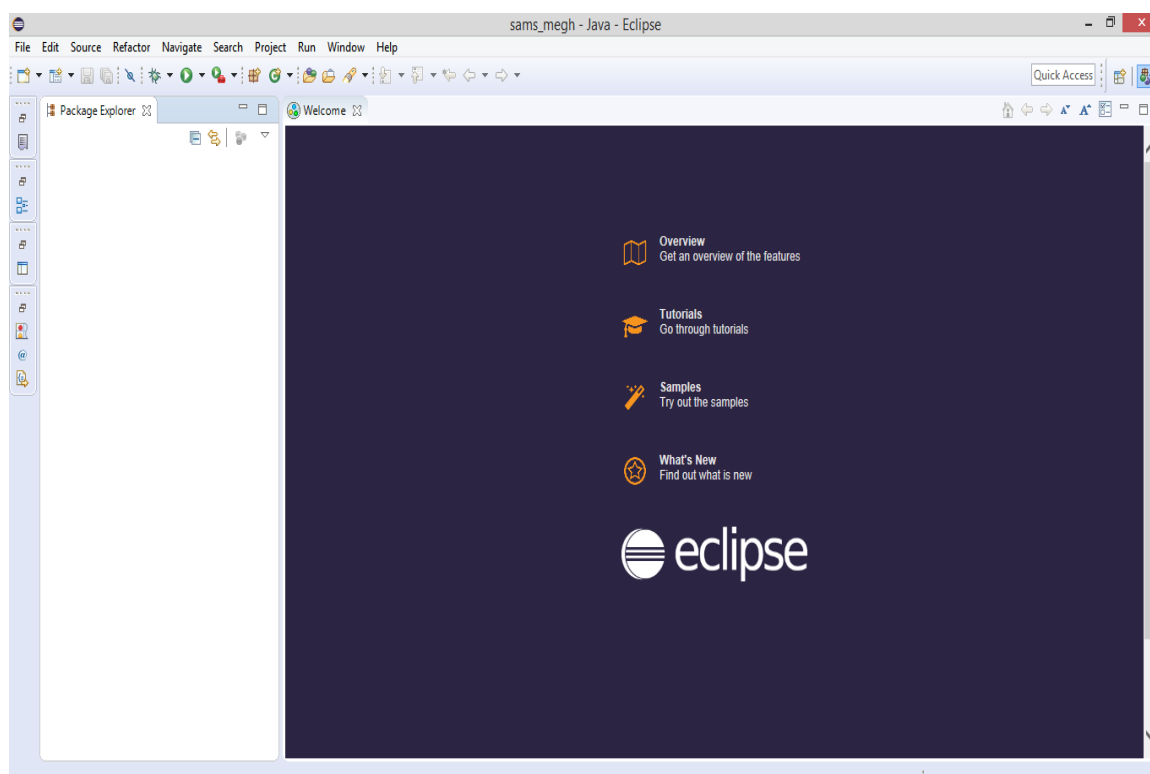


FIGURE 4.2 – Interface principale de eclipse.

#### 4.2.3.2 La plateforme JADE

JADE (Java Agent Development Framework) est une plateforme open sourcedéveloppée en Java par CSELT (Groupe de recherche de Gruppo Telecom, Italie) qui a comme but la construction des systèmes multi-agents et la réalisation d'applications conformes à la norme FIPA en mettant à disposition un middleware et plusieurs outils graphiques. JADE comprend deux composantes de base : une plate-forme agents compatible FIPA (Foundation for Intelligent Physical Agent) et un paquet logiciel pour le développement des agents Java[34].



FIGURE 4.3 – Logo de Jade

JADE contient :

- Un Runtime Environment : où les agents peuvent vivre.
- Une librairie de classes : que les développeurs utilisent pour écrire leurs agents.
- Une suite d'outils graphiques : qui facilitent la gestion et la supervision de la plateforme des agents.

Pour suivre les spécifications FIPA, JADE possède trois modules essentiels :

1. Un DF (Directory Facilitator) : fournissant un service de « pages jaunes » à la plateforme.
2. Un ACC (Agent Communication Channel) : pour gérer la communication entre les agents.
3. Un AMS (Agent Management System) : permettant de superviser l'enregistrement des agents, leur authentification, leur accès et l'utilisation du système.

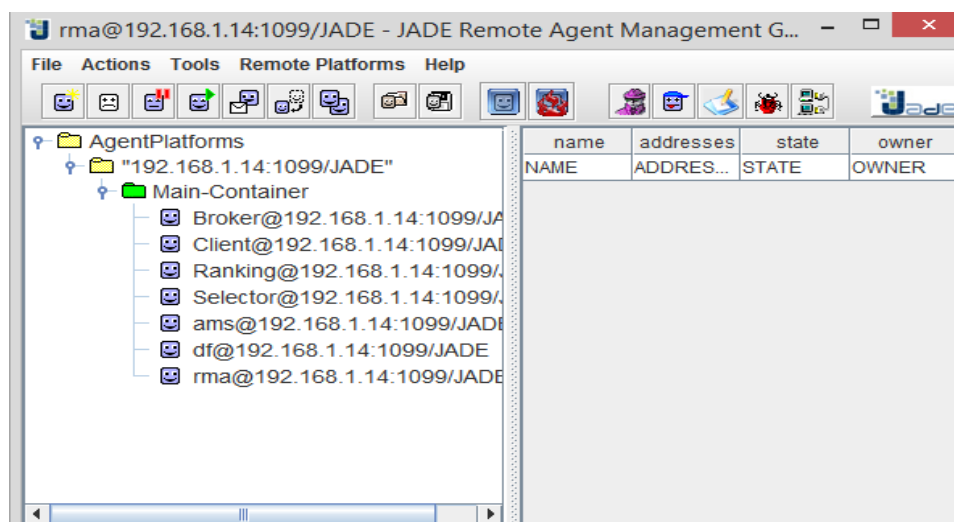


FIGURE 4.4 – Interface graphique de la plateforme JADE.

### 4.2.3.3 Gestion de base de données

- **MySQL :**

MySQL est un système de gestion de données relationnelles (SGBDR), rapide, robuste et facile d'utilisation. Il est adapté à la gestion de données dans un environnement réseau. Il est fourni avec des nombreux outils et qui est compatible avec de nombreux langages de programmation. Il est célèbre SGBDR du monde open-source, MySQL est surtout installé pour les applications Web, avec le serveur Xampp et le langage de page web dynamique JSP[35].



FIGURE 4.5 – Logo MySQL

- **PhpMyAdmin :**

phpMyAdmin est une interface d'administration pour le SGBD MySQL. Il est écrit en langage PHP et s'appuie sur le serveur HTTP Apache.

Il permet d'administrer les éléments suivants [36] :

- les bases de données.
- les tables et leurs champs (ajout, suppression, définition du type).
- les index, les clés primaires et étrangères.
- les utilisateurs de la base et leurs permissions.



FIGURE 4.6 – Logo PhpMyAdmin.

- **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 (XApache 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 et fonctionne, de plus, sur les systèmes d'exploitation les plus répandus[37].



FIGURE 4.7 – Logo XAMPP

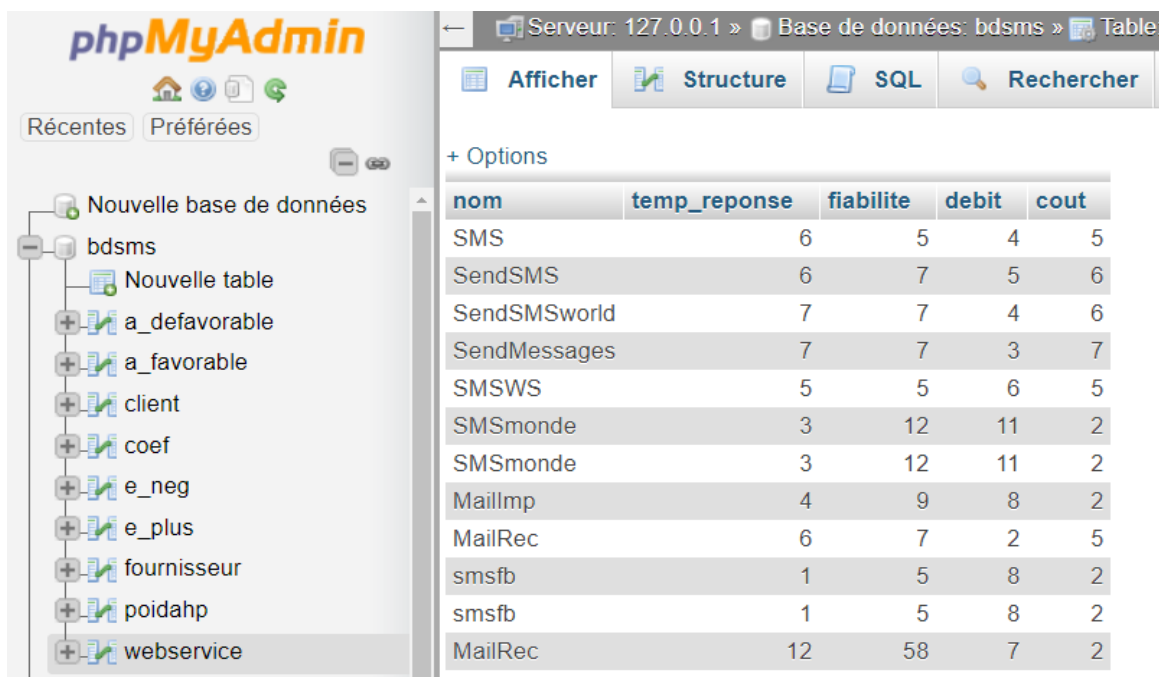
#### 4.2.4 La base de données

Les critères de qualité de service utilisés dans notre prototype sont :

- **Temps de réponse (ResponseTime) :** Il est une performance qui représente la vitesse avec laquelle un service Web répond à une requête. Il est mesuré en milliseconde.
- **Fiabilité (Reliability) :** Elle est la capacité d'un service à remplir ses fonctions requises dans les conditions indiquées pour une période de

temps déterminée. Elle est mesurée en pourcentage.

- **Débit (throughput)** :C'est le nombre de demandes de service complétées sur une période de temps[22].
- **Coût d'un service (Cost)** :C'est le coût à payer pour consommer le service, ce coût peut être fourni par le fournisseur du service.



nom	temp_reponse	fiabilite	debit	cout
SMS	6	5	4	5
SendSMS	6	7	5	6
SendSMSworld	7	7	4	6
SendMessages	7	7	3	7
SMSWS	5	5	6	5
SMSmonde	3	12	11	2
SMSmonde	3	12	11	2
MailImp	4	9	8	2
MailRec	6	7	2	5
smsfb	1	5	8	2
smsfb	1	5	8	2
MailRec	12	58	7	2

FIGURE 4.8 – Table Base de donnée1.



### 4.2.5 Communication entre les agents

La plateforme JADE offre un outil graphique permettant de visualiser les échanges de messages entre les différents agents du système. L'agent Sniffer, est une application Java créée pour le suivi des messages échangés dans un environnement d'agents. Le Sniffer est complètement intégré dans l'environnement Jade. L'annonce est particulièrement utile lors du débogage des comportements des agents.

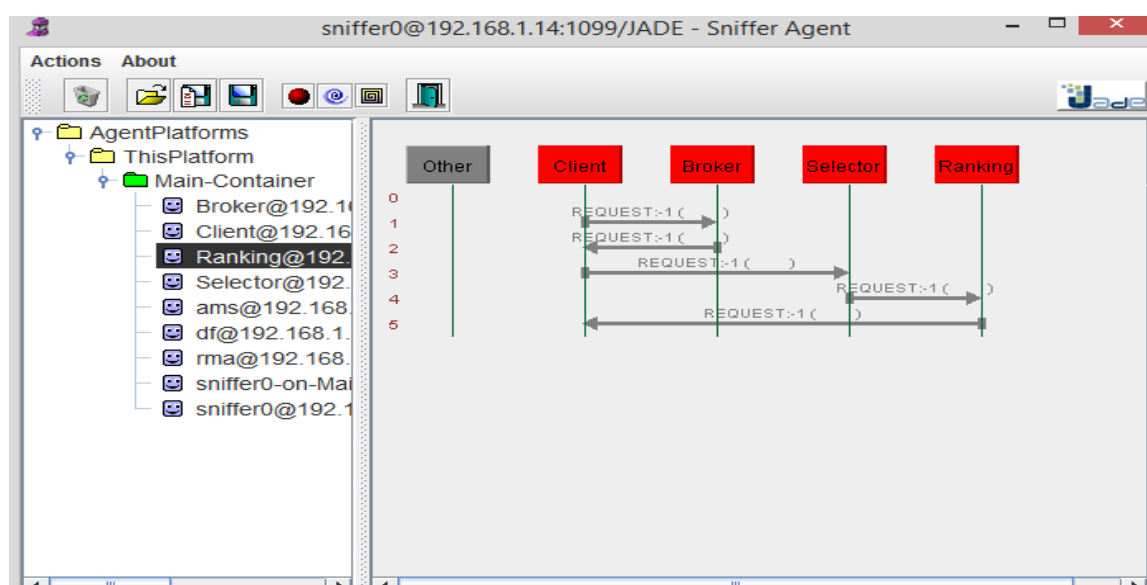


FIGURE 4.9 – Visualisation des échanges de messages entre les agents du système.

## 4.3 Présentation des interfaces graphiques

### 4.3.1 Interface d'accueil :

La qualité de l'interface est l'une des caractéristiques qui attire l'utilisateur. De cela nous avons essayé de la représenter dans une bonne forme tout en respectant l'aspect de simplicité.



FIGURE 4.10 – Interface principale de l'application.

### 4.3.2 Les interfaces principales de notre prototype :

Notre système contient plusieurs interfaces qui traitent les différents cas d'utilisation des services Web comme suit :

#### 4.3.2.1 Interface d'Inscription du Fournisseur :

Comme on a expliqué en haut l'utilisateur doit préciser son statut avant de s'inscrire. Nous montrons l'interface de formulaire d'inscription du fournisseur comme le montre la figure 4.11 Dans la partie formulaire d'inscription de fournisseur on trouve une partie concerne les informations personnelles du fournisseur.



nom X

prenom

username

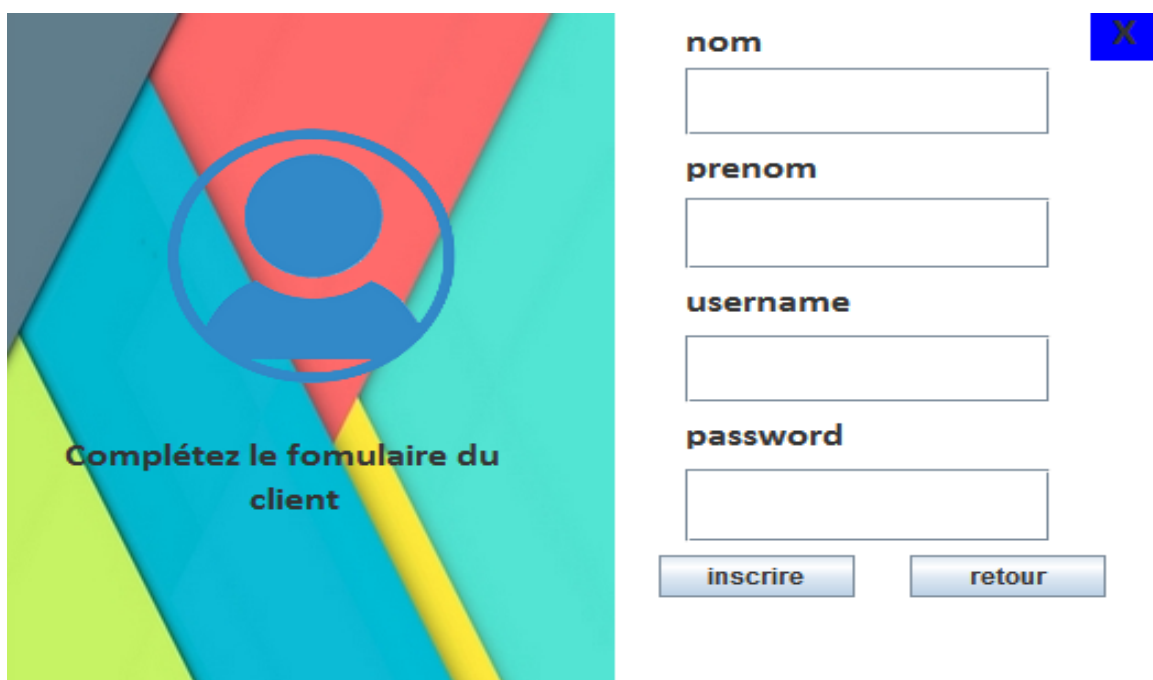
password

inscire retour

FIGURE 4.11 – interface d’inscription du fournisseur.

#### 4.3.2.2 Interface d’Inscription du Client :

Tout comme le fournisseur un client doit aussi s’inscrire afin de pouvoir réaliser ses taches. La figure 4.12 montre l’interface d’inscription du client.



nom X

prenom

username

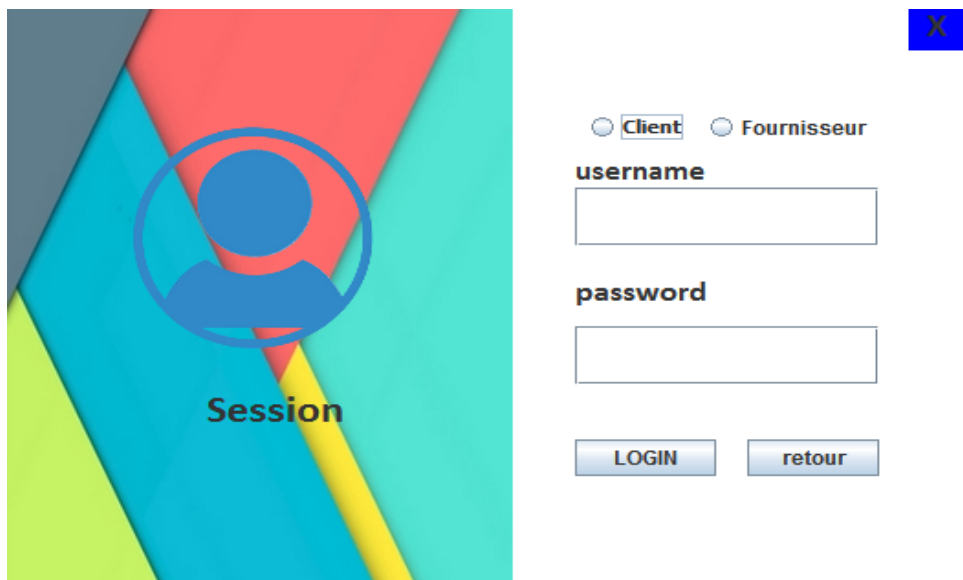
password

inscire retour

FIGURE 4.12 – interface d’inscription du client.

### 4.3.2.3 Interface Login :

Dans le cas d'ajout d'un nouveau service Web à notre base de données par un fournisseur des services il faut d'abord authentifier l'identité de fournisseur.



The image shows a login interface. On the left side, there is a colorful abstract graphic with a blue person icon and the word "Session". On the right side, there is a form with two radio buttons: "Client" (selected) and "Fournisseur". Below the radio buttons are two input fields: "username" and "password". At the bottom of the form are two buttons: "LOGIN" and "retour". In the top right corner of the form area, there is a blue square button with a white "X" icon.

FIGURE 4.13 – Interface Login.

#### 4.3.2.4 Interface d'Espace Client :

La même chose pour le client, une fois il s'inscrit il sera rediriger vers « l'espace client » qui lui permet de découvrir (chercher), sélectionner les services satisfaisant ses besoins à travers l'interface montré dans la figure 4.15.



FIGURE 4.14 – Interface Client.

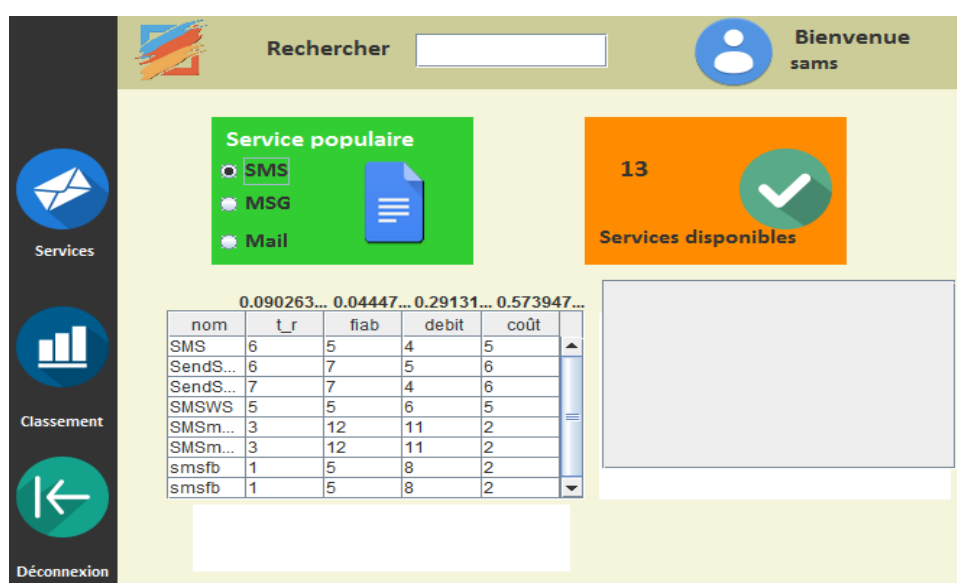


FIGURE 4.15 – Interface d'affichage des services similaires (Service SMS).

Rechercher

Bienvenue megherbi

**Service populaire**

- SMS
- MSG
- Mail

11 Services disponibles

0.090263... 0.04447... 0.29131... 0.573947...

nom	t_r	fiab	debit	coût
MSGVer...	4	44	78	2
MSGNot...	7	74	98	9
MSGSer...	5	47	98	8
MSGbook	10	9	96	4
MSGbook	10	9	96	4

Services

Classement

Déconnexion

FIGURE 4.16 – Interface d’affichage des services similaires (Service MSG).

Rechercher

Bienvenue sams

**Service populaire**

- SMS
- MSG
- Mail

13 Services disponibles

0.090263... 0.04447... 0.29131... 0.573947...

nom	t_r	fiab	debit	coût
MailImp	4	9	8	2
MailRec	6	7	2	5
MailRec	12	58	7	2

Services

Classement

Déconnexion

FIGURE 4.17 – Interface d’affichage des services similaires (Service MAIL).

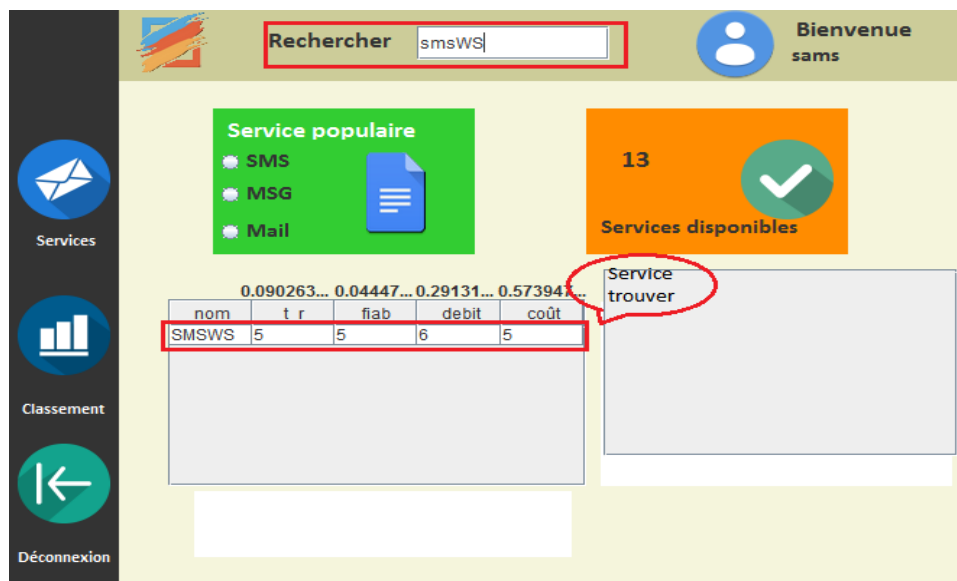


FIGURE 4.18 – Interface recherche d'un service.

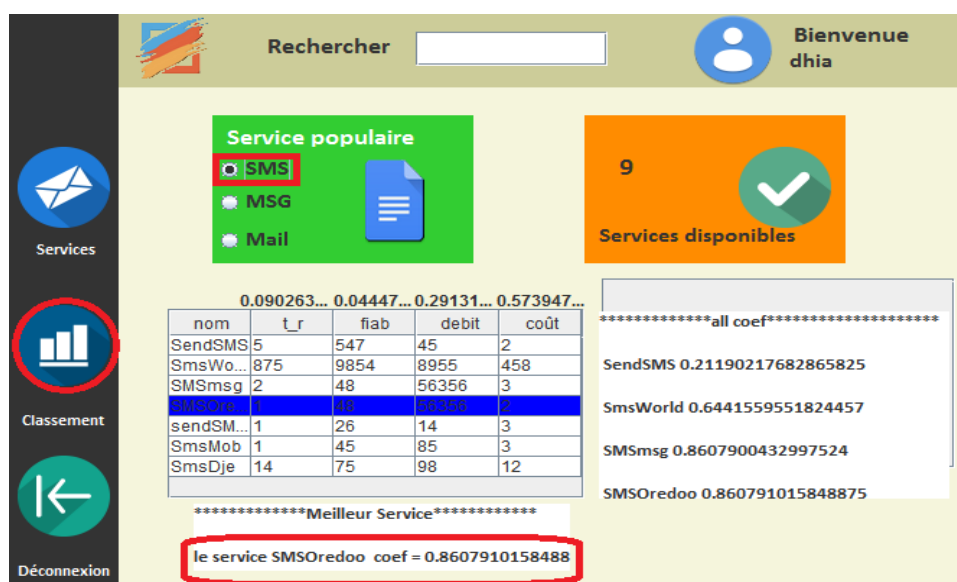


FIGURE 4.19 – Résultat calculé par TOPSIS.

### 4.3.2.5 Interface d'Espace Fournisseur :

Une fois un fournisseur s'inscrit il sera rediriger vers son espace personnel qui est l' « Espace Fournisseur » qu'à travers il pourra publier les descriptions de son service. La figure 4.20 montre l'interface de l'espace du fournisseur et les différentes fonctionnalités d'un fournisseur.



FIGURE 4.20 – Interface Fournisseur.

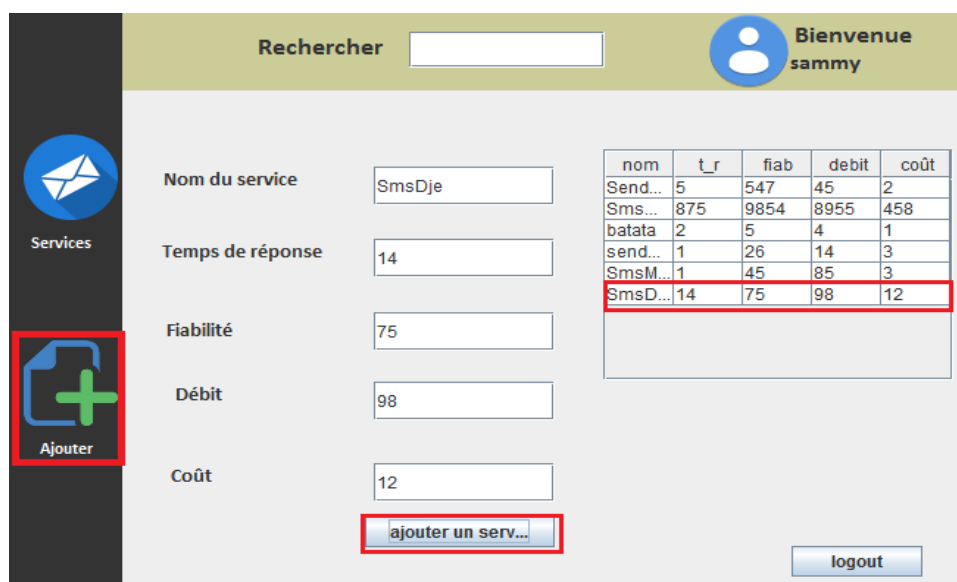


FIGURE 4.21 – Interface ajout d'un nouveau service.



Rechercher

Bienvenue sammy

Services

Ajouter

Nom du service

Temps de réponse

Fiabilité

Débit

Coût

ajouter un serv...

logout

nom	t_r	fiab	debit	coût
Send...	5	547	45	2
Sms...	875	9854	8955	458
batata	2	5	4	1
send...	1	26	14	3
SmsM...	1	45	85	3
SmsD...	14	75	98	12

FIGURE 4.22 – Interface affichage des services.

Rechercher

Bienvenue sammy

Services

Ajouter

Nom du service

Temps de réponse

Fiabilité

Débit

Coût

ajouter un serv...

logout

nom	t_r	fiab	debit	coût
SmsM...	1	45	85	3

FIGURE 4.23 – Interface Recherche d'un nouveau service.

## 4.4 Conclusion

Pour valider la conception d'un système, il faut l'implémenter en utilisant les outils adéquats. Ce chapitre, est consacré pour montrer la faisabilité de notre approche de sélection des services Web à base de qualité de service (QoS) avec une intégration de deux algorithmes le premier qui est AHP pour calculer le poids des critères et le second qui s'appelle TOPSIS, qui il va faire un classement des services avec leurs qualités à travers des étapes bien définies et bien spécifier. Nous testé notre proposition avec une BD de service web similaires.

D'après les résultats obtenus, nous pouvons confirmer que les résultats sont encourageantes et que les deux algorithmes (AHP et TOPSIS) sont très efficaces dans le domaine de sélection de services Web.

Nous avons présenté quelques détails concernant la réalisation de notre application, en choisissant le langage de programmation java pour sa compatibilité avec les concepts orienté objet, et également pour ces avantages multiples de simplicité, portabilité et sécurité. Nous avons aussi présenté un exemple illustrant les différents services offerts par notre application.

## Conclusion Générale

Les services Web sont des technologies émergentes et prometteuses pour le développement, le déploiement et l'intégration d'applications sur l'Internet. Ils constituent la technologie de base pour le développement d'architectures orientées services. Ces architectures sont de plus en plus répandues sur le Web. Le principe essentiel de l'approche service Web est de transformer le Web en un dispositif distribué d'échange et de calcul, où les services Web peuvent interagir d'une manière intelligente. Actuellement, de nombreux services Web, avec des fonctionnalités similaires sont fournis par des fournisseurs concurrents, et de ce fait les utilisateurs finaux ont besoins d'approches efficaces pour la sélection des services.

Nous avons proposé une nouvelle approche qui sert à la découverte des services Web, en s'appuyant sur deux algorithmes AHP pour le calcul des poids des critères et TOPSIS pour le classement des services similaires qui facilitera la sélection des services pertinents selon le besoin des utilisateurs.

Nous avons intégré le paradigme agent, qui représente une grande potentialité (autonomie, intelligence, coopération, coordination...etc ) dans le web.

Dans le premier chapitre, nous avons présenté les technologies des services Web et les principaux standards qu'elle supporte. Le second chapitre est consacré pour la sélection des services Web en se basent sur des propriétés non fonctionnelles, en présentant les méthodes qui

aident à la décision multi critères, ainsi les algorithmes de base utilisées (AHP et TOPSIS) et les systèmes multi agents. Dans le troisième chapitre, nous avons présenté la conception et le fonctionnement de notre système. Pour valider la conception du système, nous avons implémenté notre modèle conceptuel dont ses principales interfaces sont dans le quatrième chapitre.

Tout travail est amené à être amélioré, en ce sens, notre application peut encore évoluer et se voir améliorer. Pour une continuation de notre travail, plusieurs perspectives peuvent être envisagées :

- La mobilité dans le paradigme agent (agent mobile).
- Créer une application mobile universelle (Android, iOS, . . .) qui permet d'effectuer toutes les besoins qu'un client puisse invoquer.
- Permettre aux clients de choisir les critères.
- Maximiser le nombre de critères pour donner plus de choix aux clients.
- Tester notre prototype avec un DataSet réelles des services web.

# Bibliographie

- [1] Hamida, Soraya. *Une approche basée agent mobile pour le m-service web sémantique*. Diss. Université Mohamed Khider Biskra, 2014.
- [2] DALI YAHIA Mohammed. *Sélection des services web à base de QoS*. Université de Tlemcen, 2011.
- [3] AMARA, Manel Warda, and Nesrine ADDAD. *Implémentation d'un simple service web avec axis*. Diss. 2014.
- [4] BOUROUIS, Meriem .DJERRIRI, Sabah. *La sélection de services Web À base de l'algorithme d'abeilles*. Diss. 2014.
- [5] W3C Recommendation, URL : " [www.w3.org/2002/07/soap-translation/soap12-part1.html](http://www.w3.org/2002/07/soap-translation/soap12-part1.html)." Consulté : 24/11/2017.
- [6] W3C Recommendation, URL : " [www.w3.org/2002/07/soap-translation/soap12-part0.html](http://www.w3.org/2002/07/soap-translation/soap12-part0.html)." Consulté : 24/11/2017.
- [7] Newcomer, Eric. *Understanding Web Services: XML, WSDL, SOAP, and UDDI*. Addison-Wesley Professional, 2002.
- [8] Site officiel d'UDDI, URL : "<http://www.uddi.org/>". Consulté : 02/12/2017.
- [9] Chelbabi, M. "Découverte de Services Web Sémantiques: une Approche basée sur le Contexte, mémoire de magister." *INI Alger* (2006).
- [10] Site web ,URL : " [http://fr.wikipedia.org/wiki/Service\\_web](http://fr.wikipedia.org/wiki/Service_web)", Consulté : 14/12/2017.
- [11] Site web ,URL : "<http://www.techno-science.net/>", Consulté : 14/12/2017.
- [12] ZEPPEFELD, Klaus et FINGER, Patrick. *SOA und WebServices*. Springer-Verlag, 2009
- [13] Ulrich, Duvent. Guillaume, Ansel. *Les Architectures Orientées Services (SOA)*, Diss. Informatique ISIDIS. 2009.
- [14] BOUKAIS, Nouria. *Réalisation d'un service web selon l'architecture REST*. Diss. 2017.
- [15] Belaredj, Mohammed Amin. *La sélection des services web à base des systèmes immunitaires artificiels (CLONALG)*. Diss. 2014.
- [16] Wang, Hongbing, et al. "Web service selection for multiple agents with incomplete preferences." *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*. Vol. 1. IEEE, 2010.
- [17] IDRIS, Naziha. "Composition automatique des services Web à base des agents coopératifs.". Diss. Université de Kairouan. 2012.
- [18] Lopez-Velasco, Céline. *Sélection et composition de services Web pour la*

- génération d'applications adaptées au contexte d'utilisation*. Diss. Université Joseph-Fourier-Grenoble I, 2008.
- [19] HALFAOUI épouse GHERNAOUT, Amal. *La sélection des services web dans une composition à base de critères non fonctionnels*. Diss. 2017
- [21] Oskooei, Meysam Ahmadi, and Salwani Mohd Daud. "Quality of service (qos) model for web service selection." *Computer, Communications, and Control Technology (I4CT), 2014 International Conference on*. IEEE, 2014.
- [22] Menascé, Daniel A. "QoS issues in web services." *IEEE internet computing* 6.6 (2002): 72-75.
- [23] MOULAY KHATIR, Oussama Mohy-eddine. *Sélection de Service Web à base d'algorithme Mimétique*. Diss. 2014.
- [24] Belacel, Nabil, and Philippe Vincke. "Méthodes de classification multicritère: méthodologie et applications à l'aide au diagnostic médical." (2000).
- [25] Rolland, A. "Aide à la décision multicritère et apprentissage automatique pour la classification. 2012." (2014).
- [26] site web: URL:" <http://shodhganga.inflibnet.ac.in/bitstream/10603/2896/12>".
- [27] Zougari, Akram. *Une approche couplant logique floue et capitalisation des connaissances pour la résolution du problème de choix des fournisseurs*. Diss. Metz, 2011.
- [28] BEKHTAR, Mohammed, Cherif. *Les méthodes multi-critères pour analyser les aptitudes des terres agricoles : le cas du blé tendre en Languedoc-Roussillon, analysé avec la méthode AHP*. Diss. Université Paul Valéry de Montpellier.2015.
- [29] Ferber, Jacques. "Les systèmes multi-agents: un aperçu général." *Techniques et sciences informatiques* 16.8 (1997).
- [30] Hanachi, Chihab, and Christophe Sibertin-Blanc. "Protocol moderators as active middle-agents in multi-agent systems." *Autonomous Agents and Multi-Agent Systems* 8.2 (2004): 131-164.
- [31] Khezrian, Mojtaba, et al. "A hybrid approach for web service selection." *International Journal of Computational Engineering Research* 2.1 (2012): 190-198.
- [32] Belouaar, H., Kazar, O., & Rezeg, K. (2017, December). *Web service selection based on TOPSIS algorithm*. In *Mathematics and Information Technology (ICMIT), 2017 International Conference on* (pp. 177-182). IEEE.
- [33] Site web, URL : [https://fr.wikipedia.org/wiki/Java\\_\(langage\)](https://fr.wikipedia.org/wiki/Java_(langage)). Consulté le 29/04/2018.
- [34] Site web, URL : <http://jade.tilab.com/documentation/tutorials-guides/>. Consulté le 29/04/2018.

- [35] Site web, URL : <https://dev.mysql.com/downloads/> . Consulté le 10/05/2018.
- [36] Site web, URL : <https://doc.ubuntu-fr.org/phpmyadmin> . Consulté le 10/05/2018.
- [37] Site web, URL : <https://desgeeksetdeslettres.com/programmation-java/xampp-plateformepour-heberger-son-propre-site-web> . Consulté le 10/05/2018.
- [38] Huhns, Michael N. "Agents as Web services." *IEEE Internet computing* 6.4 (2002): 93- 95.