

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA  
RECHERCHE SCIENTIFIQUE  
Université 20 Août 1955-SKIKDA  
Faculté Des Sciences  
Département D'informatique



# *Mémoire de fin d'étude*

En vue de l'obtention du diplôme de Master

Spécialité : Informatique.

Option : Réseaux et Systèmes Distribués(RSD).

## *Thème*

---

**La sélection des services web à base de  
QoS.**

---

Réalisée par :

Mlle ZARZOUR FEDOUA

Devant le jury :

M. BENOUDINA LEZHAR

Président

M. MAAZOUZI SMAINE

Examineur

M. BOUHOUCHE ABDELOUAHID

Encadreur

PROMOTION 2023/2024

# Remerciement

*Je voudrais dans un premier temps remercier, mon encadreur de mémoire Dr. BOUHOUCHE, professeur de l'informatique à l'université de 20 aout Skikda, pour sa patience, sa disponibilité et surtout ses judicieux conseils, qui ont contribué à alimenter ma réflexion.*

*J'adresse mes sincères remerciements à tous les professeurs, intervenants et toutes les personnes qui par leurs paroles, leurs écrits, leurs conseils et leurs critiques ont guidé mes réflexions et ont accepté de me rencontrer et de répondre à mes questions durant mes recherches.*

*Je remercie mes très chers parents, Sací et Nadia, qui ont toujours été là pour moi. Je remercie mes sœurs et mes frères, pour leurs encouragements.*

*À tous ces intervenants, je présente mes remerciements, mon respect et ma gratitude.*

*Merci*

# *Dédicaces*

*Je dédié ce travail*

A toute ma famille.

A tous mes amis.

A tous ceux qui m'aime.

Et tous ceux que j'aime.

*Fedoua*

## Résumé :

Le marché de commercialisation des services Web sur internet ne cesse d'augmenter, ce qui résulte en un nombre de plus en plus croissant de services offrant des fonctionnalités équivalentes. De ce fait, la sélection d'un service web approprié pour une tâche particulière est devenue un défi difficile pour l'utilisateur. Cependant, ces services diffèrent dans leurs qualités de services (QoS). Ces dernières deviennent alors cruciales pour l'utilisateur, car elles permettent de l'aider à choisir parmi les services Web qui sont fonctionnellement équivalents.

De plus, il devient difficile de satisfaire l'exigence complexe d'un utilisateur par juste un seul service web individuel ou atomique, ce qui a conduit les concepteurs à composer les services Web existants. Par conséquent, la sélection n'est plus le simple choix d'un seul service atomique parmi tant d'autres pour une seule tâche, mais la sélection devient une opération complexe qui consiste à combiner un ensemble de services choisis pour chaque tâche, afin de former la composition qui satisfait au mieux les critères QoS de l'utilisateur.

Le problème est dans la sélection de ces services Web est comment distinguer le service qui répond aux exigences de l'utilisateur à partir de plusieurs services sur la base des besoins fonctionnels et/ou non-fonctionnels.

Le but de notre travail est de pouvoir sélectionner le meilleur service Web qui répond aux exigences de qualité de service attendues par le client.

**Mots clés :** service web, sélection des services web, qualité de service, méta-heuristique, optimisation combinatoire, algorithme génétique.

## **Abstract:**

The marketing market for web-based services on the internet continues to grow, resulting in an increasing number of services offering equivalent functionality. As a result, selecting a suitable web service for a particular task has become a difficult challenge for the user.

However, these services differ in their quality of service (QoS). These then become crucial for the user, as they help him choose among web services that are functionally equivalent.

Moreover, it becomes difficult to satisfy the complex requirement of a user by just one individual or atomic web service, which has led the designers to dial the existing web services. Therefore, selection is no longer the simple choice of one atomic service out of many for a single task, but selection becomes a complex operation of combining a set of services chosen for each task, in order to form the composition that best meets the QoS criteria of the user.

The problem in selecting these web services is how to distinguish the service that meets the user's requirements from multiple services on the basis of functional and / or non-functional needs.

The goal of our work is to be able to select the best web service that meets the quality-of-service requirements expected by the customer.

**Keywords:** web service, selection of web services, quality of service, meta-heuristics, and combinatorial optimization, Genetic Algorithm.

## الملخص:

ويتواصل نمو سوق خدمات التسويق على شبكة الإنترنت، مما يؤدي إلى زيادة عدد الخدمات التي توفر وظائف مماثلة. نتيجة لذلك، أصبح اختيار خدمة ويب مناسبة لمهمة معينة تحديًا صعبًا للمستخدم. ومع ذلك، تختلف هذه الخدمات من حيث جودة الخدمة (QoS) ثم يصبح الأخير حاسمًا للمستخدم، حيث يساعدونه في الاختيار بين خدمات الويب المكافئة وظيفيًا.

بالإضافة إلى ذلك، يصبح من الصعب تلبية متطلبات المستخدم المعقدة من خلال خدمة ويب فردية أو ذرية واحدة فقط، مما دفع المصممين إلى تكوين خدمات الويب الحالية. لذلك، لم يعد الاختيار هو الاختيار البسيط لخدمة ذرية واحدة بين العديد من المهام لمهمة واحدة، ولكن الاختيار يصبح عملية معقدة تتكون من دمج مجموعة من الخدمات المختارة لكل مهمة، لتشكيل التكوين الذي يفي بشكل أفضل بمعايير QoS للمستخدم. تكمن المشكلة في اختيار خدمات الويب هذه في كيفية التمييز بين الخدمة التي تلبى احتياجات المستخدم والعديد من الخدمات بناءً على الاحتياجات الوظيفية و/أو غير الوظيفية.

الهدف من عملنا هو أن نكون قادرين على اختيار أفضل خدمة ويب تلبى متطلبات جودة الخدمة المتوقعة من قبل العميل.

**الكلمات المفتاحية:** خدمة الويب، اختيار خدمات الويب، نوعية الخدمة، علم الاستدلال التوحيد الأمثل، الخوارزميات الوراثة.

# Table des matières

<b>Introduction générale.....</b>	<b>2</b>
<b>Chapitre I : Les services web .....</b>	<b>5</b>
1. Introduction .....	6
2. Définition d'un service web .....	6
2.1. IBM (International Business Machines Corporation).....	6
2.2. W3C (World Wide Web Consortium) .....	7
3. L'intérêt des services web .....	7
4. Caractéristiques des services web .....	7
5. Application des services web .....	9
6. Architectures des services web .....	10
6.1. Architecture en référence .....	10
6.2. Architecture en couches .....	12
7. Les principales technologies de développement des services web.....	13
7.1. XML (eXtensible Markup Language) .....	14
7.2. SOAP (Simple Object Access Protocol).....	14
7.3. WSDL (Web Service Description Language).....	18
7.4. UDDI (Universal Description, Discovery and Integration).....	19
8. Fonctionnement des services web .....	21
9. Les avantages et les inconvénients des services web.....	22
9.1. Les avantages .....	22
9.2. Les inconvénients.....	22
10. Conclusion.....	23
<b>Chapitre II : La sélection des services web à base de QoS.....</b>	<b>24</b>
1. Introduction .....	25
2. Les propriétés d'un service web .....	25
2.1. Les propriétés fonctionnelle.....	25
2.2. Les propriétés non-fonctionnelle.....	26
3. Présentation du problème de sélection dans les services web .....	26

3.1. Définition de sélection de service web.....	26
3.2. La sélection à base de QoS.....	27
4. La qualité de service (QoS).....	28
4.1. Définition de QoS.....	28
4.2. Les critères de QoS.....	28
5. Optimisation combinatoire .....	30
5.1. Définition .....	30
5.2. Approches d'optimisation combinatoire .....	31
5.2.1. L'optimisation mono-objective .....	31
5.2.2. L'optimisation multi-objective.....	31
6. Les stratégies de sélection .....	33
6.1. La stratégie de sélection locale.....	33
6.2. La stratégie de sélection globale .....	34
7. Etat de l'art des travaux sur le problème QOSSWC.....	35
7.1. Les approches exactes .....	36
7.2. Les approches heuristiques.....	37
7.3. Les approches méta-heuristiques .....	38
7.4. Les approches basée sur les techniques de base de données .....	39
8. Conclusion.....	41
<b>Chapitre III : Les algorithmes génétiques .....</b>	<b>42</b>
1. Introduction .....	43
2. Terminologies.....	43
3. Définition des algorithmes génétiques .....	44
4. Principes et fonctionnalités .....	45
4.1. Principe de base.....	45
4.2. Fonctionnement.....	46
4.3. Caractéristiques des algorithmes génétiques.....	47
4.4. Les opérateurs génétiques .....	49
5. Conclusion.....	53
<b>Chapitre IV : La conception du système.....</b>	<b>54</b>
1. Introduction .....	55

2. Les algorithmes génétiques .....	55
2.1. Expression de la fonction objective .....	57
2.2. Données de la sélection des services web .....	59
3. Diagramme de conception.....	61
3.1. La vue statique.....	61
3.2. La vue dynamique .....	64
4. Conclusion .....	68
<b>Chapitre V : Implémentation et Test .....</b>	<b>70</b>
1. Introduction .....	71
2. Environnement de développement .....	71
2.1. Environnement matériel.....	71
2.2. Environnement logiciel (langage de programmation) .....	71
3. Présentation de l'interface de l'application.....	74
4. Expérimentation .....	78
5. Conclusion .....	81
<b>Conclusion générale .....</b>	<b>83</b>
<b>Bibliographies .....</b>	<b>85</b>

# Liste des figures

<b>Figure I.1 :</b>	Architecture de référence.....	11
<b>Figure I.2 :</b>	Architecture en couches.....	13
<b>Figure I.3 :</b>	Architecture des composantes des services web.....	13
<b>Figure I.4 :</b>	Structure de message SOAP.....	15
<b>Figure I.5 :</b>	Traitement d'un message SOAP.....	17
<b>Figure I.6 :</b>	Structure et description WSDL.....	18
<b>Figure I.7 :</b>	Structure de données de l'annuaire UDDI.....	20
<b>Figure I.8 :</b>	Fonctionnement d'un service web.....	21
<b>Figure II.1 :</b>	Les approches de sélection.....	32
<b>Figure II.2 :</b>	Sélection locale.....	34
<b>Figure II.3 :</b>	Sélection globale.....	35
<b>Figure II.4 :</b>	Approches d'optimisation du problème QOSSWC.....	36
<b>Figure II.5 :</b>	Quelques méthodes de résolution exactes.....	36
<b>Figure III.1 :</b>	Organigramme des AG.....	46
<b>Figure III.2 :</b>	Les cinq niveaux d'organisation d'une population d'un AG.....	48
<b>Figure III.3 :</b>	Principe général de l'évolution d'une population d'un AG.....	49
<b>Figure III.4 :</b>	Croisement à un point .....	51
<b>Figure III.5 :</b>	Croisement à deux points.....	51
<b>Figure III.6 :</b>	Illustration du principe de la mutation.....	52
<b>Figure IV.1 :</b>	Schéma générale d'un AG.....	56
<b>Figure IV.2 :</b>	Diagramme de cas d'utilisation.....	62
<b>Figure IV.3 :</b>	Diagramme de classes.....	63
<b>Figure IV.4 :</b>	Diagramme de séquences « Introduire les critères de sélection ».....	65
<b>Figure IV.5 :</b>	Diagramme de séquences « Définition des critères de filtrage ».....	66
<b>Figure IV.6 :</b>	Diagramme de séquences « Chargement de BDD ».....	67
<b>Figure IV.7 :</b>	Diagramme de séquences « calcul de la solution optimale ».....	68
<b>Figure V.1 :</b>	Logo du Delphi 7.....	72
<b>Figure V.2 :</b>	Page d'accueil.....	74
<b>Figure V.3 :</b>	Fenêtre principale du système .....	75
<b>Figure V.4 :</b>	Interface mise à jour BDD.....	76
<b>Figure V.5 :</b>	Chargement des données client.....	76
<b>Figure V.6 :</b>	Interface résultat de la sélection de service web.....	77
<b>Figure V.7 :</b>	Sortir de l'application.....	77
<b>Figure V.8 :</b>	Choix d'une base de données.....	78
<b>Figure V.9 :</b>	Chargement de la base de données.....	79
<b>Figure V.10 :</b>	Choix de fichier C.txt.....	79
<b>Figure V.11 :</b>	Chargement des données client.....	80
<b>Figure V.12 :</b>	Résultat de la sélection.....	80

# Liste des tableaux

<b>Tableau II.1 :</b>	Comparative entre la sélection statique et la sélection dynamique.	27
<b>Tableau II.2 :</b>	Les domaines de valeurs de critères.....	30
<b>Tableau II.3 :</b>	Quelques travaux utilisant des approches exactes.....	37
<b>Tableau II.4 :</b>	Quelques travaux utilisant des approches heuristiques.....	38
<b>Tableau II.5 :</b>	Quelques travaux utilisant des approches méta-heuristiques à base de solution unique.....	39
<b>Tableau II.6 :</b>	Quelques travaux utilisant des approches méta-heuristiques à base de population de solutions.....	40
<b>Tableau IV.1 :</b>	Un exemple de base de données.....	60
<b>Tableau IV.2 :</b>	Identificateurs des acteurs et leurs rôles .....	61
<b>Tableau IV.3 :</b>	Description de diagramme de séquences.....	64

Introduction

générale

# **Introduction générale**

## **Contexte**

Le Web a évolué pour englober diverses sources d'information accessibles 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. Les services Web sont devenus le facteur le plus significatif technologique par produit.

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 elle n'a été efficace ou n'est maintenant acceptée comme le moyen d'effectuer une recherche de service.

## **Problématique**

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 ou la composition de ceux-ci est à l'égard de l'utilisateur qui s'attend à être fourni de qualité.

## **Contribution**

Notre travail consiste à chercher et retourner une meilleure sélection des services web composites en fonction de leurs(QoS).Pour cela dans ce mémoire nous

proposons une approche qui utilise les algorithmes génétiques qui sont une abstraction de la théorie de l'évolution. L'utilisation de la théorie de l'évolution comme modèle informatique pour trouver une solution optimale peut se justifier par le fait que la théorie de l'évolution permet de rechercher la solution parmi un très grand nombre de possibilités dans un laps temps raisonnable.

## **Plan du mémoire**

Le mémoire est composé de cinq chapitres et une conclusion générale, qui sont organisés comme suit :

### **Chapitre I**

Est consacré aux notions fondamentales des Services Web où nous avons défini l'architecture SOA, l'intérêt de Service web, leurs technologies, leurs standards de base, ainsi que leurs fonctionnements et architecture. Leurs avantages et inconvénients, et ensuite nous donnons notions sur la qualité de service (QoS ou QoS en anglais : Quality of Service).

### **Chapitre II**

Ce chapitre est consacré à l'analyse et à la définition de la problématique de sélection des services web dans une composition à base de paramètres de QoS (QoSWSC). Une classification des différentes approches existantes relatives à l'optimisation du problème QoSWSC est présentée. Cette classification regroupe 4 catégories qui sont : les approches de résolutions exactes, heuristiques, méta-heuristiques et celles utilisant les techniques de bases de données.

### **Chapitre III**

Ce chapitre expose les concepts fondamentaux des algorithmes génétiques. Après une terminologie pour définir le vocabulaire employé par notre travail, ce chapitre explique le fonctionnement de l'approche et les différentes méthodes de sélection

## **Chapitre IV**

Nous avons présenté la méthode proposée, et ce en deux parties différentes. Dans la première, nous avons formulé le problème à résoudre, et nous avons exprimé notre fonction objective en fonction des critères de qualité de service (QoS), ainsi que l'introduction des algorithmes génétiques. En seconde partie, nous avons présenté une conception de notre système en utilisant certains diagrammes UML que nous avons jugé nécessaires dans notre cas.

## **Chapitre V**

Ce chapitre est consacré à l'implémentation et la réalisation de l'application qui permet la sélection multi-objective des services web à base des algorithmes génétiques.

# Chapitre I

## Les services Web

*« Most science is only high falutin' nature studies. »*

*Stephen Strauss*

## 1. Introduction

Depuis les années 70, nous remarquons un besoin de communication entre les systèmes d'information des entreprises (fournisseurs ou clients). Ce type de communication est appelé intégration B2B. Dans un sens technologique, l'intégration B2B se réfère à « la technologie de logiciel utilisée pour connecter n'importe quel système d'information d'une entreprise à tous ses associés de commerce » [1]. Les protocoles assurant ces besoins s'appellent protocoles B2B. Des exemples de ces protocoles sont EDI [2] et RossetaNet [3].

Avec l'essor du Web qui a eu dans les dernières années, il a surgi le besoin de permettre qu'une application cliente invoque un service d'une application serveur en utilisant l'Internet. Ce besoin a été l'origine de ce qui se connaît comme services Web. En tenant en compte que les services Web permettant de connecter des applications différentes, l'utilité de cette technologie devient évidente et importante. Pour cette raison les activités de recherche et développement autour du sujet services Web ont un dynamisme très haut. Dans ce chapitre, nous décrivons les technologies liées aux services Web.

**2. Définition d'un service web :** On retrouve plusieurs définitions des services Web :

### 2.1. IBM (International Business Machines Corporation)

Un service Web [4] est un terme générique pour une fonction logicielle interopérable de machine à machine qui est hébergée à un emplacement adressable réseau.

Un service Web possède une interface qui masque les détails de l'implémentation afin de pouvoir être utilisé indépendamment de la plateforme matérielle ou logicielle sur laquelle il est implémenté et indépendamment du langage de programmation dans lequel il est écrit. Cette

indépendance encourage les applications basées sur les services Web à être implémentées de manière lâche, orientées composants et multi-technologies.

Les services Web peuvent être utilisés seuls ou avec d'autres services Web pour effectuer une agrégation complexe ou une transaction commerciale.

## **2.2. W3C (World Wide Web Consortium)**

Un service Web [5] est un système logiciel conçu pour prendre en charge l'interaction interopérable de machine à machine sur un réseau. Il a une interface décrite dans un format exploitable par machine (spécifiquement WSDL<sup>1</sup>). D'autres systèmes interagissent avec le service Web d'une manière prescrite par sa description à l'aide de messages (SOAP<sup>2</sup>), généralement transmis à l'aide des protocoles Internet (HTTP<sup>3</sup>) avec une sérialisation XML<sup>4</sup> conjointement avec d'autres normes liées au Web.

## **3. L'intérêt des services web**

Les services Web comportent de nombreux avantages, ils sont utilisables à distance via n'importe quel type de plate-forme, ils peuvent servir au développement d'applications distribuées et sont accessibles depuis n'importe quel type de clients. Les services Web appartiennent à des applications capables de collaborer entre elles de manière transparente pour l'utilisateur.

## **4. Caractéristiques des services web**

Les services Web ont les caractéristiques comportementales spéciales suivantes [6] :

---

<sup>1</sup>WSDL: Web Service Description Language.

<sup>2</sup> SOAP: Simple Object Access Protocol.

<sup>3</sup> HTTP: Hyper Text Transfer Protocol.

<sup>4</sup> XML: eXtensible Markup Language.

**-Un Web service est une application logicielle qui est reconnue par un URI :**

URI est la façon d'identifier un point de contenu sur le Web comme un document tel qu'un texte, audio ou vidéo. L'URI le plus connu est l'adresse d'une page Web. Le Web service est donc accessible en spécifiant son URI, c'est-à-dire que le Web service est caractérisé par un seul objet et une seule fonctionnalité.

A partir de cela, on peut faire la construction d'une application logicielle très large comportant plusieurs fonctionnalités, afin de sélectionner les fonctionnalités qui sont recherchées par les URI spécifiques.

**-Basés sur XML :** Les services Web utilisent XML au niveau des couches de représentation et de transport des données. L'utilisation de XML élimine toute liaison réseau, système d'exploitation ou plate-forme. Les applications basées sur les services Web sont hautement interopérables à leur niveau de base.

**-Faiblement couplés :** Un consommateur d'un service Web n'est pas directement lié à ce service Web. L'interface du service Web peut changer au fil du temps sans compromettre la capacité du client à interagir avec le service. Un système étroitement couplé implique que les logiques clients et serveur sont étroitement liées l'une à l'autre, ce qui implique que si une interface change, l'autre doit être mise à jour. L'adoption d'une architecture faiblement couplée a tendance à rendre les systèmes logiciels plus faciles à gérer et permet une intégration plus simple entre les différents systèmes.

**-Capacité à être synchrone ou asynchrone :** Synchroniser fait référence à la liaison du client à l'exécution du service. Dans les appels synchrones, le client bloque et attend que le service termine son opération avant de continuer. Les opérations asynchrones permettent à un client d'appeler un service, puis d'exécuter d'autres fonctions. Les clients asynchrones récupèrent leur résultat ultérieurement, tandis que les clients synchrones reçoivent leur

résultat lorsque le service est terminé. La capacité asynchrone est un facteur clé pour activer les systèmes à couplage lâche.

- **Prend en charge l'échange de documents** : L'un des principaux avantages de XML est sa façon générique de représenter non seulement les données, mais aussi les documents complexes. Ces documents peuvent être aussi simples que représenter une adresse actuelle, ou ils peuvent être aussi complexes que représenter un livre entier ou une demande de devis (RFQ<sup>5</sup>). Les services Web prennent en charge l'échange transparent de documents pour faciliter l'intégration de l'entreprise.

## 5. Applications des services web

Les technologies des services Web peuvent être appliquées à toutes sortes d'applications auxquelles elles offrent des avantages considérables en comparaison aux anciennes API<sup>6</sup> propriétaires, aux implémentations spécifiques à une plate-forme et à quelques autres restrictions classiques que l'on peut rencontrer (multi-plateforme, multi-langage, disponible sur Internet avec une information actualisée disponible en temps réel, ...).

L'application des services Web est multiple, autant dans les domaines du **B2C**, **B2B** que pour des domaines de gestion, par exemple gestion de stock, gestion commerciale, etc...

- **B2C (Business to Consumer)** : Qualifie une application, un site Internet destiné au grand public.
- **B2B (Business to Business)** : Qualifie une application, un site Internet destiné au commerce de professionnel à professionnel.

Les services Web peuvent être utiles dans la plupart des scénarios applicatifs lorsque la communication peut être établie sur un modèle bidirectionnel

---

<sup>5</sup> RFQ: Request For Quotation.

<sup>6</sup>API: Application Programming Interface.

(requête/réponse). C'est néanmoins loin d'être aussi limitatif, beaucoup d'autres modèles peuvent avoir recours aux services Web, sans même que vous vous en rendiez compte. Les entreprises qui mettent à disposition leurs services Web permettent aux développeurs intéressés par ses fonctionnalités de les réutiliser sans avoir à les recoder. Le principe des services Web permet d'avoir un partage des fonctionnalités et facilite grandement le développement [7].

## 6. Architecture des services web

### 6.1. Architecture de référence

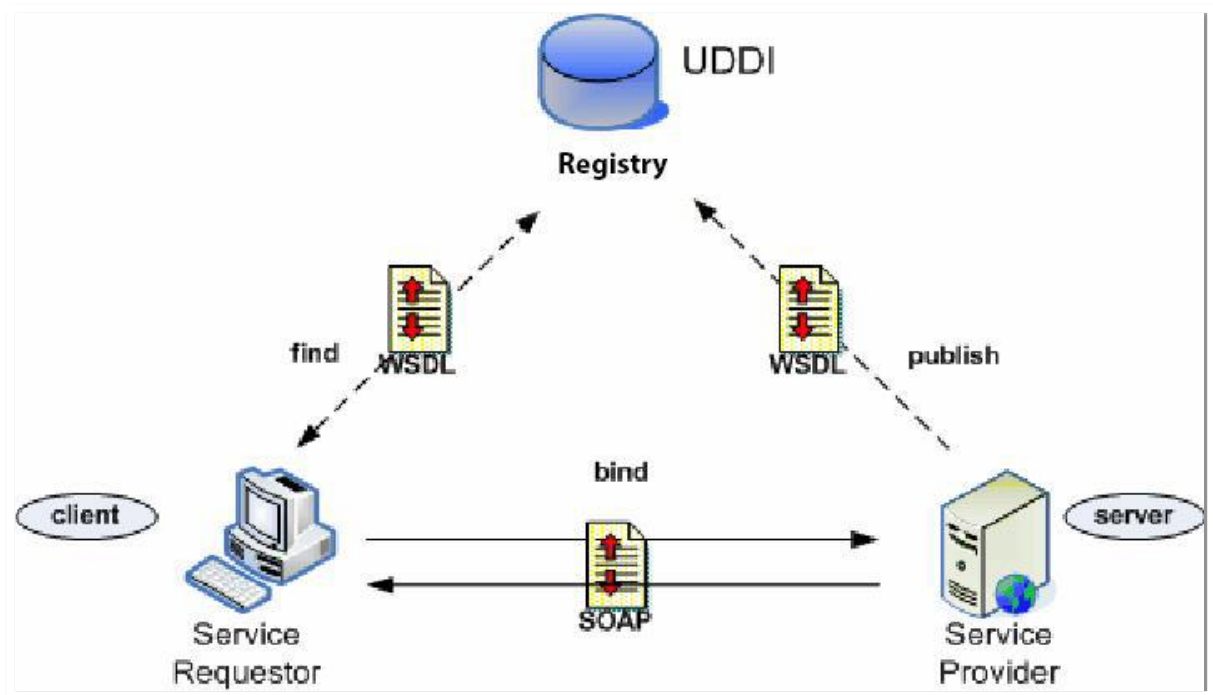
Pour promouvoir l'interopérabilité et l'extensibilité du paradigme des services Web, une architecture de base est nécessaire afin de préserver les objectifs initiaux visés par les services Web lors des évolutions technologiques successives.

L'architecture orientée services [8] est un modèle architectural dans la conception de logiciels informatiques dans lequel les composants d'application fournissent des services à d'autres composants via un protocole de communication, généralement sur un réseau.

Cette architecture s'articule autour des trois rôles suivants :

- **Fournisseur du service** : Celui qui crée le service Web et publie toutes ses fonctionnalités dans l'annuaire du service.
- **Le client** : Le consommateur accède à l'annuaire pour rechercher le service web dont il a besoin et avec lui la normalisation à obtenir.
- **L'annuaire** : Il désigne l'entité logicielle qui joue le rôle de l'intermédiaire entre les consommateurs et les fournisseurs de services qui met à disposition les interfaces d'accès aux services et donnant le contrat et l'architecture utilisés pour permettre les interactions.

Les interactions basiques entre ces trois rôles sont illustrées par la (figure I.1)



*Figure I.1 : Architecture de référence [9].*

**-Publication (publish) :** Un fournisseur informe [8] l'annuaire (registre des services) de l'existence du service Web en utilisant l'interface de publication de l'annuaire pour rendre le service accessible aux clients

**-Recherche (find) :** Le demandeur consulte l'annuaire (le registre) pour localiser un service Web publié.

**-Invocation (bind) :** Avec les informations (description) qu'il a obtenues de l'annuaire (registre de services) sur le service Web, le demandeur peut appeler le service Web.

Les services Web reposent sur une collection de normes de protocoles appelées Web Services Protocol Stack et qui contient entre autre :

- **SOAP [8] :** est connu comme un protocole de messagerie indépendant du transport. Il est basé sur le transfert de données XML sous forme de messages SOAP.

- **WSDL [10]** : est un standard qui décrit un service web sous forme d'une description en XML (fichier). Il spécifie l'emplacement du service et les méthodes du service.
- **UDDI <sup>7</sup>[6]** : est une norme basée sur XML pour décrire, publier et rechercher des services Web.

## 6.2. Architecture en couches

Une architecture étendue [11] est constituée de plusieurs couches se superposant les unes sur les autres, d'où le nom de pile des Web services. La figure 1.2 décrit un exemple d'une telle pile. La pile est constituée de plusieurs couches, chaque couche s'appuyant sur un standard particulier. On retrouve, au-dessus de la couche de transport, les trois couches formant l'infrastructure de base décrite précédemment. Ces couches s'appuient sur les standards émergents SOAP, WSDL et UDDI.

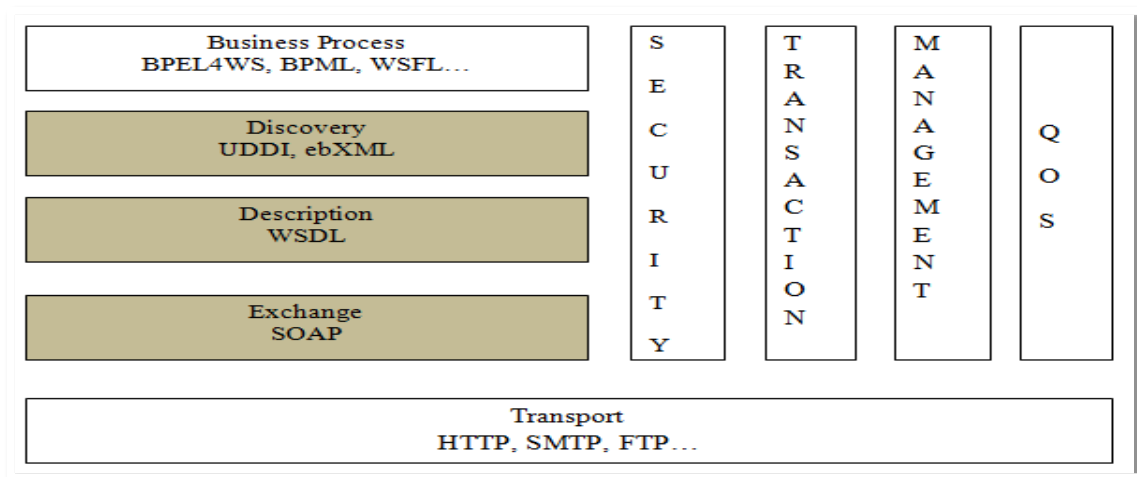
Comme on a mentionné précédemment, l'infrastructure de base définit les fondements techniques permettant de rendre les business processus accessibles à l'intérieur d'une entreprise et au-delà même des frontières d'une entreprise. Dans ce contexte deux types de couches permettent de la compléter :

- **Les couches dites transversales** : (Sécurité, administration, transactions et qualité de services (QoS<sup>8</sup>) rendent viable l'utilisation effective des Web services dans le monde industriel.
- **Une couche Business processus** : permet l'utilisation effective des Web services dans le domaine du e-business.

---

<sup>7</sup> UDDI: Universal Description, Discovery and Integration.

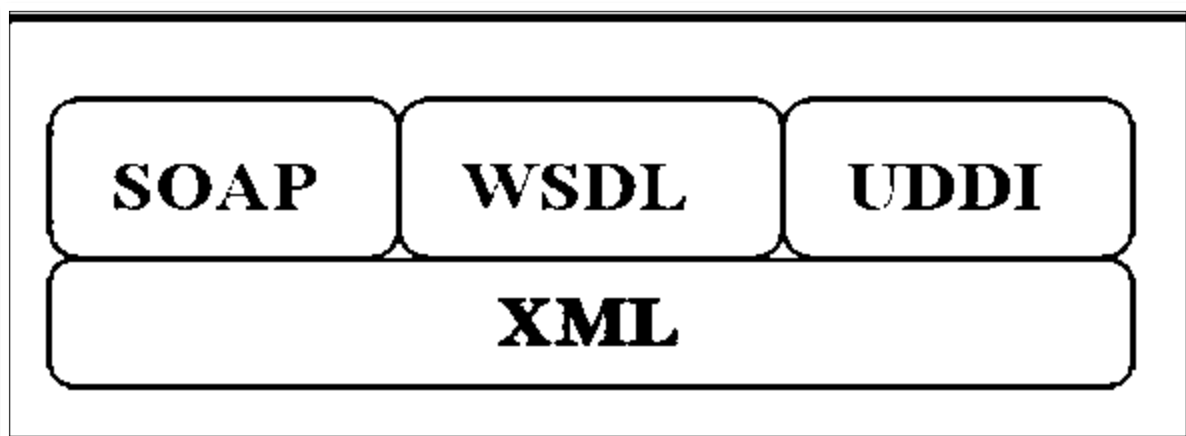
<sup>8</sup>QoS: Quality Of Service.



*Figure I.2 : Architecture en couches [14].*

## 7. Les principales technologies de développement des services web

Les services Web emploient un ensemble de technologies qui ont été conçues afin de respecter une structure en couches sans être dépendante de façon excessive de la pile des protocoles.



*Figure I.3 : Architecture des composants des services web.*

On présente les technologies utilisées pour les services web illustré dans la Figure I.3 Comme suit :

## 7.1. XML (eXtensible Markup Language)

XML [5] est un ensemble de règles de formatage pour composer des messages valides. XML est un langage de balisage extensible qui a été mis au point par le XML Working Group sous l'égide du World Wide Web Consortium (W3C) en 1996, il est né de la tentative de mettre SGML<sup>9</sup> sur le Web.

La première spécification de XML est apparue en février 1998 et se concentre sur les données, contrairement à HTML<sup>10</sup> (par exemple) qui focalise sur la présentation. XML permet donc de transformer Internet d'un univers d'information et de présentation des sites Web statiques à un univers Web programmable et dynamique, centré sur les données. Comme il est indépendant des plates-formes informatiques, il est lisible par l'humain mais est destiné à être lu par la machine. Il est flexible en ce sens qu'on peut définir d'autres langages à partir d'XML.

### ➤ XML\_RPC

XML-RPC est un protocole simple utilisant XML pour effectuer des messages RPC. Les requêtes sont écrites en XML et envoyées via HTTP POST. Les requêtes sont intégrées dans le corps de la réponse HTTP. XML-RPC est indépendant de la plate-forme, ce qui lui permet de communiquer avec diverses applications.

## 7.2. SOAP (Simple Object Access Protocol)

SOAP [5] est une spécification de communication entre services web, par échange de messages en XML au travers du web. Simple et facile à

---

<sup>9</sup>SGML: Standard Generalized Markup Language.

<sup>10</sup>HTML: Hyper Text Markup Language.

implémenter dans les serveurs web ou dans les serveurs d'application, SOAP est indépendant des langages de programmation ou des systèmes d'exploitation employés pour l'implémentation des services web.

SOAP est un protocole de communication entre applications fondé sur XML, visant à satisfaire un double objectif : servir de protocole de communication sur les intranets, dans une optique d'intégration d'applications d'entreprise, et permettre la communication entre applications et services web, en particulier dans un contexte d'échanges inter-entreprises (le réseau internet) [12].

➤ **Structure d'un message SOAP :**

Les messages échangés lors de l'utilisation du protocole SOAP sont basés sur le langage XML. Ils sont composés de deux parties, l'en-tête de protocole de transport et l'enveloppe SOAP (la Figure I.4) [13].



*Figure I.4 : Structure de message SOAP.*

a) **L'en-tête du protocole de transport** : qui dépend du protocole de transport utilisé, par exemple si le protocole HTTP est utilisé, l'en-tête contient :

- La version de protocole HTTP utilisée.
- La date de génération de message SOAP.
- Le type d'encodage du contenu (généralement de type XML).

b) **L'enveloppe SOAP** : la partie principale d'un message SOAP est l'enveloppe (symbolisée par la balise enveloppe), cette dernière est subdivisée en deux sous-parties : la partie en-tête (Header) et la partie corps du message (Body).

- L'en-tête du message SOAP : est optionnelle et extensible. Les balises XML qui permettent de symboliser cette partie sont : `<env:Header>` et `</env:Header>`. Ces balises 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 du traitement du message.

Un nœud étant un intermédiaire SOAP, incluant le récepteur et l'émetteur SOAP, désignable depuis un message SOAP. Son rôle est de traiter l'en-tête, ensuite de transférer le résultat (le message SOAP modifié) à un autre intermédiaire (qui peut être le récepteur final).

Les principaux attributs des éléments qui forment le bloc d'en-tête sont :

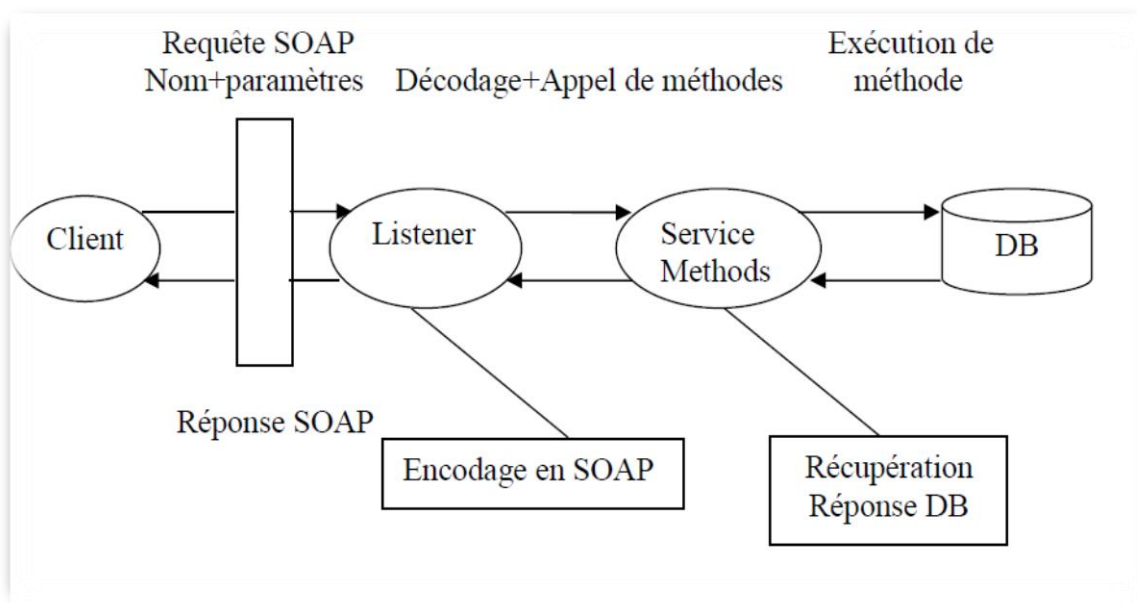
- **env:role**: permet d'indiquer à quel nœud la fonction décrite est destinée.
- **env:mustUnderstand**: c'est une valeur booléenne, elle permet de préciser que le traitement devient obligatoire pour un nœud

intermédiaire, par exemple, pour un calcul très long, il peut être utile d'envoyer un e-mail à chaque étape.

- **env:relay:** cet attribut permet de relayer un message à un autre nœud si le premier nœud n'est pas capable de le traiter.

- Le corps du message SOAP : l'élément corps du 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 (SOAP Fault).

### ➤ Fonctionnement :



*Figure 1.5 : Traitement d'un message SOAP [13].*

La requête SOAP transmise par le client à l'aide des protocoles de transmissions (exemple http).

Si l'exécution des méthodes a besoin des informations de la base de données alors il doit connecter avec la base de données.

### 7.3. WSDL (Web Service Description Language)

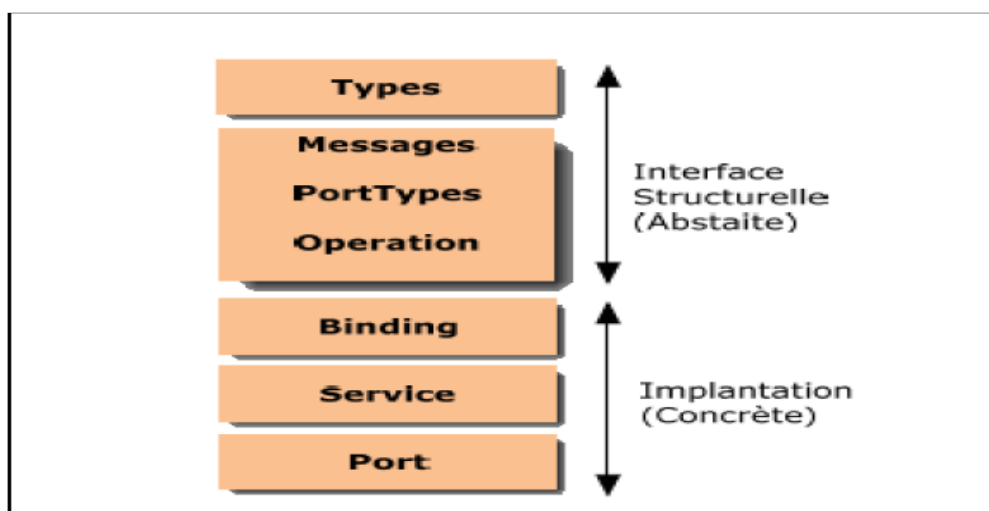
WSDL [5] a été développé conjointement par IBM, Microsoft et Ariba et a été présenté pour analyse au W3C qui l'a accepté comme une notice et publié sur leur site.

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. Il est simplement un IDL (Interface Definition Language) tout comme Corba IDL ou les interfaces de Java par exemple [10].

#### ➤ 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 en 7 types de composantes descriptives dans la définition des services réseaux. Ces éléments sont : (figure I.5) [14].



*Figure I.6 : Structure et description WSDL.*

**-Data Type Définition :** Identification du contenu et du type de données qui sont dans le message.

- Data types** : une enveloppe pour les « data type définition » utilisant des systèmes tel que XSD<sup>11</sup>.
- Message** : une définition abstraite du type de données qui est communiqué.
- **Abstract Operations** : Définition de la manière dont les données seront échangées.
- Operation** : La description abstraite d'une action supportée par le service.
- Port Types** : Un ensemble d'opérations supporté par un ou plusieurs points d'accès (port).
- Binding** : Un protocole spécifique est une spécification du format de données pour un point d'accès particulier (port).
- **Service Binding** : Définition de la couche transport qui servira au message.
- **Port** : Un point d'accès unique défini comme une combinaison de l'adresse Réseau et du point d'accès particulier (port).
- **Service** : Un ensemble de terminaisons reliées.

#### 7.4. UDDI (Universal Description, Discovery and Integration)

UDDI est un annuaire de services. Il fournit l'infrastructure de base pour la publication et la découverte des services Web. UDDI permet aux fournisseurs de présenter leurs services Web aux clients. Les informations qu'il contient peuvent être séparées en trois types :

- les pages blanches qui incluent l'adresse, le contact et les identifiants relatifs au service Web.
- les pages jaunes qui identifient les secteurs d'affaires relatifs au service Web.
- les pages vertes qui donnent les informations techniques.

---

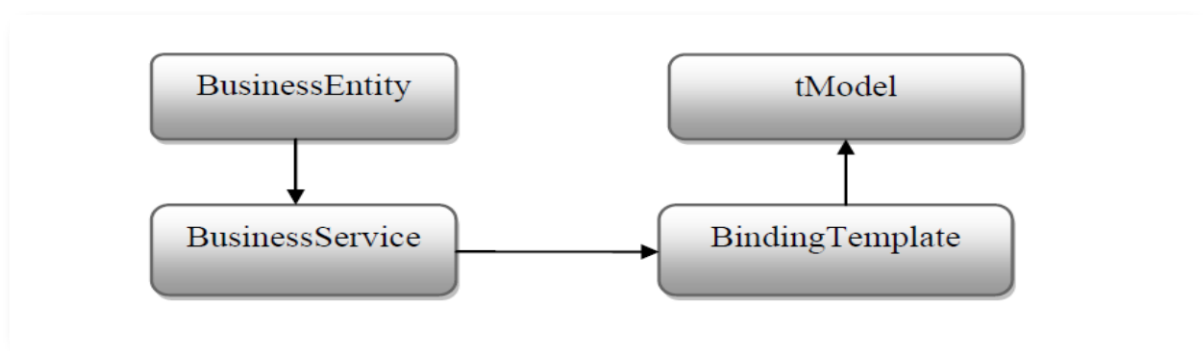
<sup>11</sup> XSD: XML Schéma Définition.

### ➤ Structure de données de l'annuaire UDDI

Selon « Arnaud » [15], Un registre UDDI se compose de quatre types de structures de données : le BusinessEntity, le BusinessService, le BindingTemplate et la Tmodel (figure I.6) [15].

Cette répartition par type fournit des partitions simples pour faciliter la localisation rapide et la compréhension des différentes informations qui constituent un enregistrement.

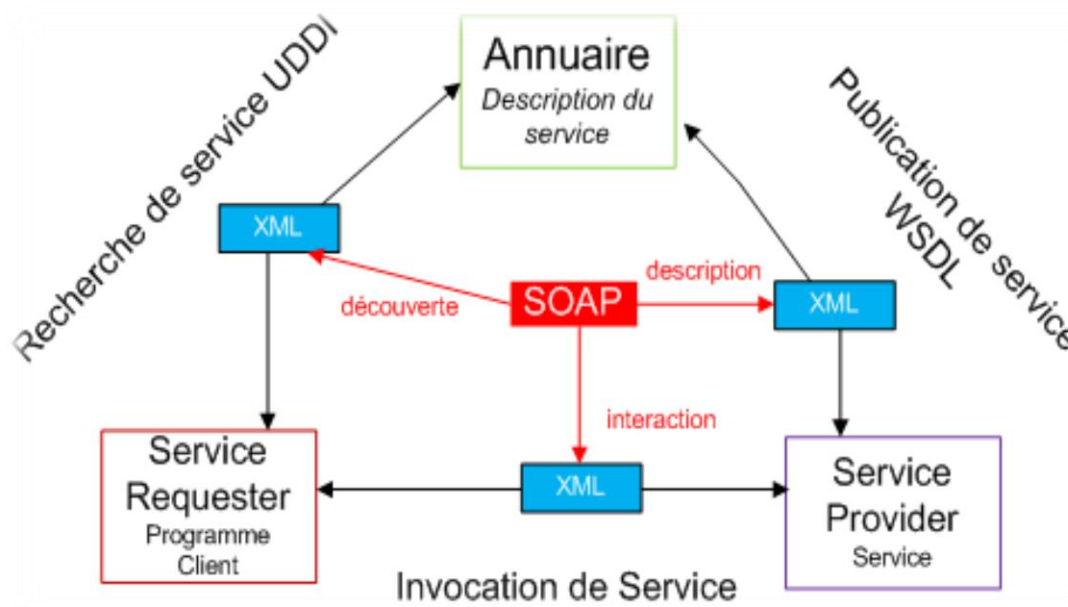
- **BusinessEntity** : contiennent des informations concernant les entreprises fournisseurs de services, la recherche se fait par contact, nom et adresse de fournisseur.
- **BusinessService** : présentent les services selon leurs fonctionnalités en suivant une taxonomie industrielle standard, le service est recherché par sujet et par domaine.
- **BindingTemplate** : offrent des informations techniques sur les services web ainsi que la référence de la description WSDL. Le service est recherché en fonction de ses caractéristiques techniques.
- **Tmodel** : ensemble d'information sur le mode d'accès au service, il peut s'agir aussi d'une spécification abstraite ou d'une taxonomie [16].



*Figure I.7 : Structure de données de l'annuaire UDDI.*

## 8. Fonctionnement des services web :

Le fonctionnement des services Web s'articule autour de trois acteurs principaux illustrés dans la (figure I.8) [17].



*Figure I.8 : Fonctionnement d'un service web.*

- 1) **Service provider service** : Le fournisseur de service met en application le service Web et le rend disponible sur Internet.
- 2) **Service requester programme client** : C'est n'importe quel consommateur du service Web. Le demandeur utilise un service Web existant en ouvrant une connexion réseau et en envoyant une demande en XML (XML-RPC, SOAP).
- 3) **Annuaire service registry** : Le registre de service est un annuaire de services. Le registre fournit un endroit central où les programmeurs peuvent publier de nouveaux services ou en trouver. Les interactions entre ces trois acteurs suivent plusieurs étapes :
  - ✓ **La publication du service** : le fournisseur diffuse les descriptions de ses services Web dans l'annuaire.

- ✓ **La recherche du service** : le client cherche un service particulier, il s'adresse à un annuaire qui va lui fournir les descriptions et les URL des services demandés afin de lui permettre de les invoquer.
- ✓ **L'invocation du service** : une fois que le client récupère l'URL et la description du service, il les utilise pour l'invoquer auprès du fournisseur de services.

## 9. Les avantages et les inconvénients d'un service web

### 9.1. 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-feu 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.

### 9.2. 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<sup>12</sup>, CORBA<sup>13</sup>, ou DCOM<sup>14</sup>.

---

<sup>12</sup> RMI : Remote Method Invocation

- En utilisant le protocole HTTP, les services Web peuvent contourner les mesures de sécurité mises en place à travers les firewalls.

## 10. Conclusion

Dans ce chapitre on a présenté quelques définitions sur les services web avec une architecture ainsi que les standards de communication entre les services web. Dans le chapitre suivant on va présenter le problème de sélection avec un exemple de motivation, en suite on va montrer un état de l'art qui regroupe les différentes approches proposées pour résoudre ce problème.

---

<sup>13</sup> CORBA: Common Object Request Broker Architector

<sup>14</sup>DCOM: Distributed Component Object Model.

# Chapitre II

## La sélection des services Web à base de QoS

*La plus grande problématique de la vie c'est de ne pas connaître ce qu'on ne connaît pas ; Et en cherchant à connaître ce qu'on ne connaît pas, on devient étudiant pour toujours.*

*Landry Makana*

## 1. Introduction

Avec la croissance explosive du nombre des services publiés sur internet, il est difficile de sélectionner des services web satisfaisants parmi les services web candidats qui offrent des fonctionnalités similaires. La qualité de service (QoS) est considérée comme le critère non fonctionnel le plus important pour la sélection des services.

Dans ce chapitre, nous présentons l'état de l'art qui introduit un problème spécifique émergeant qui est celui de sélectionner le meilleur service en respectant les contraintes de QoS. Pour cela, nous décrivons les différentes stratégies de sélection et nous citons quelques approches utilisées pour résoudre ce problème.

## 2. Les propriétés d'un service web

L'identité d'un service est définie par ses propriétés fonctionnelles et non fonctionnelles. En effet, les services se distinguent par les fonctionnalités qu'ils peuvent fournir. Deux services équivalents, c'est à dire offrant sémantiquement les mêmes fonctionnalités (exemple un service "Moisture" et un service "Humidité" qui offrent des fonctions de mesure d'humidité), peuvent être différenciés par leurs propriétés non fonctionnelles. Par exemple le service "Humidité" a un temps d'exécution inférieur au service "Moisture" alors que le service "Moisture" est caractérisé par un taux de consommation d'énergie inférieur au premier. Les propriétés non fonctionnelles sont la base de définition de la qualité de service.

### 2.1. Les propriétés fonctionnelles

Les propriétés fonctionnelles d'un service Web [18] désignent les fonctionnalités que ce service peut fournir. Ces Propriétés sont décrites dans les descriptions sémantiques des services Web qui permettent de calculer le

degré de correspondance sémantique entre la demande de l'utilisateur et une solution candidate (service composite).

## **2.2. Les propriétés non-fonctionnelles**

Les propriétés non fonctionnelles de services [18] sont les valeurs de qualité des services (coût, temps de réponse, latence, etc.). Donc ces composants du service sont utilisés pour calculer la capacité du service à fonctionner dans de bonnes conditions (en termes de ces composants).

## **3. Présentation du problème de sélection dans les services web**

Lorsqu'une application est basée sur le Web, ses tâches de composantes peuvent être effectuées à l'aide de services Web. En particulier, pour une même tâche, nous pouvons découvrir plusieurs services web capables de l'exécuter, nous serions alors confrontés à un problème de sélection de services web afin de choisir la combinaison de services qui offre la meilleure qualité de service. Cette sélection doit également prendre en compte le respect des contraintes temps réel imposées lorsqu'il s'agit d'une application temps réel.

### **3.1. Définition de sélection de service web**

La sélection de service web consiste à choisir les meilleurs services parmi un ensemble de services découverts en prenant en considération les descriptions des clients qui sont déclarées à l'aide d'une interface qui identifie leurs préférences pour qualifier le service . Nous distinguons deux manières de sélection : sélection statique et sélection dynamique.

	Sélection statique	Sélection dynamique
<b>La découverte et la sélection</b>	Manuellement en se basant les protocoles de sur découverte telle qu'UDDI.	Dynamiquement en se basant sur des représentations compréhensibles par la machine telle qu'OWLS (Web Ontology Language for Services).
<b>Les propriétés non fonctionnelles</b>	L'utilisateur qui les décrit.	Des propriétés du service mesurées ou estimées durant son exécution [19].
<b>Le service sélectionné</b>	Une fois un service sélectionné on ne peut pas le modifier (le remplacer) par la suite si de nouvelles exigences apparaissent.	Le service sélectionné est choisi une seule fois, puis lorsque de nouvelles exigences apparaissent, un autre service peut être sélectionné [20].

*Tableau II.1 : tableau comparative entre la sélection statique et la sélection dynamique.*

### 3.2. La sélection basée sur la Qos

Dans une sélection de services web basée sur la Qos deux cas se présentent :

- ✓ Si la réponse à la requête d'un client exige la sélection d'un seul service web non composite, alors la sélection est très simple.

Le candidat qui présente la meilleure Qos sera désigné pour répondre à la demande du client.

- ✓ Si la réponse à la requête d'un client exige la combinaison de plusieurs services existants, alors la sélection dans ce cas sera plus complexe du fait qu'il faut choisir la combinaison des services composants qui répond mieux aux besoins du client.

La deuxième éventualité émet une hypothèse de sélection des service web composites selon la description non fonctionnelle. La modélisation de cette hypothèse dépend de la nature de la stratégie de sélection.

## 4. La qualité de service (Qos) :

### 4.1. Définition de Qos :

La qualité de service (QoS) [21] est un ensemble de propriétés et de caractéristiques d'une entité ou d'un service qui lui confère l'aptitude à satisfaire les besoins déclarés ou implicites. Ces besoins peuvent être liés à des paramètres tels que l'accessibilité, la disponibilité, le temps de réponse, le coût, la fiabilité, etc. Ces paramètres peuvent être alors considérés comme un critère de choix lorsqu'on a affaire à sélectionner parmi plusieurs services Web découverts ceux qui respectent au mieux les contraintes imposées par l'utilisateur.

### 4.2. Les critères de Qos :

Cinq critères de qualité génériques pour des services élémentaires sont considérés :

- **Latence** : Latence mesure le retard prévu en secondes entre le moment où l'exécution commence et moment où l'exécution finit. La latence est calculée par :

$$\text{Latence (Sol)} = \sum_{i=1}^n \text{lat}(\text{Ci})$$

- **Coût** : le prix d'exécution est la somme d'argent qu'un demandeur de service doit payer pour exécuter une opération d'un service. Les allocataires de service annoncent directement le prix d'exécution de leurs opérations.

$$\mathbf{Coût (sol)} = \sum_{i=1}^n \mathbf{coût(Ci)}$$

- **Réputation** : la réputation d'un service est une mesure de sa fidélité. Elle dépend principalement des expériences de l'utilisateur d'employer ce service.

Les différents utilisateurs peuvent émettre des avis différents sur le même service. La valeur de la réputation est définie comme rang moyen indiqué au service par des utilisateurs.

$$\mathbf{Rep (Sol)} = 1/n \sum_{i=1}^n \mathbf{rep(Ci)}$$

- **Disponibilité** : La disponibilité d'un service est la probabilité que le service est accessible.

$$\mathbf{Disp (Sol)} = \log_2 \prod_{i=1}^n \mathbf{disp (Ci)}$$

- **Fiabilité** : La fiabilité d'un service est la probabilité qu'une demande est correctement répondue dans le délai de temps prévu maximum (ce critère calcule le taux d'erreur). La fiabilité est une mesure liée à la configuration de matériel et/ou de logiciel des services et aux connexions réseau entre les demandeurs de service et les fournisseurs.

$$\mathbf{Fiab (Sol)} = \log_2 \prod_{i=1}^n \mathbf{fiab (Ci)}$$

On suppose que les domaines de valeurs de critères de QoS sont définis comme suit :

<b>Critères</b>	<b>Domaines de valeurs</b>
Coût	0 ..3000 Dinars
Temps d'exécution	0..300s
Réputation	0..5
Disponibilité	0 ..1
Fiabilité	0..1

*Tableau II.2 : Les domaines de valeurs de critères.*

## **5. Optimisation combinatoire :**

### **5.1. Définition**

L'optimisation combinatoire est une branche des mathématiques et de l'informatique qui étudie les problèmes d'optimisation où la solution recherchée est une combinaison, ou une configuration, d'éléments. Ces problèmes sont souvent caractérisés par un grand nombre de possibilités, ce qui rend la recherche exhaustive de la solution impossible dans un temps raisonnable.

Un problème d'optimisation combinatoire consiste à trouver la meilleure solution dans un ensemble discret de solutions appelé ensemble des solutions réalisables. En général, cet ensemble est fini mais de cardinalité très grande.

Il s'agit, en général, de maximiser (problème de maximisation) ou de minimiser (problème de minimisation) une fonction objectif sous certaines contraintes. Le but est de trouver une solution optimale dans un temps d'exécution raisonnable.

Un problème d'optimisation en général est défini par un espace de recherche  $S$  et une fonction objective  $f$ . le but est de trouver la solution  $s^* \in S$  de meilleure qualité  $f(s^*)$ . Suivant le problème posé, on cherche soit le minimum, soit le maximum de la fonction  $f$  [22].

## 5.2. Approches d'optimisation combinatoire

Plusieurs efforts de recherches ont été rapportés dans la littérature sur la sélection automatique de service [22]. Les travaux existants sont scindés en deux types [23] :

**5.2.1. L'optimisation mono-objective :** comporte plusieurs approches, elle peut employer un modèle de sélection global ou un modèle de sélection local.

-Le modèle de sélection global se concentre sur toute la composition, il peut obtenir la solution optimale, mais il a une complexité exponentielle.

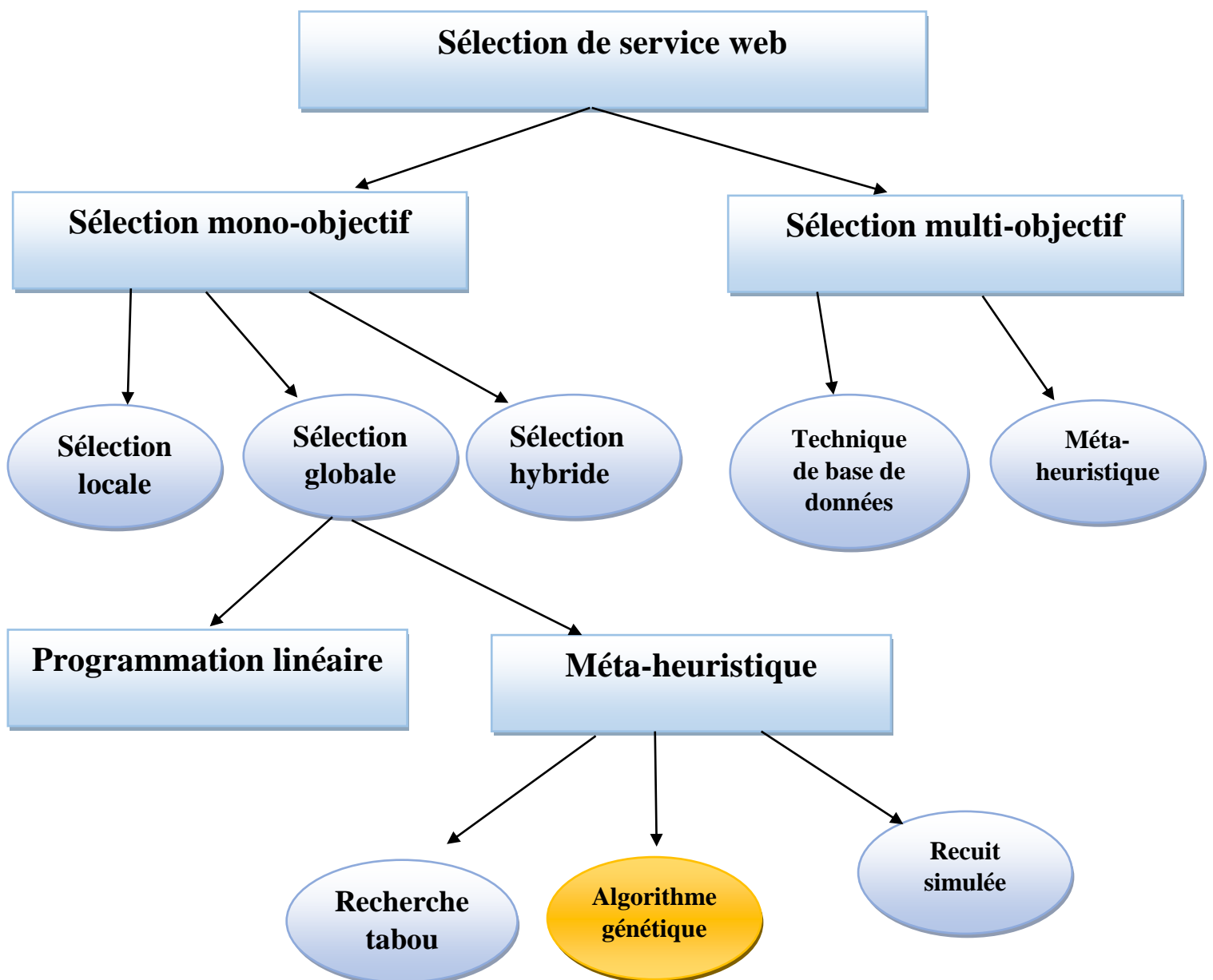
-Néanmoins, le modèle local a seulement une complexité linéaire mais ne peut pas manipuler les contraintes globales (il manipule seulement des contraintes locales).

-Le troisième modèle est un compromis des deux approches, il commence la recherche par une optimisation globale, puis il continue le travail avec une optimisation locale, sa complexité temporelle est inférieure par rapport à l'optimisation globale, et il peut également manipuler les contraintes globales.

**5.2.2. L'optimisation multi-objective :** L'optimisation multi objectif est née du besoin en industrie de satisfaire plusieurs critères contradictoires, simultanément. Les bases de cette optimisation ont été posées par Pareto et Edgeworth au 19<sup>ème</sup> siècle. Ses théories trouvent leurs premières applications en économie et ces dernières années dans les sciences pour l'ingénieur. L'optimisation mono objectif consiste à maximiser (ou minimiser) une seule fonction

objectif par rapport à un ensemble de paramètres. Or, la plupart des problèmes d'optimisation réels sont décrits à l'aide de plusieurs objectifs souvent contradictoires devant être optimisés à l'aide de plusieurs objectifs souvent contradictoires devant être optimisés simultanément. La solution optimale correspond alors aux meilleurs compromis possibles permettant de résoudre le problème. Il s'agit de l'optimisation multi objectifs, qui fournit aux décideurs un ensemble de solutions optimales.

La figure suivante montre l'état de l'art des différents travaux effectués sur le problème de sélection de service web



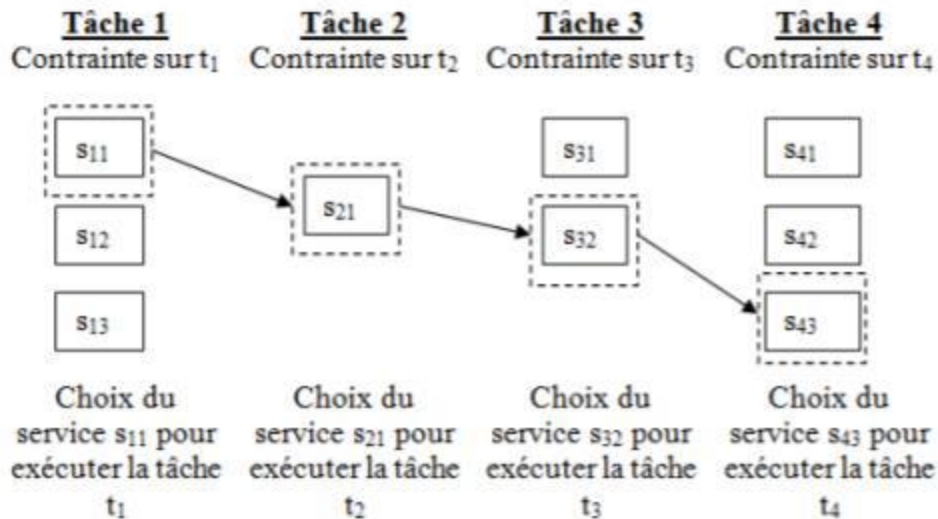
## **6. Les stratégies de sélection :**

Dans la littérature, il existe deux stratégies de sélection de services web [24] : Une stratégie de sélection locale et une stratégie de sélection globale. Chaque stratégie est définie en fonction de la nature des contraintes de QoS imposées.

En fait, ces contraintes n'ont pas un caractère obligatoire, mais permettent de définir les limites de fonctionnement d'un service web composant ou composite.

### **6.1. La stratégie de sélection locale :**

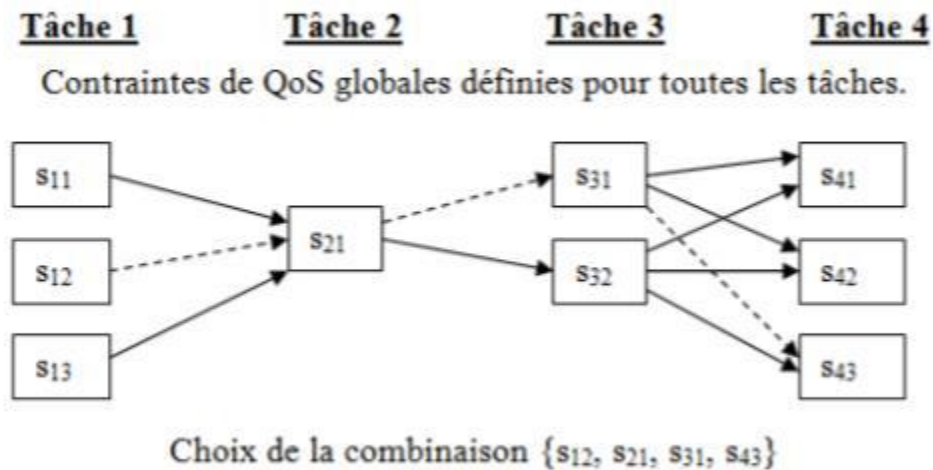
Elle a pour objectif de choisir le meilleur Web service pour chaque tâche individuelle [25]. Dans cette approche [26] la sélection du service Web qui exécutera une tâche donnée d'une spécification de service composite se fait au dernier moment possible et sans tenir compte des autres tâches impliquées dans le service composite. Lorsqu'une tâche doit réellement être exécutée, le système collecte des informations sur la qualité de service de chacun des services Web qui peuvent exécuter cette tâche. Après avoir collecté ces informations de QoS, un vecteur de qualité est calculé pour chacun des services Web candidats, et sur la base de ces vecteurs de qualité, le système sélectionne l'un des services Web candidats. La figure II.2 décrit cette stratégie de sélection.



*Figure II.2 : sélection locale [25].*

## 6.2. La stratégie de sélection globale :

La stratégie de sélection globale [27] a pour but de choisir la combinaison de services web qui garantit la meilleure qualité globale en tenant compte des contraintes de QoS et des préférences globales assignées pour l'ensemble des tâches. Pour le cas d'une application temps réel, il s'agit de sélectionner la combinaison de services qui offre la meilleure QoS et qui respecte les contraintes temps réel et d'autres préférences globales imposées pour l'ensemble des tâches de l'application comme par exemple imposer que la durée d'exécution de bout en bout doit être inférieure à une échéance. La figure II.3 décrit la stratégie de sélection globale.



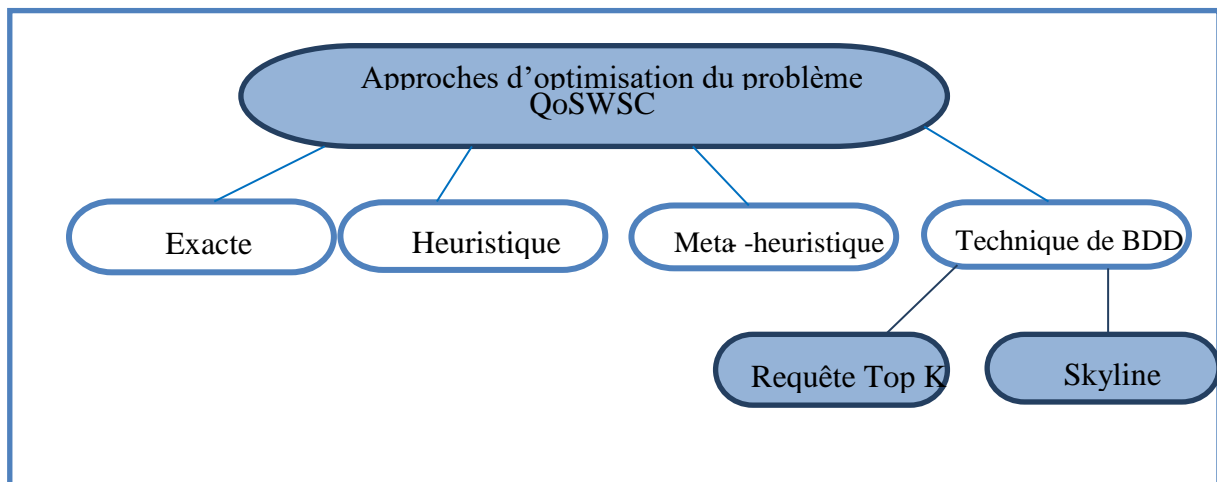
*Figure II.3 : la sélection globale [25].*

## 7. Etat de l'art des travaux sur le problème QoSSWC

Le but de la sélection des services web est de trouver le service ou les services qui satisfassent les besoins de l'utilisateur et pour cela plusieurs travaux ont vu le jour pour résoudre ce problème. Ces travaux sont classés en 3 grandes classes distinctes selon leur stratégie, paramètre utilisé et résultat obtenu.

Dans cet état de l'art, nous nous concentrons sur les travaux d'optimisation de ce problème (QoSWSC) de premier rang, 4 catégories peuvent être considérées dans la résolution du problème de genre : [28]

1. Les approches Exactes.
2. Les approches Heuristiques (approximatives).
3. Les approches Méta-heuristiques (approximatives).
4. Les approches Techniques de bases de données.

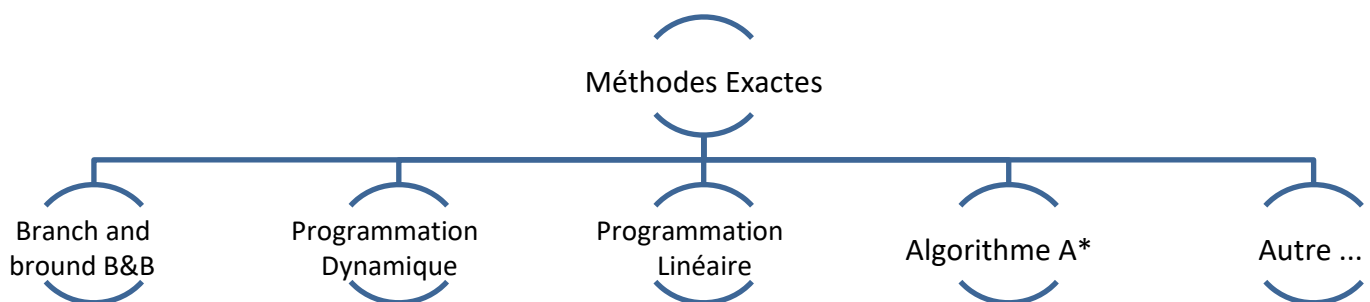


*Figure II.4 : Approches d'Optimisation du Problème QOSWSC.*

### 7.1. Les approches exactes :

Toujours dans l'obtention de la solution optimale d'un problème traité, les méthodes exactes assurent cet objectif avec un parcours complet de l'ensemble de l'espace de recherche afin de retirer toutes les solutions qui ne peuvent pas être bien meilleures que la solution optimale courante.

La figure suivante montre quelques méthodes exactes :



*Figure II.5 : Quelques méthodes de résolution exactes.*

Le Tableau qui suit contient quelques exemples des travaux utilisant des approches exactes concernant la sélection des services web avec explication.

Travaux	Explication
(Yu T. a.-J., 2005a) [29] et (Liu, 2012) [30]	présentent une méthode en considérant de multiples critères QoS en utilisant l'algorithme branch and bound et traitent différents types de work-flows.
(Yu T. a.-J., 2005b) [31]	proposent une méthode qui permet de satisfaire les contraintes globales, et ont considéré le problème comme un problème de sac à dos à choix multiples. Ils ont utilisé l'algorithme branch and bound pour retourner le résultat.
(Rodriguez-Mier, 2010) [32]	utilisent l'algorithme A*

**Tableau II.3 : Quelques travaux utilisant des approches exactes concernant la sélection des services web.**

L'inconvénient de ces approches exactes est la taille du problème traité, l'augmentation des variables (les tâches) aboutit à l'explosion de l'espace de recherche.

## 7.2. Les approches heuristiques :

Une méthode heuristique est une méthode de résolution de problème qui ne s'appuie pas sur une analyse détaillée ou exhaustive du problème. Elle consiste à fonctionner par approches successives en s'appuyant, par exemple, sur des similitudes avec des problèmes déjà traités afin d'éliminer progressivement les alternatives et ne conserver qu'une série limitée de solutions pour tendre vers celle qui est optimale. Les méthodes heuristiques

prennent beaucoup moins de temps pour trouver la solution optimale par rapport aux Méthodes exactes.

Travaux	Explication
(Moustafa, 2013) [33] et (Feng, 2013) [34]	ils utilisent les algorithmes d'apprentissage par renforcement pour trouver l'ensemble de solutions Pareto optimales qui satisfait les facteurs QoS multiples et les exigences de l'utilisateur.
(Li, 2012)[35]	adoptent la stratégie gagnant-gagnant (winwin stratégie) et proposent un algorithme qui porte le nom de la stratégie elle-même.
(Lecue, 2009) [36], (Klein, 2011) [37]	utilisent l'algorithme hill-climbing (Méthode de Descente) pour réduire la complexité de temps de calcul et comparent leur méthode avec celle utilisant LIP.

*Tableau II.4 : Quelques travaux utilisant des approches heuristiques concernant la sélection des services web.*

### 7.3. Les approches méta-heuristiques :

Les méta-heuristiques sont généralement des algorithmes stochastiques itératifs, qui progressent vers un optimum global, c'est-à-dire l'extremum global d'une fonction, par échantillonnage d'une fonction objectif. Elles se comportent comme des algorithmes de recherche, tentant d'apprendre les caractéristiques d'un problème afin d'en trouver une approximation de la meilleure solution (d'une manière proche des algorithmes d'approximation).

Les méta-heuristiques se partagent en deux classes :

### 7.3.1. Les méta-heuristiques à base de solution unique :

Cette catégorie initialise la recherche avec une solution unique puis elle l'améliore au cours d'une série d'itérations en se basent sur la notion du voisinage. La solution initiale subit des modifications selon son voisinage afin d'améliorer progressivement sa qualité.

Nous trouvons dans ces classes deux familles d'algorithmes de recherche locale :

- ✓ Pour une optimisation locale : recherche locale simple (la descente).
- ✓ Pour une optimisation globale : recuit simulé

Travaux	Explication
(Ko, 2008) [38]	proposent un algorithme de composition de service Web basé sur la satisfaction de contraintes qui combine la recherche Tabou et la méta-heuristique recuit simulé.
(Rosenberg, 2010) [39]	la méta-heuristique recuit simulé.
(Hadjila, 2012) [40]	la composition des services est modélisée sous forme d'un workflow. Les auteurs ne considèrent que la structure séquentielle et utilisent la recherche Tabou pour optimiser la fonction mono-objective qui considère 5 paramètres QoS.

*Tableau II.5 : quelques travaux utilisant des approches méta-heuristiques à base de solution unique.*

### 7.3.2. Les méta-heuristiques à base de population de solutions :

Cette catégorie initialise la recherche avec un ensemble de solutions afin d'obtenir la meilleure (solution optimale) qui est la solution du problème traité. L'utilisation d'un ensemble de solutions augmente la possibilité d'arriver à une solution de bonne qualité.

Catégories de cette classe :

- ✓ les algorithmes évolutionnaires.
- ✓ les algorithmes génétiques.
- ✓ algorithmes à base d'intelligence par essaims : Colonies de fourmis, recherche coucou.

Travaux	Explication
(Wang, 2008) [41]	les auteurs proposent un algorithme génétique en utilisant une optimisation multi-objectif, ils définissent une fonction objectif pour chaque paramètre QoS considéré qui sont au nombre de 3.
(Ming, 2007) [42]	montrent que l'optimisation en essaims particuliers PSO est plus rapide et plus fiable par rapport aux algorithmes génétiques (version standard sans hybridation), dans ce travail les auteurs modifient les services concrets des particules en changeant la vitesse avec des opérateurs tels que l'addition, la soustraction, la multiplication.

<b>(Merzoug, 2014) [43]</b>	utilisent l'optimisation par harmonie pour résoudre le problème QoSWSC.
-----------------------------	---

*Tableau II.6 : quelques travaux utilisant des approches méta-heuristiques à base de population de solutions.*

#### **7.4. Les approches basées sur les techniques de base de données :**

Cette catégorie traite la sélection comme un problème multi-objectif qu'il soit traité en local ou en global (sans le réduire à une fonction mono-objectif). Elle repose surtout sur la notion de Pareto dominance [28] et sur le principe des requêtes Skyline ou Top-k. Elle utilise des algorithmes comme celui de Divide and conquer ou autres.

### **8. Conclusion :**

Ce chapitre était une présentation et formulation du problème de sélection des services web, avec quelques travaux de résolution. Une partie état de l'art consacré aux 4 catégories 'approches de sélection de services web (Exacte, heuristique, méta-heuristique, technologie de base de données), avec quelques travaux faits dans le domaine et qui sont toujours encours d'amélioration.

# Chapitre III

## Les algorithmes génétiques

*« We cannot solve our problems with some thinking we used when we created »*

*Albert Einstein*

## 1. Introduction

Les algorithmes génétiques sont des algorithmes d'optimisation s'appuyant sur des techniques dérivées de la génétique et des mécanismes d'évolution de la nature : croisements, mutations, sélections, ....etc. Ils appartiennent à la classe des algorithmes évolutionnaires.

## 2. Terminologie

Avant d'aborder comment un problème soit résolu à l'aide d'un AG<sup>1</sup>, nous devons également définir le vocabulaire.

**Chromosome** : une séquence de gènes qui représente une solution potentielle dans l'espace de recherche.

**Gène** : une variable du problème qui peut prendre une valeur ou une autre. Les gènes sont regroupés dans un chromosome.

**Population** : un ensemble de chromosomes représentant des solutions potentielles au problème.

**Individu** : l'individu est réduit à un chromosome et on l'appelle donc chromosome ou individu pour désigner le même objet.

**Fitness** : la mesure de la qualité d'une solution par rapport à l'objectif recherché. Elle est souvent définie comme une fonction objective à maximiser ou minimiser.

**Sélection** : le processus de choix des chromosomes les plus adaptés pour la reproduction.

**Croisement (cross over)** : le processus de création d'une nouvelle génération de chromosomes en combinant les caractéristiques de deux parents

---

<sup>1</sup> AG : Algorithme Génétique

sélectionnés.

**Mutation** : le processus de modification aléatoire d'un gène pour créer de nouvelles solutions potentielles.

**Elitisme** : la stratégie consistant à maintenir les meilleurs chromosomes de la population dans les générations suivantes sans modification.

**Génération** : une itération complète de l'algorithme génétique, comprenant la sélection, le croisement, la mutation et l'évaluation de la fitness.

**Descendants** : la descendance est un terme qui fait référence aux nouveaux individus qui sont créés à partir de la reproduction de deux individus parents. Ces nouveaux individus sont souvent appelés "enfants" ou "descendants".

**Parents** : les parents correspondent aux individus pouvant s'imposer ce qui donne naissance à de nouveaux individus descendants afin de former une nouvelle génération.

**Convergence** : le processus par lequel les chromosomes les plus adaptés deviennent de plus en plus similaires au fil des générations, ce qui peut indiquer que l'algorithme génétique a atteint une solution optimale.

### **3. Définition des algorithmes génétiques**

Les algorithmes génétiques sont des méthodes d'optimisation et de recherche inspirées par le processus de sélection naturelle observé dans l'évolution biologique. Ils sont utilisés dans le domaine informatique pour résoudre des problèmes complexes où il est difficile de trouver une solution optimale de manière exhaustive.

## 4. Principes et fonctionnalités

### 4.1. Principe de base

Les algorithmes génétiques sont des approches d'optimisation qui utilisent des techniques dérivées de la science génétique et de l'évolution naturelle : la sélection, la mutation et le croisement. Pour utiliser ces approches, on doit disposer des éléments suivants [44] :

- 1) Le codage d'un élément de population : une fonction qui permet de modéliser les données du problème réel en données utilisables par l'algorithme génétique.
- 2) Une fonction pour générer la population initiale : la génération de la population initiale est importante puisque cette génération représente le point de départ de l'algorithme et son choix influe sur la rapidité et l'optimalité de la solution finale.
- 3) Une fonction à optimiser (la fonction objective) : une fonction qui retourne une valeur d'adaptation pour chaque individu. Cette valeur permet de déterminer la solution pertinente puisque le problème se restreint à chercher le groupe d'individus qui ont les valeurs optimums.
- 4) Des opérateurs qui permettent d'évoluer d'une population à une autre tout en améliorant la fonction objective. L'opérateur de croisement recompose les gènes d'individus existant dans la population, alors que l'opérateur de mutation a pour but de garantir l'exploration de l'espace d'états.
- 5) Des paramètres de dimensionnement : la taille de la population, nombre total de générations (critère d'arrêt), probabilités d'application des opérateurs de croisement et de mutation, etc.

## 4.2. Fonctionnement

### 4.2.1. L'algorithme

\* Initialisation : Générer un ensemble de solutions initiales.

\* Boucle Principale

1. Evolution :

-Sélection : Choisir avec une probabilité proportionnelle à leur qualité une liste d'individus.

-Reproduction : Générer à partir de cette liste de nouveaux individus à l'aide des opérateurs génétiques

-Remplacement : Eliminer avec une probabilité inversement proportionnelle à leur qualité certains individus.

2. Réactualisation de la meilleure solution

3. Allez en 1. tant que Nombre de Générations  $\leq$  Valeur Prédéterminée [45].

### 4.2.2. Analyse de l'algorithme

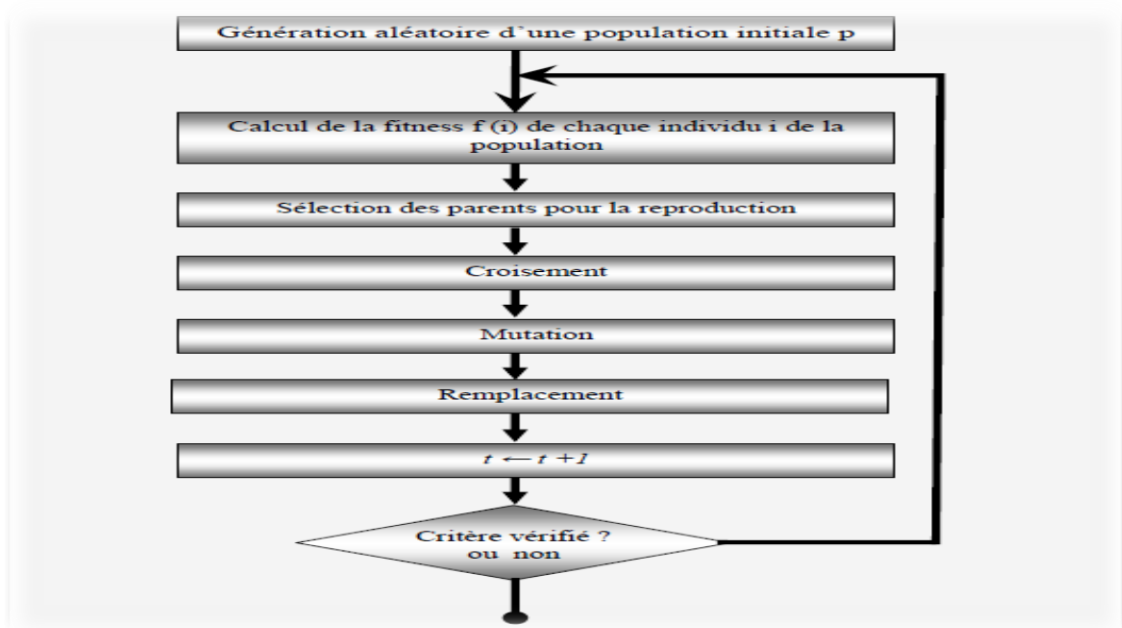


Figure III.1 : Organigramme des AG.

- 1) On génère une population aléatoire de taille de la population contrôleurs.
- 2) On évalue les contrôleurs de la génération 0 (on calcule leur fitness) en les injectant dans la grille 2D.
- 3) On sélectionne les contrôleurs qui vont servir de géniteurs.
- 4) On crée la nouvelle génération en copiant, mutant et croisant les géniteurs, selon des proportions à définir.
- 5) On évalue les contrôleurs de la nouvelle génération (on calcule leur fitness) en les injectant dans la grille 2D.
- 6) On remplace l'ancienne population par la nouvelle.
- 7) On recommence 3), 4), 5) et 6) jusqu' à ce qu'on ait atteint une fitness suffisante ou le nombre maximum de générations.

Le mode de sélection et la fitness sont à définir par l'utilisateur.

Les paramètres taille de la population, proportion de copies, proportion de mutations, proportion de cross-over, et nombre de générations, sont à choisir au lancement de l'expérimentation.

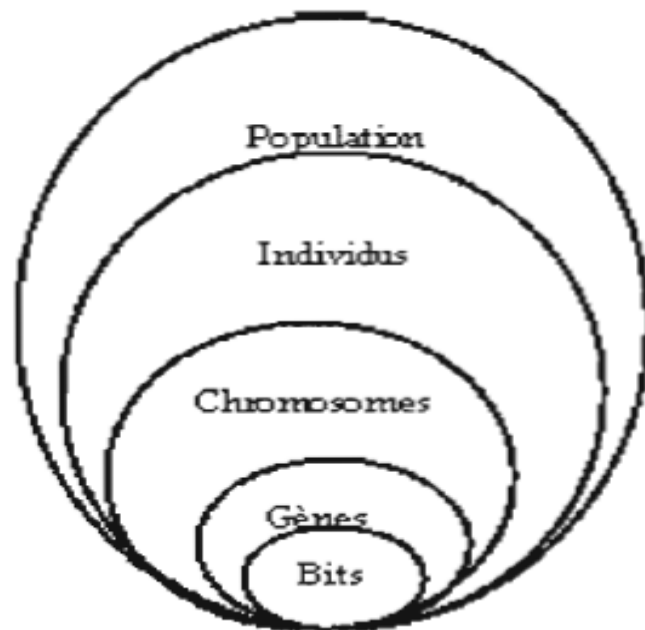
### **4.3. Caractéristiques des algorithmes génétiques**

Les algorithmes génétiques, en tant qu'approche de résolution de problèmes, se caractérisent par certains aspects particuliers: le codage des paramètres du problème à traiter, l'espace de recherche et la fonction d'évaluation qui permet de déterminer la pertinence d'une solution trouvée et l'évolution d'une génération à une autre par la sélection des chromosomes qui participent à la reproduction et les chromosomes à disparaître [46].

#### **4.3.1. Le codage**

Le codage est une fonction qui permet de passer de la donnée réelle du problème traité à la donnée utilisée par l'algorithme génétique (figure III.2). Le choix du codage est l'élément le plus important dans la conception de

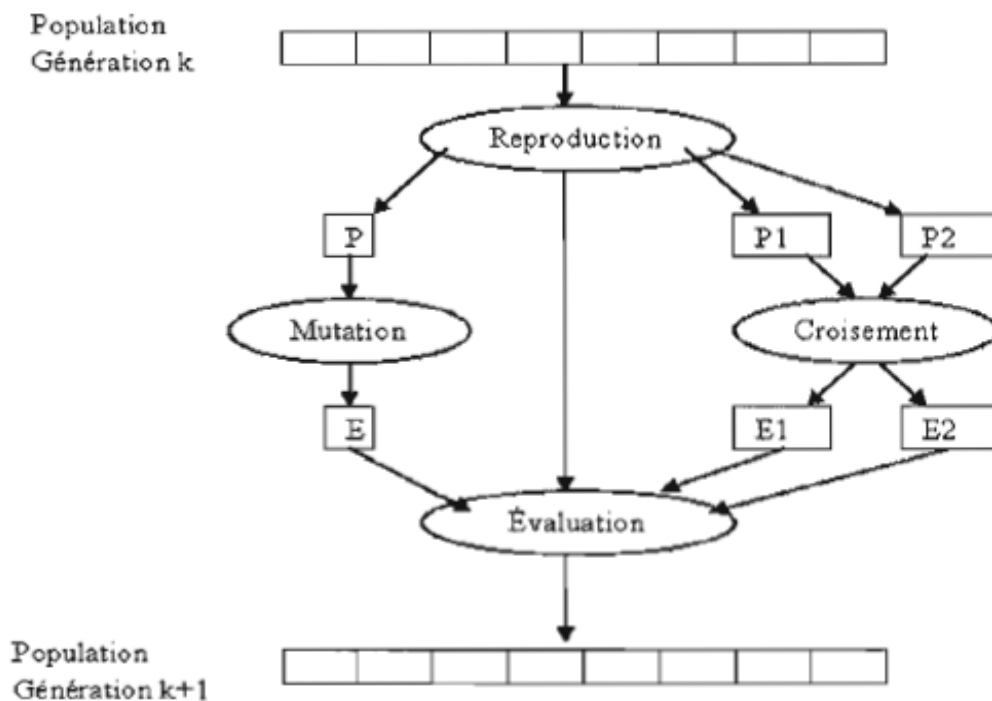
l'algorithme puisqu'il permet d'une part de représenter les données, les paramètres et les solutions et d'autre part il influe sur la mise en œuvre des opérations génétiques telles que le croisement et la mutation qui influent directement sur le bon déroulement de l'algorithme génétique et de sa convergence vers la bonne solution.



*Figure III.2 : Les cinq niveaux d'organisation d'une population d'un algorithme génétique [44].*

#### **4.3.2. La fonction d'évaluation**

La fonction de performance qu'on appelle aussi fonction d'adaptation, fonction objectif ou fonction fitness associe une valeur de performance à chaque individu ce qui offre la possibilité de le comparer à d'autres individus et permet à l'algorithme génétique de déterminer qu'un individu sera sélectionné pour être reproduit ou pour déterminer s'il sera remplacé [47]



*Figure III.3 : Principe général de l'évolution d'une population d'un algorithme génétique [48].*

#### 4.4. Les opérateurs génétiques

La reproduction est le processus qui permet de construire une population  $K+1$  à partir d'une population  $K$ . Ce processus est constitué par l'utilisation de l'opération de sélection, de l'opération de croisement ou/et de l'opération de mutation (Figure III.3).

##### 4.4.1. La sélection

L'opération de sélection permet de déterminer quels individus sont plus enclins à obtenir les meilleurs résultats [47]. On trouve deux types de sélection :

- 1) **La sélection pour la reproduction** : On l'appelle tout simplement l'opération de sélection, et elle permet de choisir les individus qui participent à une reproduction (croisement ou mutation). Cette

opération choisit, généralement, les individus les plus forts (meilleurs scores d'adaptation) pour produire les enfants les plus performants.

2) **La sélection pour le remplacement** : On l'appelle tout simplement l'opération de remplacement, et elle choisit les individus les plus faibles pour être remplacés par les nouveaux.

On trouve plusieurs techniques de sélection :

- **Sélection par roulette** : Cette méthode de sélection est basée sur la probabilité proportionnelle à la valeur de fitness de chaque individu. Les individus avec une valeur de fitness plus élevée ont une plus grande probabilité d'être sélectionnés, simulant ainsi le processus de roulette où les individus "gagnants" ont une plus grande part de la roue.
- **Sélection par rang** : Cette méthode attribue un rang à chaque individu en fonction de sa valeur de fitness par rapport à l'ensemble de la population. Les individus sont ensuite sélectionnés en fonction de leur rang plutôt que de leur valeur de fitness réelle. Cela permet de réduire l'effet de grande disparité de fitness entre les individus.
- **Sélection par tournoi** : Dans cette méthode, un certain nombre d'individus sont sélectionnés au hasard et comparés entre eux. L'individu ayant la meilleure valeur de fitness est sélectionné pour la reproduction. Le processus de tournoi est répété plusieurs fois pour sélectionner plusieurs individus.
- **Sélection par élitisme** : L'élitisme consiste à sélectionner les meilleurs individus de la génération actuelle et à les conserver intacts dans la génération suivante. Cela garantit que les meilleures solutions ne sont pas perdues au fil des générations.

#### 4.4.2. Le croisement

L'opération de croisement (*crossover*) est une opération de reproduction qui permet l'échange d'information entre les chromosomes (individus). Il utilise deux parents pour former un ou deux enfants. Les deux parents sont choisis par l'opération de sélection. Le croisement permet l'innovation (les enfants sont différents de leurs parents) et repose sur l'idée que deux parents performants produiront des enfants plus performants.

Dans le cas d'une représentation binaire, le croisement de deux chromosomes peut se faire en un seul point de coupure (Figure III.4) ou en deux points de coupure (Figure III.5).

Le taux de croisement détermine la proportion des individus qui sont croisés parmi ceux qui remplaceront l'ancienne génération.

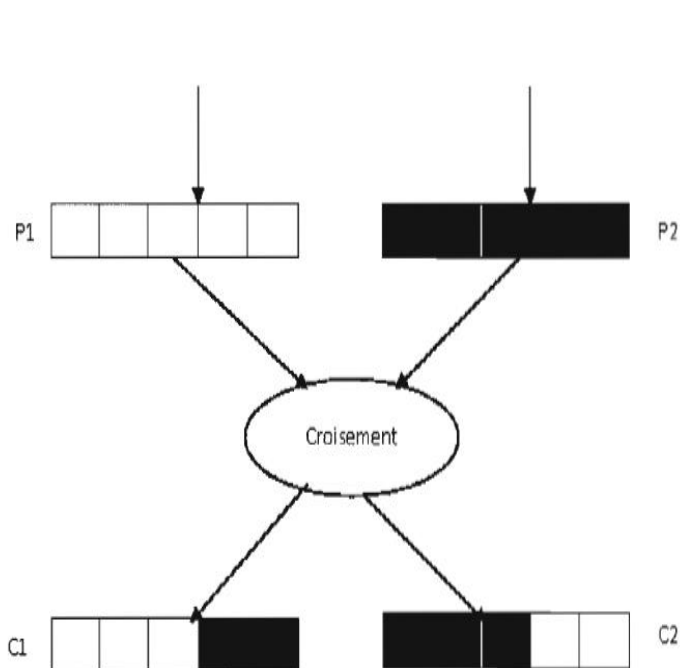


Figure III.4 : croisement a un point [48].

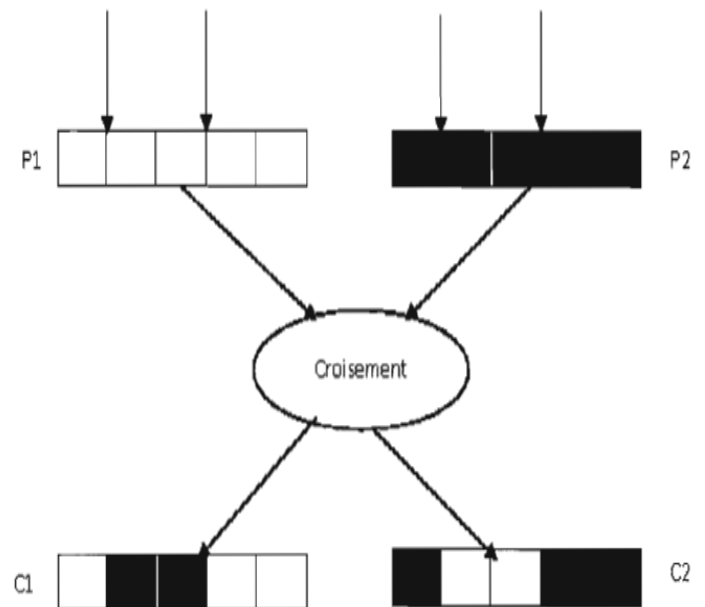
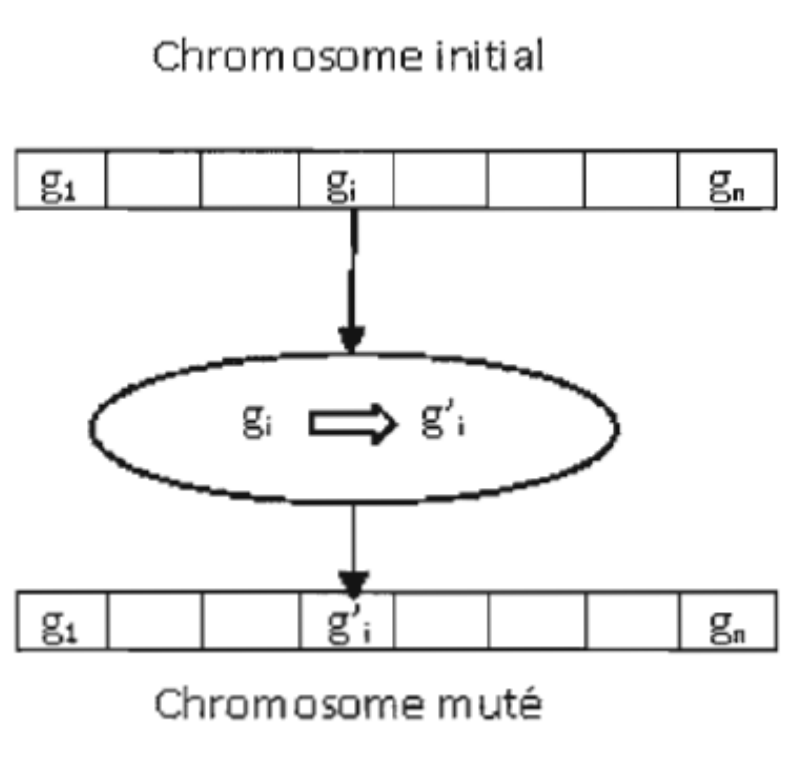


Figure III.5 : croisement a deux points [48].

#### 4.4.3. La mutation

Le rôle de cet opérateur est de modifier aléatoirement, avec une certaine probabilité, la valeur d'un composant de l'individu [49]. La figure III.6 présente un exemple de mutation de chromosome tel que le gène est retiré aléatoirement et est remplacé par la gène  $g'_i$ .

La mutation est un phénomène rare mais permet d'explorer de nouvelles zones dans l'espace de recherche et aide l'algorithme génétique à possiblement aller vers une solution optimale globale, sans rester pris dans une solution optimale locale.



*Figure III.6 : Illustration du principe de la mutation.*

## **5. Conclusion**

Les algorithmes génétiques fournissent des solutions proches de la solution optimale à l'aide des mécanismes de sélection, d'hybridation et de mutation. Ils sont applicables à de nombreux problèmes, dont le problème du voyageur de commerce par exemple.

# Chapitre IV

## La conception du système

*«If at first you do not succeed, try, try again. »*

*W. E. Hickson*

## 1. Introduction

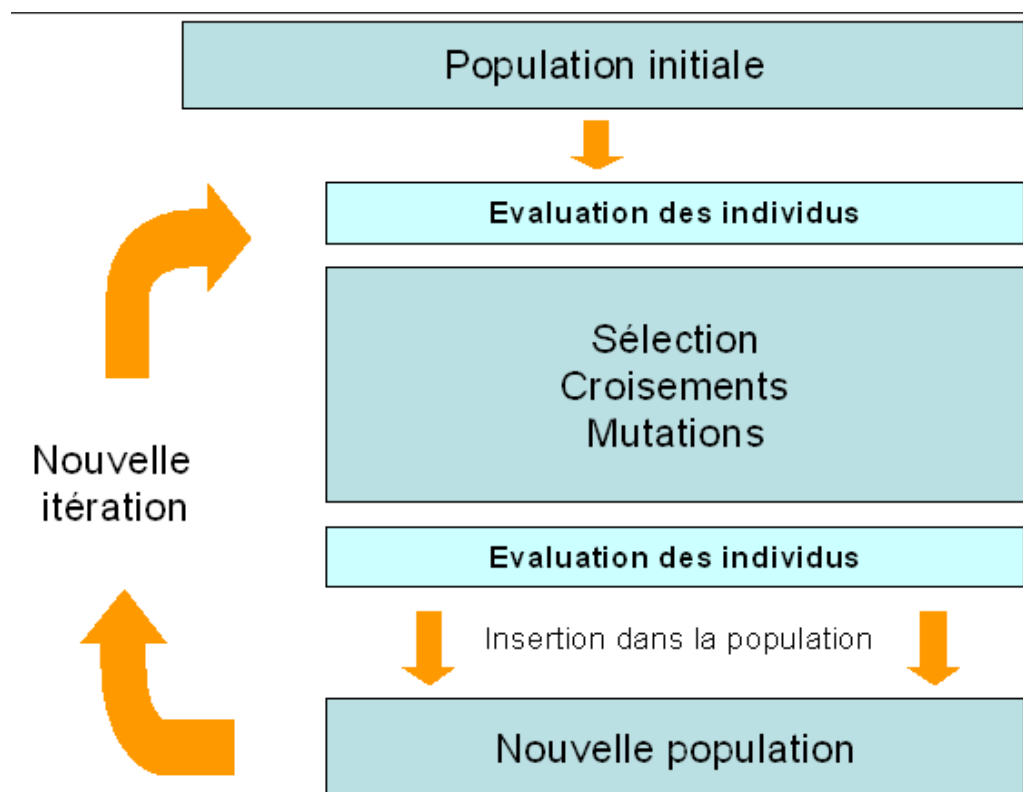
Les requêtes orientées services permettent aux utilisateurs d'accéder à plusieurs services Web de manière transparente et efficace. En outre, comme les services Web avec des fonctionnalités similaires devraient être fournis par des fournisseurs concurrents, un défi majeur est de concevoir des stratégies d'optimisation pour trouver les meilleurs services Web ou de la composition de ceux-ci à l'égard de la qualité attendue fournie par l'utilisateur (au niveau par exemple du temps, du coût, et de la réputation). Les normes existantes à base de technologies de découverte de services sont nettement insuffisantes pour la construction d'une infrastructure de service à part entière de requête, le paradigme actuel de recherche par mots clés ne peut pas toujours localiser précisément les services Web en partie à cause de la sémantique riche énoncée dans ces services. Le traitement des requêtes sur les services Web est un nouveau concept qui va au-delà de la vision traditionnelle centrée sur les données du traitement des requêtes, qui est principalement axé sur la performance. Dans ce chapitre nous commençons par la présentation des différentes étapes de la solution. Ensuite, nous introduisons notre fonction objective formulant la grandeur à optimiser et qui correspond à la meilleure sélection. Nous définissons aussi nos algorithmes génétiques (croisement et mutation). A la fin de chapitre nous présentons l'essentiel des diagrammes UML, à savoir les diagrammes respectivement des cas d'utilisation, de séquences et de classes.

## 2. Les algorithmes génétiques

Les algorithmes génétiques ont été développés à partir des années 1950 par John Holland et ses collègues à l'Université du Michigan. Holland était un chercheur en sciences de l'information et en systèmes adaptatifs, et il a utilisé des concepts issus de la biologie de l'évolution pour concevoir des méthodes d'optimisation et de résolution de problèmes.

L'inspiration principale pour les algorithmes génétiques provient de la théorie de l'évolution de Darwin. Holland s'est intéressé à l'idée d'utiliser des mécanismes évolutifs pour résoudre des problèmes complexes. Il a introduit des concepts tels que la représentation des solutions sous forme de chaînes de bits (chromosomes), les opérations génétiques (croisement et mutation) et la sélection naturelle pour guider l'évolution des solutions vers des solutions de meilleure qualité.

Il convient de noter que les algorithmes génétiques ne sont qu'une des nombreuses techniques d'optimisation et de recherche opérationnelle disponibles en informatique. Ils sont particulièrement adaptés aux problèmes difficiles à résoudre de manière déterministe ou analytique, et qui nécessitent une exploration approfondie de l'espace des solutions possibles.



*Figure IV.1 : Schéma général d'un AG.*

## 2.1. Expression de la fonction objective

En considérant les critères cités ci-dessus, la fonction objective  $f(x)$  est définie comme une fonction des valeurs des attributs de la qualité de service QoS. Les valeurs de cette fonction correspondent aux différentes sélections qui peuvent être effectuées pour classer et composer la solution globale, en tenant compte des contraintes globales et des besoins des utilisateurs. A titre d'exemple et en considérant une solution  $x = S$  et les critères QoS, les contraintes peuvent être :

- **Coût (S)  $\leq$  2500 dinars** : Le coût global d'une solution doit être inférieur ou égal à 2500 dinars.
- **Latence (S)  $\leq$  1000ms** : La latence globale doit être inférieure ou égale à 1000ms.
- **Réputation (S)  $\geq$  50 %** : La réputation globale doit être supérieure ou égale à 50 %.
- **Disponibilité (S)  $\geq$  50 %** : La disponibilité globale doit être supérieure ou égale à 50 %.
- **Fiabilité (S)  $\geq$  50 %** : La fiabilité globale doit être supérieure ou égale à 50 %.

Les valeurs des critères doivent optimiser une fonction globale qui fonctionne en maximisant les critères positifs (Réputation, Disponibilité et Fiabilité) et en minimisant les critères négatifs (Coût et Latence). Dans ce travail, et contrairement à d'autres travaux qui utilisaient d'autres fonctions objectives, nous définissons une nouvelle fonction qui exprime le score d'une solution, et permet ainsi de classer les différentes solutions avec attention, et de rechercher la solution.

La fonction objective  $f$  pourrait s'exprimer indifféremment comme :

$$Max(f) = \sum_{i=1}^N \frac{\sum_{k \in Pos} Log(C_i^k)}{\sum_{k \in Neg} Log(C_i^k)}$$

Ou :

$$Min(g) = \frac{1}{Max(f) = \sum_{i=1}^N \frac{\sum_{k \in Pos} Log(C_i^k)}{\sum_{k \in Neg} Log(C_i^k)}}$$

Où

- ✓ **k ∈ Pos** est l'ensemble des critères positifs (Réputation, Disponibilité, Fiabilité).
- ✓ **k ∈ Neg** est l'ensemble des critères négatifs (Coût, Latence).
- ✓ **Pos, Neg** sont respectivement les ensembles de critères positifs et négatifs.

D'après l'expression de f, elle est directement proportionnelle aux variables de l'ensemble Pos, et inversement proportionnelle aux variables de l'ensemble Neg. Nous avons introduit le logarithme des critères afin que les produits ne dépassent pas la taille de représentation.

L'algorithme qui illustre le croisement est le suivant

– **Déclaration**

soit X,Y,K,L :des chromosomes ;

soit r :la propabilité de croisement.

---

**Algorithm 1** croisement

---

**Début**

**croisement( X, Y, r)**

**Génération()** // générer une variable aléatoire m ;

**si** ( m < r ) **alors**

**pour** (i=1 ; i ≤ (taille.X) / 2) ; i++) **faire**

    X1[i] ← X[i] ;

    Y1[i] ← Y[i] ;

**fin pour**

**pour** (i=(taille.X) / 2) ; i ≤ (taille.X) ; i++) **faire**

    X1[i] ← Y[i] ;

    Y1[i] ← X[i] ;

**fin pour**

**finsi**

**sélectionner(K,L)** ; //sélectionner les deux meilleurs chromosomes entre les quatre.

**retourner(K,L)** ;

**Fin.**

---

L'algorithme qui illustre la mutation est le suivant :

– Déclaration

soit  $X$  : un chromosome;

soit  $r$  : la probabilité de mutation.

---

**Algorithm 2** mutation

---

Début

Mutation( $X,r$ )

choix() // choisir aléatoirement une position( $i$ );

Génération() // générer une variable aléatoire  $m$ ;

si ( $m < r$ ) alors

$X1[i] \leftarrow X$ ;

$X1[i] \leftarrow P$ ; //  $P$  Web service compatible choisi aléatoirement;

$FMeilleur \leftarrow (FMeilleur + X2)$ ;

finsi

Fin.

---

## 2.2. Données de la sélection des services web

### ➤ Description de la base

Nous avons considéré dans ce qui suit dix types de services web. Chaque service est proposé par un certain nombre de compagnies (fournisseurs de services). Aussi, chaque service offert est caractérisé par un certain nombre de critères de qualité de service. Nous utilisons cinq paramètres pour évaluer la qualité des services : la latence, la fiabilité, la disponibilité, le coût et la réputation. Nous avons saisi manuellement toutes les valeurs afin d'assurer le maximum de combinaisons de ces critères. Un échantillon de notre base est illustré dans le tableau IV.1.

<b>Campanie</b>	<b>Service</b>	<b>Coût</b>	<b>Latence</b>	<b>Fiabilité</b>	<b>Disponibilité</b>	<b>Réputation</b>
<b>1</b>	<b>6</b>	<b>1000</b>	<b>400</b>	<b>70</b>	<b>50</b>	<b>50</b>
<b>1</b>	<b>3</b>	<b>950</b>	<b>300</b>	<b>90</b>	<b>70</b>	<b>70</b>
<b>1</b>	<b>2</b>	<b>2100</b>	<b>700</b>	<b>50</b>	<b>70</b>	<b>70</b>
<b>2</b>	<b>1</b>	<b>2400</b>	<b>400</b>	<b>60</b>	<b>80</b>	<b>80</b>
<b>2</b>	<b>5</b>	<b>900</b>	<b>900</b>	<b>70</b>	<b>90</b>	<b>90</b>
<b>2</b>	<b>7</b>	<b>2000</b>	<b>500</b>	<b>80</b>	<b>90</b>	<b>90</b>
<b>2</b>	<b>3</b>	<b>700</b>	<b>450</b>	<b>90</b>	<b>70</b>	<b>70</b>
<b>3</b>	<b>4</b>	<b>500</b>	<b>500</b>	<b>70</b>	<b>70</b>	<b>70</b>
<b>3</b>	<b>6</b>	<b>2100</b>	<b>950</b>	<b>90</b>	<b>50</b>	<b>50</b>
<b>3</b>	<b>9</b>	<b>800</b>	<b>1000</b>	<b>60</b>	<b>80</b>	<b>80</b>
<b>2</b>	<b>9</b>	<b>700</b>	<b>600</b>	<b>50</b>	<b>70</b>	<b>90</b>
<b>1</b>	<b>5</b>	<b>600</b>	<b>300</b>	<b>60</b>	<b>60</b>	<b>70</b>
<b>1</b>	<b>3</b>	<b>850</b>	<b>250</b>	<b>80</b>	<b>70</b>	<b>90</b>
<b>4</b>	<b>6</b>	<b>600</b>	<b>400</b>	<b>50</b>	<b>90</b>	<b>80</b>
<b>5</b>	<b>4</b>	<b>1600</b>	<b>650</b>	<b>70</b>	<b>90</b>	<b>70</b>
<b>5</b>	<b>1</b>	<b>1000</b>	<b>700</b>	<b>80</b>	<b>60</b>	<b>50</b>

*Tableau IV.1 : Un exemple de base de données.*

### 3. Diagrammes de conceptions

La conception d'un projet informatique quelconque est la phase essentielle, dans laquelle une étude préalable globale est menée. L'objectif principal de cette phase est d'analyser l'ensemble des besoins puis d'imaginer des contextes d'utilisation. Pour arriver à une bonne conception il est nécessaire de choisir la bonne méthode de conception, pour cela nous allons présenter la conception de notre projet sous forme de diagrammes UML.

Dans notre conception nous avons utilisé 3 types de diagrammes :

- ✓ Le diagramme de cas d'utilisation qui permet de représenter les deux acteurs de notre système et les tâches qui leurs sont associées.
- ✓ Le diagramme de classes, qui exprime les classes et les associations entre elles dans notre système.
- ✓ Le diagramme de séquences pour représenter les échanges de messages entre les acteurs et le système.

#### 3.1. La vue statique

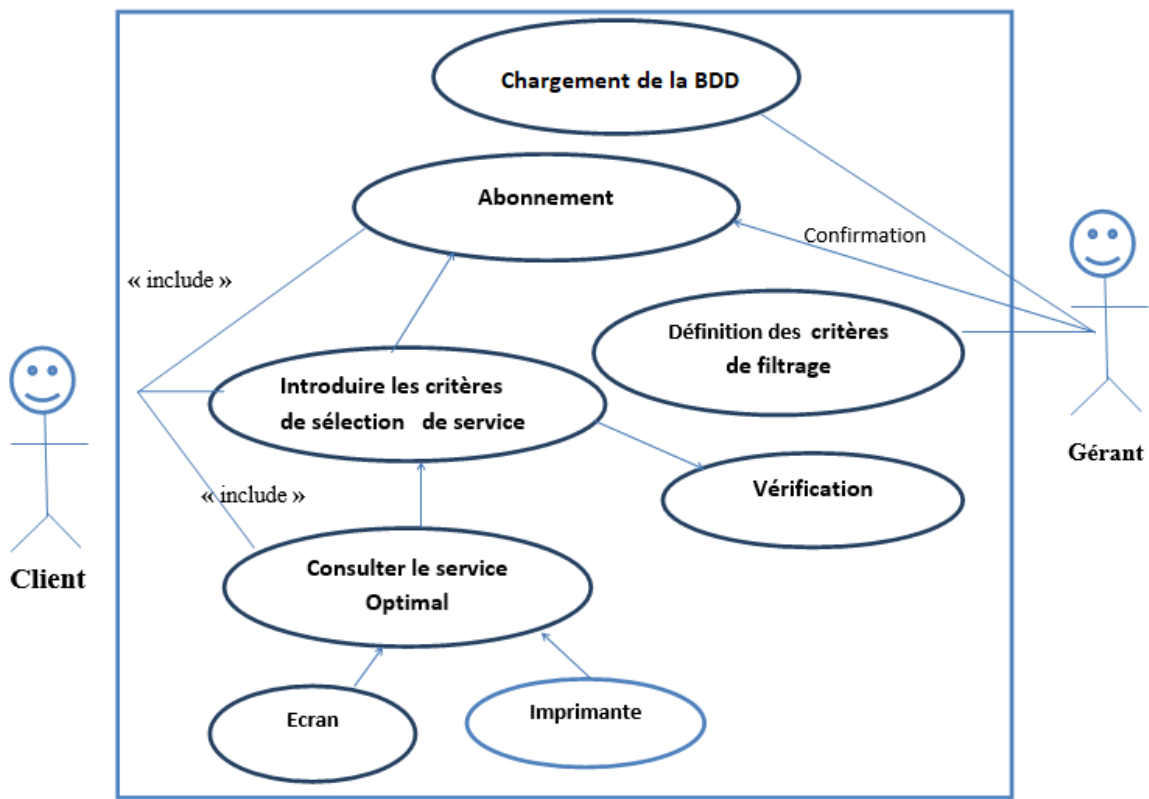
##### 3.1.1. Diagramme de cas d'utilisation

Un diagramme de cas d'utilisation permet de recueillir, d'analyser et d'organiser les besoins des utilisateurs, et de recenser les grandes fonctionnalités d'un système.

Acteur	Rôle
Client	<ul style="list-style-type: none"> <li>▪ Introduire les critères de sélection de service</li> <li>▪ Consulter le service optimal</li> </ul>
Gérant	<ul style="list-style-type: none"> <li>▪ Définition des critères de filtrage</li> <li>▪ Chargement de la BDD</li> </ul>

*Tableau IV.2 : Identification des acteurs et leurs rôles.*

La figure IV.2 représente le diagramme des cas d'utilisation de notre système

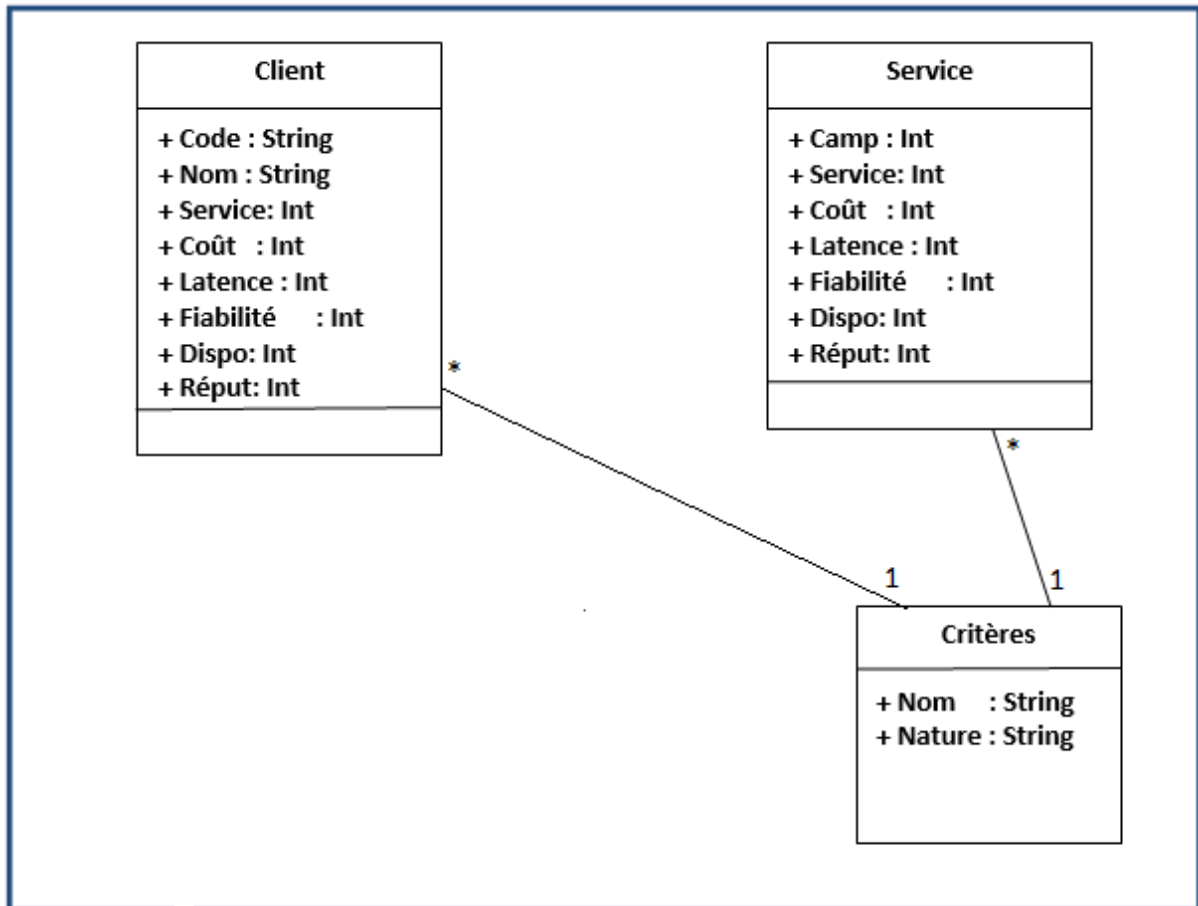


*Figure IV.2 : Diagramme de cas d'utilisation.*

### 3.1.2. Diagramme de classes

Un diagramme de classes permet de représenter les classes intervenant dans le système, notamment celles en rapport avec les données qui sont utilisées. Il constitue un élément très important de la modélisation et permet de définir quelles seront les composantes du système final. Il permet de structurer le travail de développement de manière très efficace.

La figure IV.3 représente le diagramme de classes de notre application.



*Figure IV.3 : Diagramme de classes.*

➤ **Description des classes :**

- **La classe "Service" :** Regroupe tous les critères de qualité de service offerts par les différentes compagnies.
- **La classe "Client" :** Regroupe tous les critères de qualité de service exigés par les Client.
- **La classe "Critère" :** Permet de définir la nature des critères de qualité de service
  - 1) Positive comme (La fiabilité, disponibilité, réputation).
  - 2) Négative comme (Le coût, la latence).

## 3.2. La vue dynamique

### 3.2.1. Diagramme de séquences

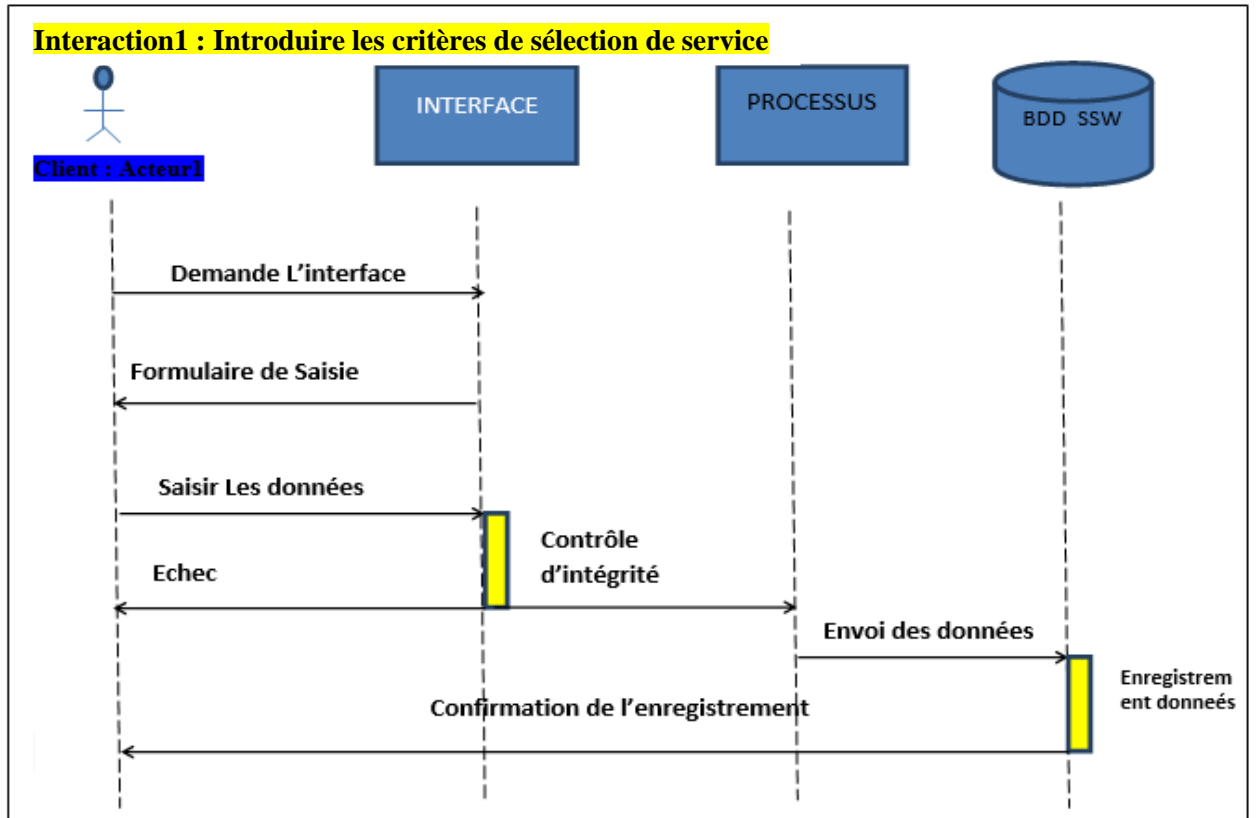
C'est un diagramme dynamique, il permet de représenter la dynamique d'un use cas ou la collaboration d'un ensemble d'objets internes au système. Il montre la séquence, représentation verticale chronologique, des messages passés entre blocs au sein d'une interaction.

➤ **Description des diagrammes de séquences de chaque cas**

Cas d'utilisation	Acteur principal	Pré condition
<b>Scénario 1</b> : Introduire les critères de Sélection de service	Client	<ul style="list-style-type: none"> <li>▪ Le client doit être abonné.</li> <li>▪ Le client atteint l'interface de saisie de données.</li> </ul>
<b>Scénario 2</b> : Définition de critères de filtrage	Gérant	<ul style="list-style-type: none"> <li>▪ Le gérant vérifie les critères de filtrage.</li> </ul>
<b>Scénario 3</b> : Chargement de la BDD	Gérant	<ul style="list-style-type: none"> <li>▪ Le gérant atteint l'interface de chargement de la BDD.</li> <li>▪ Chargement de données client.</li> </ul>
<b>Scénario 4</b> : Consulter le service optimal	Client	<ul style="list-style-type: none"> <li>▪ Le client doit être abonné.</li> <li>▪ Le client atteint l'interface de la meilleure solution.</li> </ul>

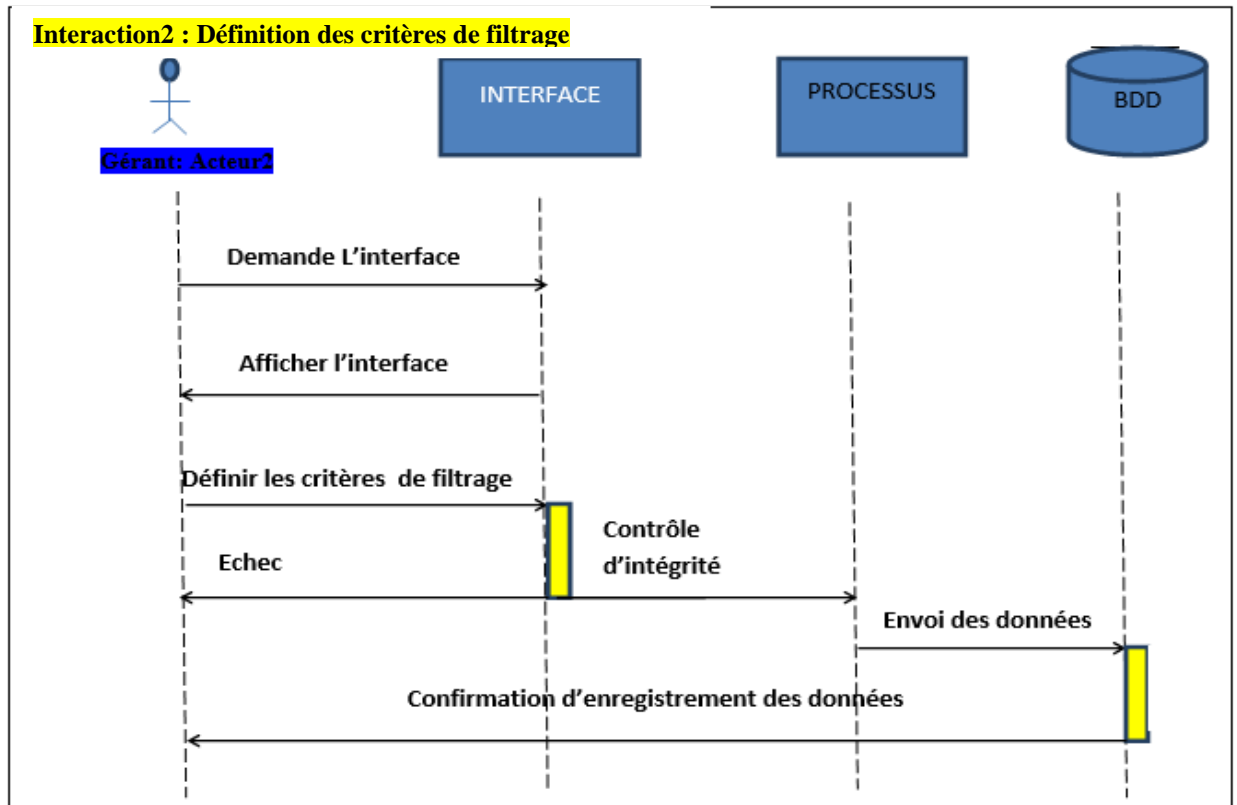
*Tableau IV.3 : Description de diagramme de séquences.*

- ✓ **Scénario 1 « Introduire les critères de sélection de service »** : Ce cas d'utilisation commence lorsque le client veut introduire ses critères de sélection du service optimal.



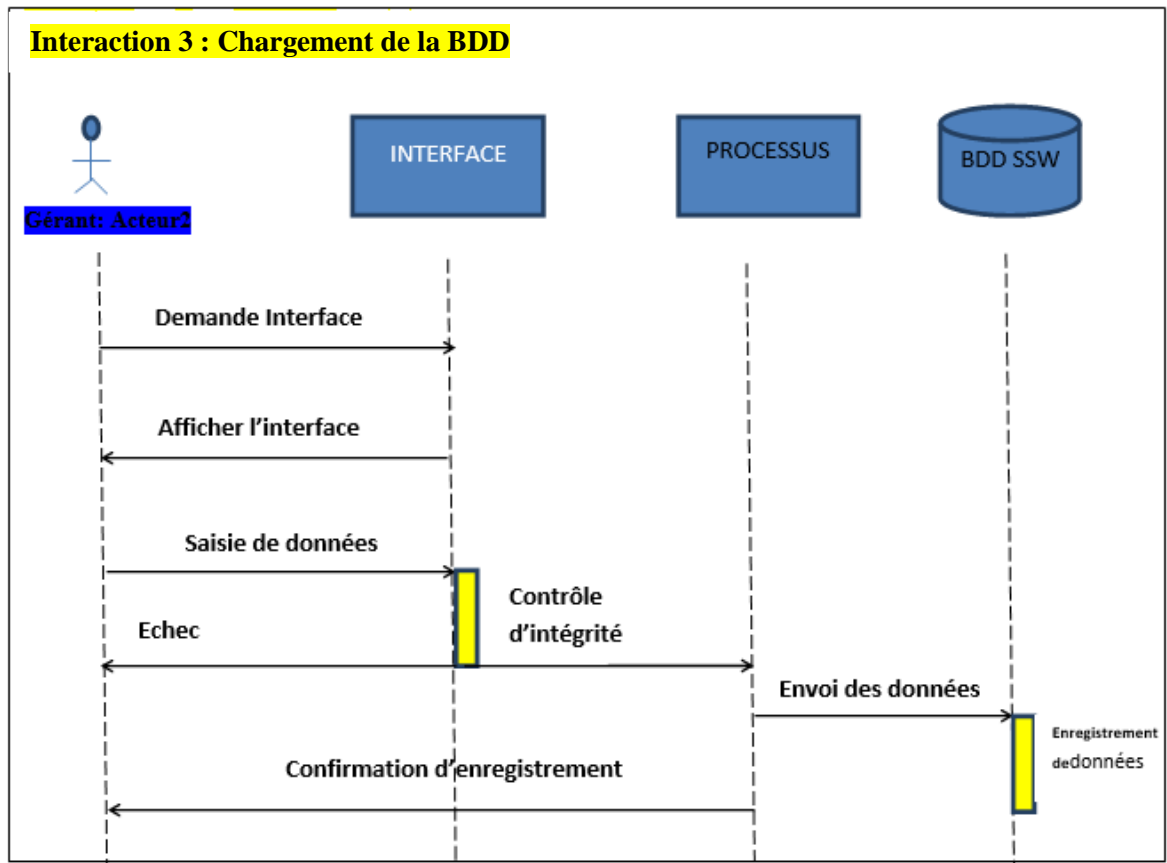
*Figure IV.4 : Diagramme de séquences « Introduire les critères de sélection de service ».*

- ✓ **Scénario 2 « Introduire les critères de filtrage »** : Ce cas commence lorsque le gérant veut définir les critères de filtrage.



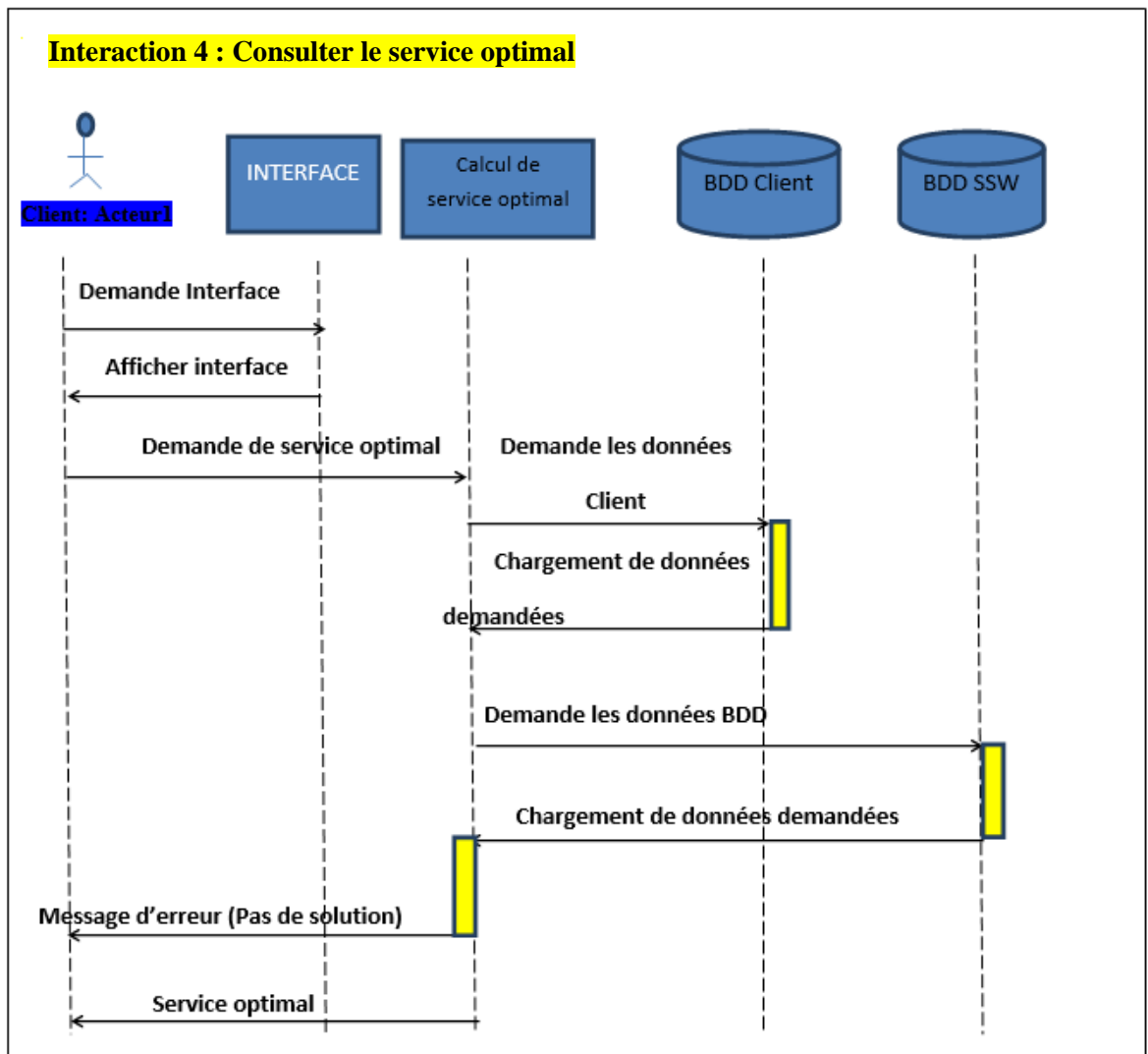
*Figure IV.5 : Diagramme de séquences « Définition des critères de filtrage ».*

- ✓ **Scénario 3 « Chargement de la BDD »** : Ce cas d'utilisation commence lorsque le gérant veut ajouter les données à la BDD.



*Figure IV.6 : Diagramme de séquences « Chargement de BDD».*

- ✓ **Scénario 4 « Consulter le service optimal »** : Ce cas d'utilisation commence lorsque le client veut consulter le service optimal selon les critères choisis.



*Figure IV.7 : Diagramme de séquences « Demande du calcul de la solution optimale ».*

## 4. Conclusion

Dans ce chapitre, nous avons présenté notre système en expliquant son fonctionnement et sa conception à l'aide du langage UML. Les diagrammes issus de cette conception nous permettent de maîtriser et comprendre toutes les vues de notre système, ce qui nous permettra par la suite de l'implémenter facilement.

Dans le dernier chapitre de ce mémoire, on entamera le côté pratique de notre travail et qui consiste à la mise en œuvre d'une application **de sélection de services web** en utilisant le langage de programmation Delphi.

# Chapitre V

## Implémentation et Test

*« If you want something done right, do it yourself. »*

*Charles-Guillaume Étienne*

## 1. Introduction

Ce chapitre constitue la dernière étape conduisant à la réalisation de notre application. L'étape d'implémentation consiste à traduire le résultat obtenu lors de l'étape de conception en un programme ou logiciel informatique exécuté sur une machine en utilisant les outils de développement et programmation adaptés au problème à traiter.

Nous allons décrire les composants de l'environnement de travail, ainsi que les outils d'implémentation et on donnera quelques exemples d'interfaces qui ont été réalisées dans notre logiciel.

## 2. Environnement de développement

### 2.1. Environnement matériel

➤ **PC portable DELL :**

- processeur : Intel i5
- mémoire vive : 4Go
- taille de l'écran : 12 pouces
- dalle : mat

### 2.2. Environnement logiciel (Langage de programmation) :

Le langage de programmation utilisé pour le développement de notre application est le Delphi, une des versions du langage Pascal sous Windows.

#### 2.2.1. Présentation du Delphi

Créé en 1995 par l'éditeur de logiciels Borland Software Corporation, le Delphi est d'abord un langage de programmation de haut niveau orienté objet, mais aussi un environnement de développement intégré (EDI) fonctionnant sous Windows. Une version spéciale pour Linux a vu le jour en 2001, toujours créée par Borland. C'est une interface qui aide les programmeurs dans leur développement de logiciels

exécutables. Généralement, un EDI comporte un éditeur de texte, un compilateur, un débogueur, un éditeur de liens...

Désormais, il existe des versions du Delphi pour presque tous les systèmes d'exploitation : Windows et Linux, mais aussi MacOS, iOS et Android. Le Delphi correspond aussi à un langage de programmation orienté objet hérité du langage Pascal [50].

En 2007 le Delphi a été revendu à la société Embarcadero

Cet environnement est particulièrement adapté pour :

- La programmation d'applications graphiques pour le mobile.
- La gestion d'une base de données.
- Le développement de logiciels d'entreprise.



*Figure V.1 : Logo de Delphi 7.*

### 2.2.2. Les avantages du Delphi

- **Rapidité de développement** : Le Delphi offre une productivité élevée grâce à son environnement de développement convivial, ses composants visuels et son langage de programmation intuitif. Cela permet de développer des applications plus rapidement.
- **Performance** : Les applications développées avec Delphi sont connues pour leur performance élevée, en particulier pour les applications Windows.

- **Accès aux fonctionnalités Windows** : Delphi offre un accès complet aux fonctionnalités du système d'exploitation Windows, ce qui permet de créer des applications puissantes et intégrées à la plateforme.
- **Héritage de Pascal** : Le langage de programmation utilisé par Delphi, dérivé de Pascal, est réputé pour sa lisibilité et sa facilité d'apprentissage, ce qui en fait un bon choix pour les débutants en programmation.
- **Support de la programmation orientée objet** : Delphi prend en charge la programmation orientée objet, ce qui permet de développer des applications modulaires, maintenables et évolutives.

### 2.2.3. Les inconvénients du Delphi

- **Plateforme spécifique** : Le Delphi est principalement utilisé pour le développement de logiciels Windows, ce qui limite sa portabilité vers d'autres plateformes.
- **Coût élevé** : Les licences Delphi peuvent être coûteuses, ce qui peut être un obstacle pour les petites entreprises et les développeurs indépendants.
- **Communauté restreinte** : La communauté de développeurs Delphi est plus petite par rapport à d'autres langages de programmation populaires, ce qui peut rendre plus difficile la recherche de ressources et d'aide en cas de besoin.
- **Support pour les technologies émergentes** : Le Delphi peut être en retard dans le support des dernières technologies et des frameworks émergents, ce qui peut limiter les options de développement avancées.

### 2.2.4. Ce que distingue le Delphi des autres langages

Ce qui distingue le Delphi des autres langages de programmation est son intégration étroite avec la plateforme Windows et son environnement de développement convivial. Le Delphi offre un accès direct aux fonctionnalités du système d'exploitation Windows, ce qui permet de créer des applications puissantes et intégrées. De plus, son langage de programmation Pascal, réputé pour sa

lisibilité, facilite le développement de code clair et compréhensible. Avec son IDE riche en fonctionnalités, le Delphi offre une productivité élevée, permettant aux développeurs de créer des applications rapidement et efficacement.

### 2.2.5. La version du Delphi

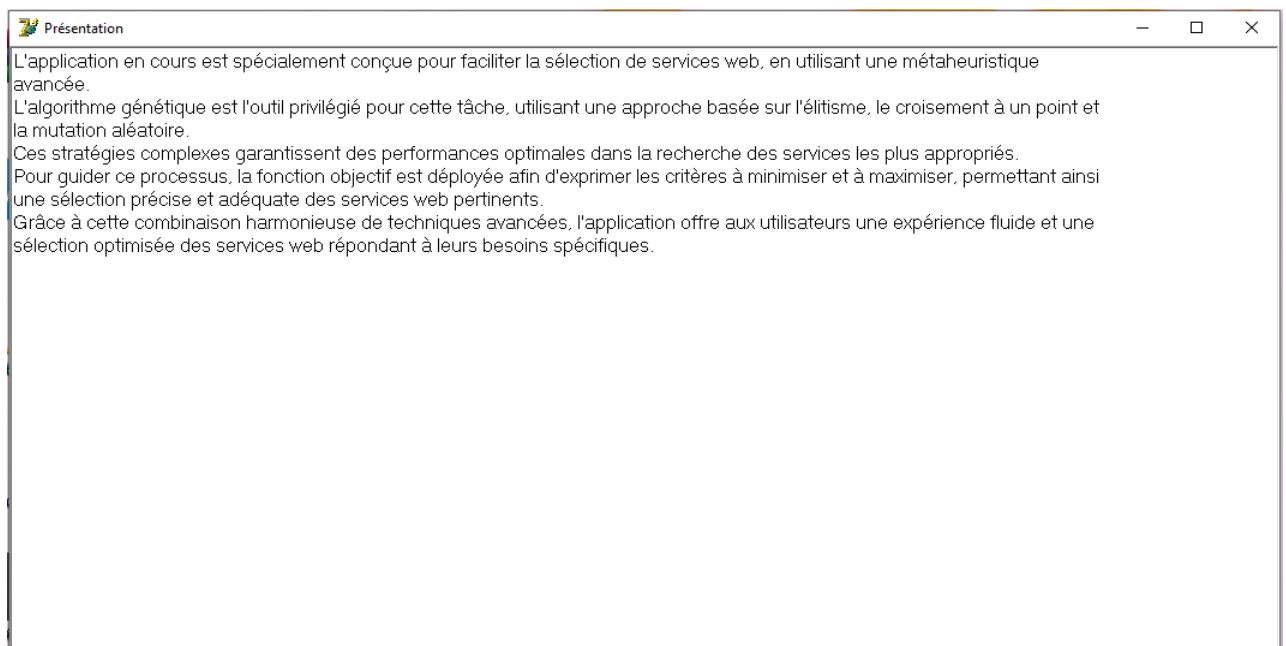
La version du Delphi a été utilisée est Delphi 7.0

## 3. Présentation de l'interface de l'application

Dans cette partie nous allons présenter le mode de fonctionnement et les différentes fenêtres de notre application.

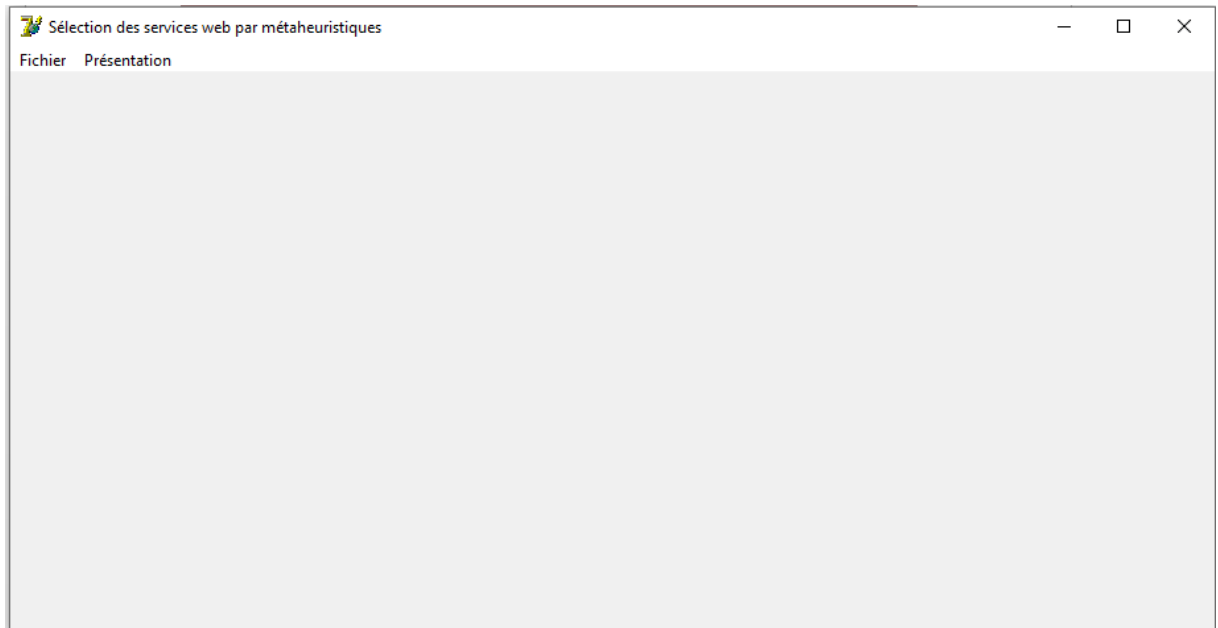
Commençant par la page d'accueil de l'interface, cette première fenêtre

« **Présentation** » donne une idée générale sur l'objectif de notre application.



*Figure V.2 : Page d'accueil.*

La deuxième fenêtre présente la fenêtre principale de notre application. Elle est constituée de deux onglets **Fichier** et **Présentation**.

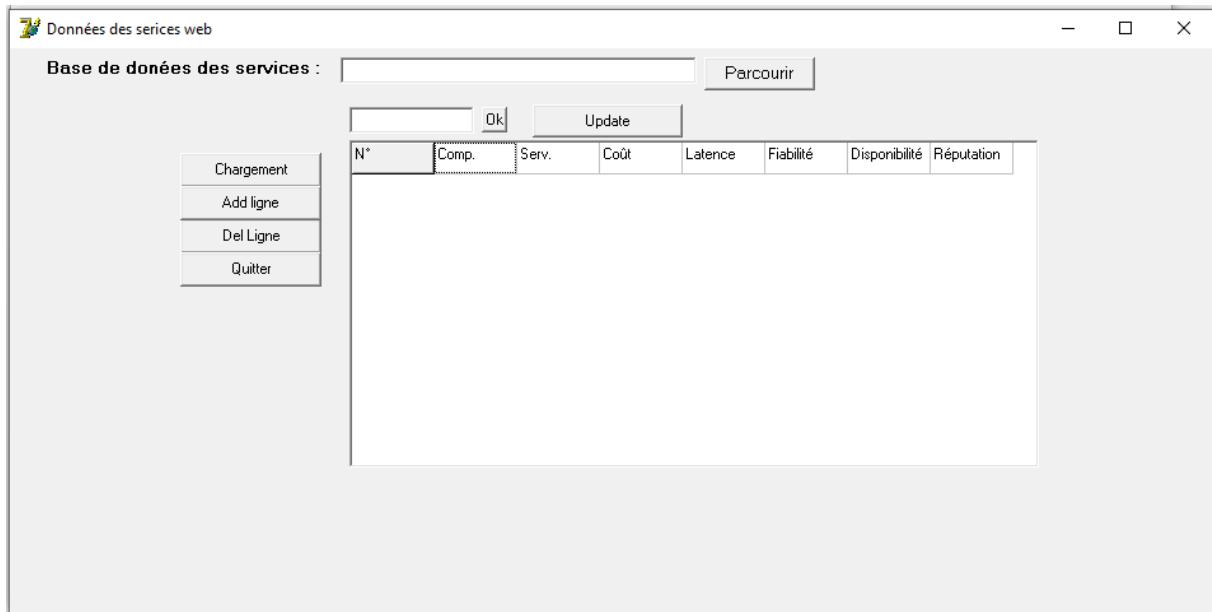


*Figure V.3 : Fenêtre principale du système.*

A partir du menu **Fichier** on peut accéder à quatre sous menu de notre application

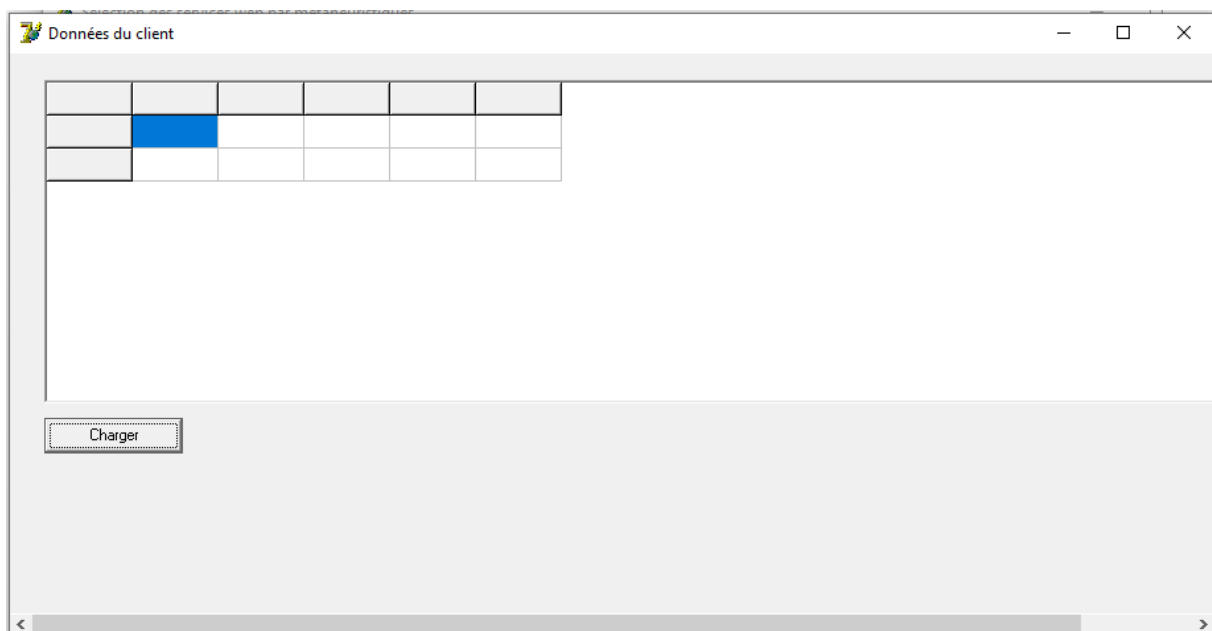
- ❖ **Données des services**
- ❖ **Données du client**
- ❖ **Services sélectionnés**
- ❖ **Sortir**

En cliquant sur le bouton « **Données des services** » le gérant peut accéder à la BDD de sélection de services web, où il peut charger et faire la mise à jour de la BDD (Figure V.4)



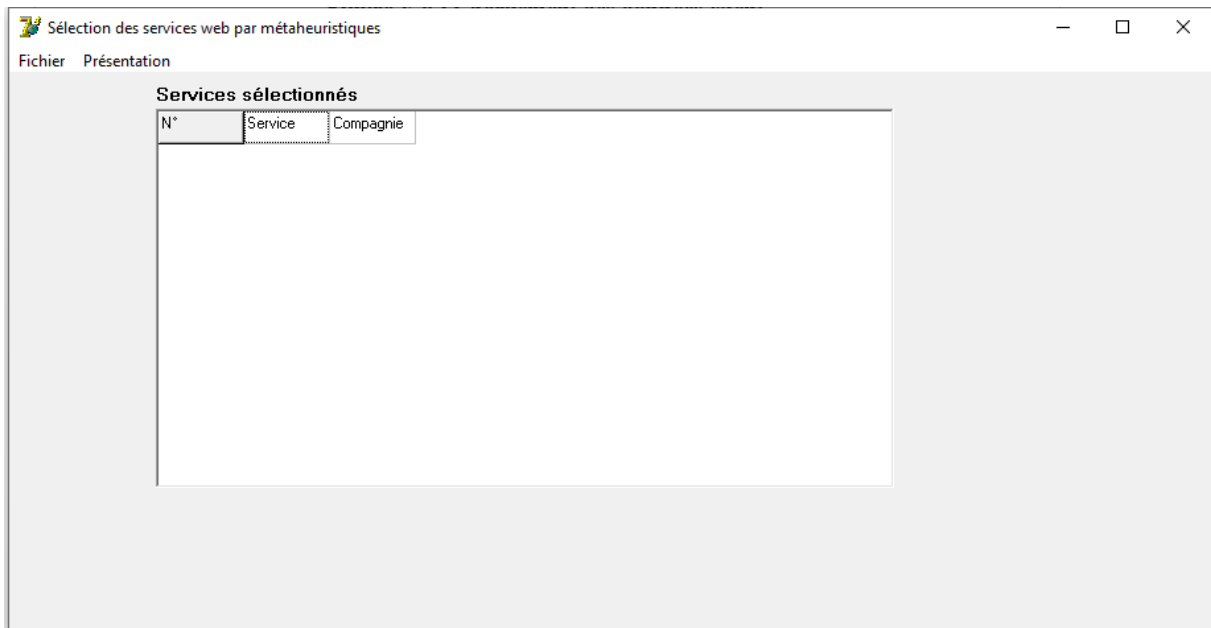
*Figure V.4 : Interface Mise à jour BDD.*

Pour accéder à l'espace client on doit cliquer sur le bouton « **Données du client** » Cet espace permet de sélectionner le fichier **C.txt** celui-ci est enregistré dans notre base de données **Client**. Le client va avoir une sélection optimale de services web en fonction de la BDD des services web. (Figure V.5).



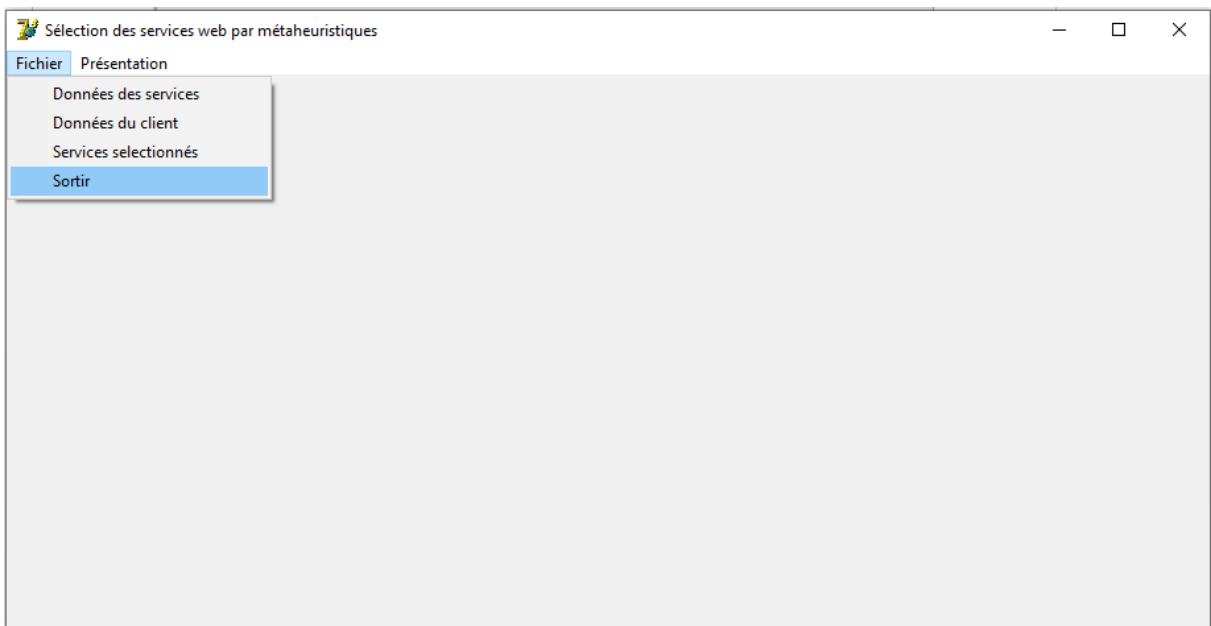
*Figure V.5 : Chargement des données client.*

Et pour calculer et afficher le service web optimal qui répond aux exigences de notre client avec notre Meta-heuristique « **Algorithme Génétique** » on doit cliquer sur le bouton « **Services sélectionnés** » Ceci consiste à afficher le **numéro de la compagnie** et le **numéro du service** qui assure le service le plus proche aux critères introduits le client. (Figure V.6).



*Figure V.6 : Interface Résultats de la sélection de services web.*

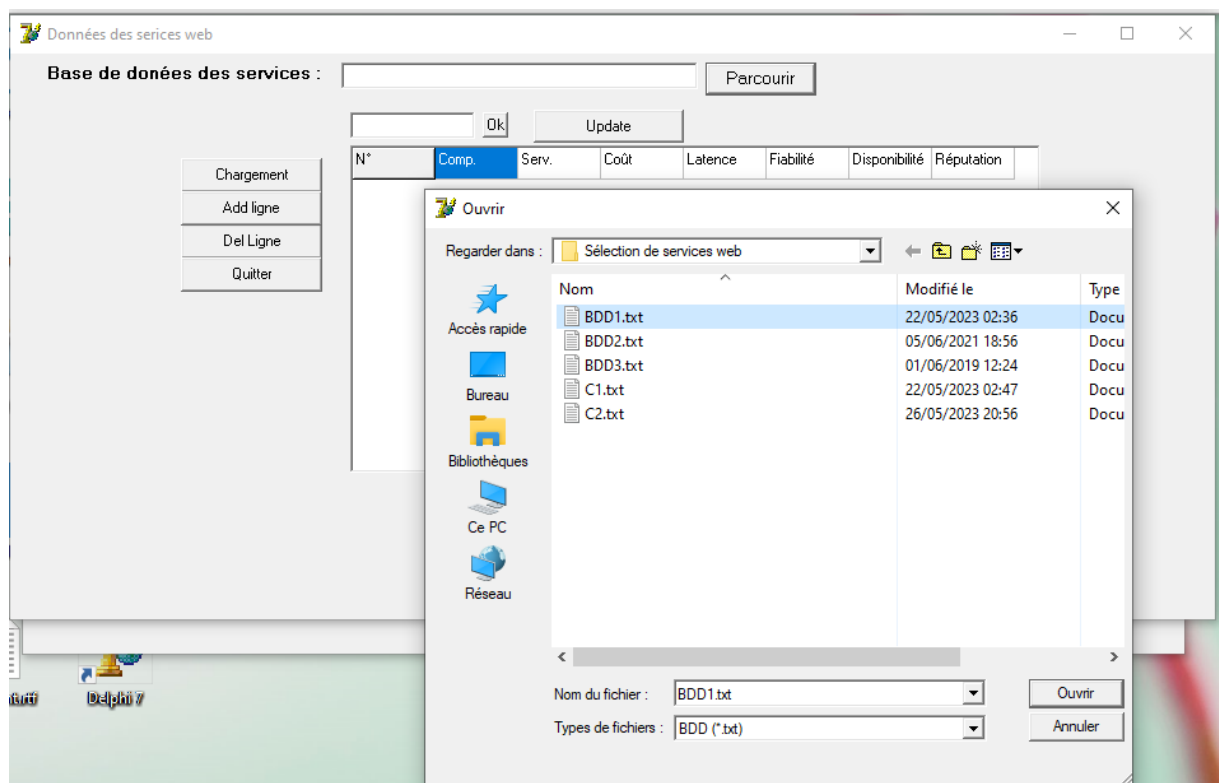
Et à la fin et pour quitter notre application on doit cliquer sur le bouton « **Sortir** ».



*Figure V.7 : Sortir de l'application.*

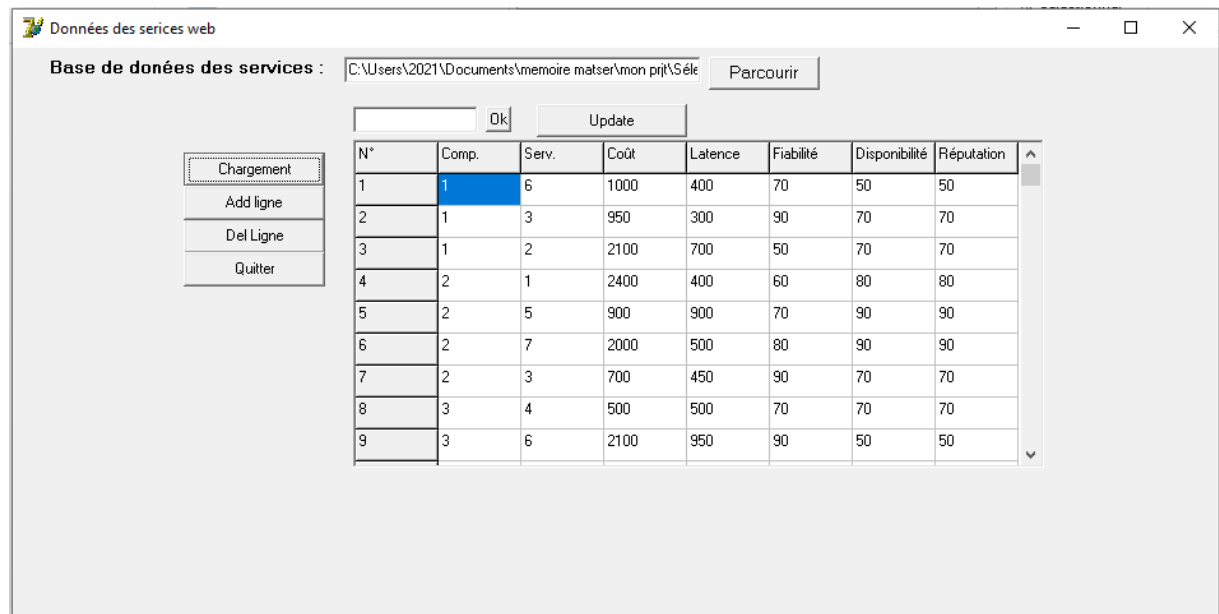
## 4. Expérimentation

Nous proposons une configuration qui se base sur la sélection de trois services, chaque service est caractérisé par les cinq contraintes de QoS. Ces contraintes doivent être vérifiées durant la sélection des services web, en respectant ces dernières nous devons ne pas dépasser une certaine valeur de coût et latence, et nous devons garantir un maximum de disponibilité, fiabilité et réputation de ces services sélectionnés.



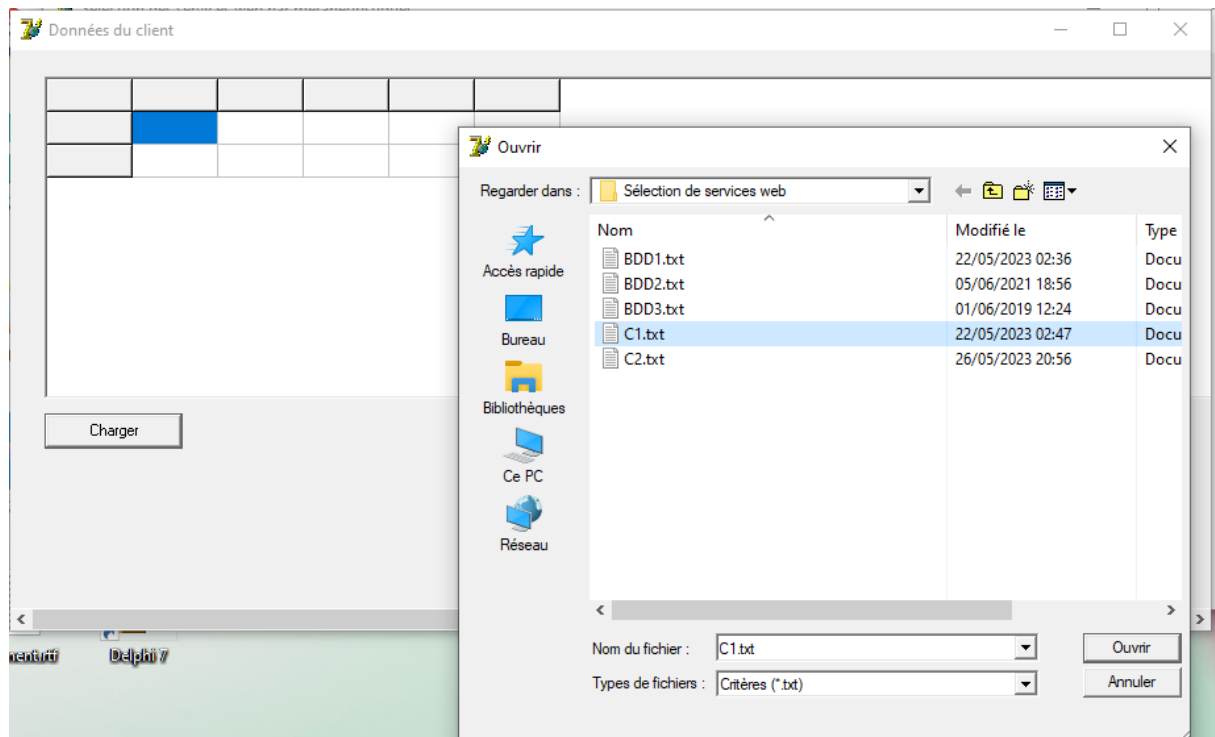
*Figure V.8 : Choix d'une base de données.*

Nous avons introduit les valeurs des critères proposés par plusieurs compagnies et nous avons chargé la BDD (figure V.9).

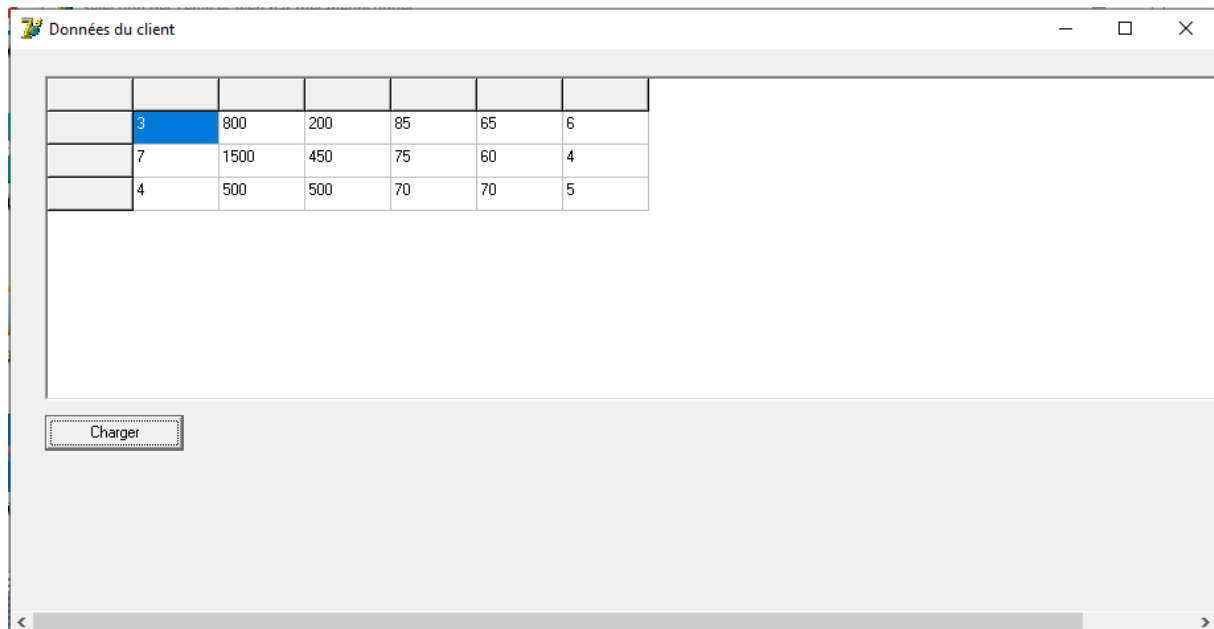


*Figure V.9 : Chargement de la base de données.*

La figure (V.10) et la figure (V.11) représentent le chargement des valeurs des critères des trois services exigés par le client.



*Figure V.10 : Choix du fichier C.txt.*

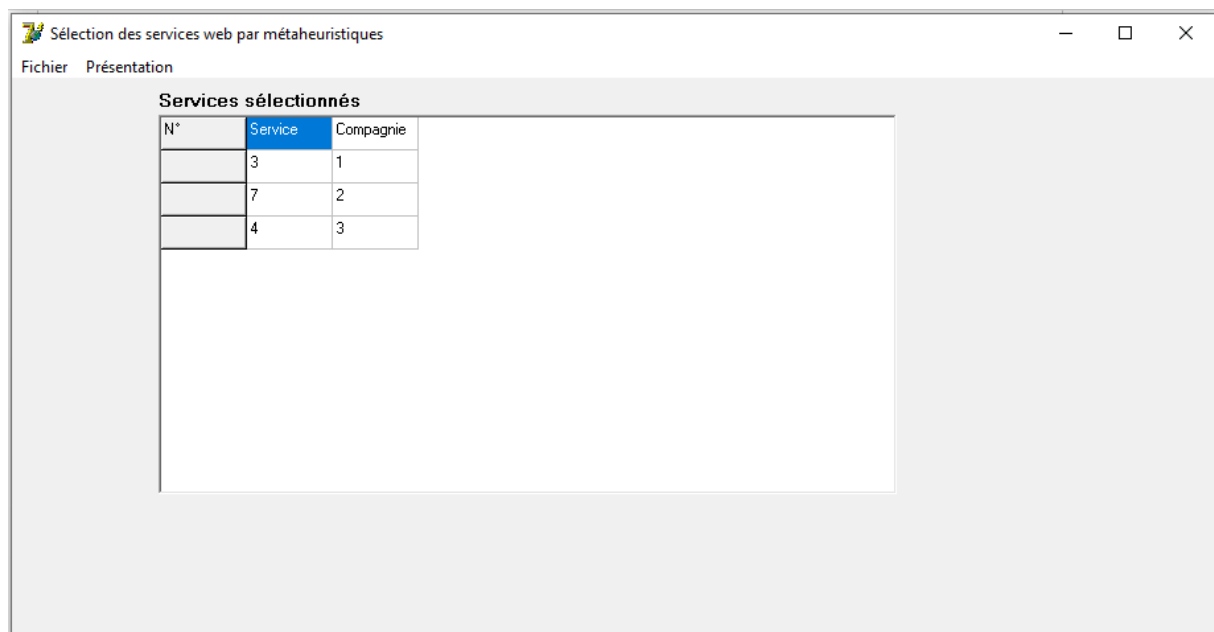


	3	800	200	85	65	6
	7	1500	450	75	60	4
	4	500	500	70	70	5

Charger

*Figure V.11 : chargement des données client.*

La (figure V.12) représente le résultat de la sélection du service web optimal calculé via l'exécution de l'algorithme génétique.



Sélection des services web par métaheuristiques

Fichier Présentation

**Services sélectionnés**

N°	Service	Compagnie
3		1
7		2
4		3

*Figure V.12 : Résultat de la sélection.*

## **5. Conclusion**

Dans ce chapitre nous avons présenté les résultats de la validation de notre approche, afin d'évaluer l'efficacité de ce dernier. Les résultats obtenus sont très acceptables et montrent qu'ils satisfassent les attentes de l'utilisateur. A partir des résultats expérimentaux obtenus précédemment.

D'après les résultats qu'on a obtenus, on a confirmé l'efficacité des algorithmes génétiques.

# Conclusion générale

## Conclusion générale

Les services Web sont des technologies émergentes et prometteuses pour le développement, le déploiement et l'intégration d'applications sur 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 besoin d'approches efficaces pour la sélection des services.

La sélection des services web est l'une des problématiques les plus importantes de l'architecture orientée service. Elle constitue aussi l'une des étapes les plus importantes dans le processus de composition. Au cours de ce mémoire, nous nous sommes intéressés au problème de sélection des services web sur la base des besoins non fonctionnels (QoS).

Dans ce travail, nous avons traité le problème de sélection de Web services selon leurs caractéristiques de qualité de service. Nous avons présenté dans ce mémoire les technologies liées aux Web services, aussi un état de l'art sur l'optimisation combinatoire multi objectif pour passer ensuite aux méthodes proposées dans la littérature pour résoudre le problème de sélection des Web services.

Notre prototype sélectionne les compositions de services les plus satisfaisantes, en se basant sur 05 critères de QOS : Latence, fiabilité, disponibilité, coût, réputation. La composition concrète recherchée doit maximiser un ensemble de ces critères positifs et minimiser un ensemble d'autres critères négatifs, en plus elle doit satisfaire un groupe de contraintes globales. Pour améliorer les résultats de sélection, nous avons défini une nouvelle fonction objective, exprimée par des termes logarithmiques simplifiant son calcul.

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

- Le recuit simulé.
- La programmation par contrainte.
- La programmation dynamique.
- ...

## Bibliographies

- [1]: **Bussler C** “B2B Integration: Concepts and Architecture“; Springer-Verlag Berlin, 2003.
- [2]: The Accredited Standards Committee (ASC) X12; Disponible sur: <http://www.x12.org/>
- [3] : Rosetta Net ; Disponible à : <http://www.rosettanet.org/>
- [4] :[https://www.ibm.com/support/knowledgecenter/SSGMCP\\_5.2.0/com.ibm.ci.cs.ts.webservices.doc/concepts/dfhws\\_definition.html](https://www.ibm.com/support/knowledgecenter/SSGMCP_5.2.0/com.ibm.ci.cs.ts.webservices.doc/concepts/dfhws_definition.html) (dernière consultation 22 mars 2020)
- [5] : <https://www.w3.org/TR/ws-arch/> (dernière consultation 22 mars 2020)
- [6]:[https://www.tutorialspoint.com/webservices/web\\_services\\_characteristics.htm](https://www.tutorialspoint.com/webservices/web_services_characteristics.htm) (dernière consultation 23 mars 2020)
- [7] : **Valette M.**, Web services Communication inter langage, Version 2.0, Ecole supérieur d’Informatique de Paris, 8 mars 2006
- [8] : <https://www.guru99.com/web-service.html> (dernière consultation 23 mars 2020)
- [9]:[https://www.researchgate.net/figure/Web-Services-architecture\\_fig2\\_228781095](https://www.researchgate.net/figure/Web-Services-architecture_fig2_228781095) (dernière consultation 23 mars 2020)
- [10] : <https://www.w3schools.com/>(dernière consultation 23 mars 2020)
- [11]:[https://www.irit.fr/journali3/hors\\_serie/annee2004/revue\\_i3\\_hs2004\\_01\\_07.pdf](https://www.irit.fr/journali3/hors_serie/annee2004/revue_i3_hs2004_01_07.pdf) (dernière consultation 23 mars 2020)
- [12]: J-M. Chauvet, Services Web avec SOAP, WSDL, UDDI, ebXML..., Edition EYROLLES
- [13] : N. Mitra et Y. Lafon, SOAP Version 1.2 Part 0 : Primer (Second Edition 2003).
- [14] : Tableau tire et traduit par [M. Leblanc, HEC Montréal, 14 novembre 2002] de Eric Newcomer, Understanding Web Services : XML, WSDL, SOAP and UDDI. Éd.Addison-Wesley, 2002 p.25
- [15] : A. Vezain, Les service web - présentation générale, rapport technique, Association HERMES, Février 2005.

- [16] : M. Vialette, Web Services Communication inter langage, Version 2.0 Ecole Supérieur d'Informatique de Paris, 8 mars 2006.
- [17] : <http://openclassrooms.com/courses/les-services-web>
- [18]:[https://www.researchgate.net/publication/260870411\\_A\\_Genetic\\_Algorithm\\_approach\\_to\\_QoS-based\\_Semantic\\_Web\\_service\\_Composition](https://www.researchgate.net/publication/260870411_A_Genetic_Algorithm_approach_to_QoS-based_Semantic_Web_service_Composition)(dernière consultation 28 mars 2020)
- [19]: Clement, A. H. (2004). UDDI v.3.0.2. OASIS Specification.
- [20]: D. Roman, U. K. (2005). "Web Service Modeling Ontology,"
- [21] : <https://www.exoco-lmd.com/technologies-et-services-web/le-controle-de-qos-pour-les-services-web/> (dernière consultation 29 mars 2020)
- [22]: E. Alrifai, T. Risse Combining Global Optimization with local selection for Efficient QoS-aware Service Composition in WWW09, April 20-24, 2009, Madrid, Spain.
- [23]: E. Alrifai, T. Risse Selecting Skyline Services for QoS-based Web Service Composition in Proceedings of the WWW 2010, April 26-30, 2010, Raleigh, North Carolina, USA.
- [24]: Zeng, L. B. (2004). QoS-Aware Middleware for Web Services Composition. . IEEE transactions.
- [25] : BOUDJELABA Hakim. Sélection des services Web sémantiques Université A / Mira de Béjaia, juin 2012(dernière consultation 27 mars 2020)
- [26] : <https://eprints.qut.edu.au/377/1/377.pdf> (dernière consultation 31 mars 2020)
- [27] : Hazar TURKI KOSENTINI, Dr. Leila BACCOUCHE, Pr. Henda HAJJAMI BEN GHEZALA. Etude de cas pour la selection des services web basée sur les contraintes temporelles (dernière consultation 31 mars 2020)
- [28] : La sélection des services web dans une composition à base de critères non fonctionnels. Thèse de doctorat Tlemcen.
- [29] : Yu, T. a.-J. (2005a). Service selection algorithms for composing complex services with multiple qos constraints. . International Conference.
- [30]: Liu, M. M. (2012). A quality of service (qos)-aware execution plan selection approach for a service composition.
- [31]: Yu, T. a.-J. (2005b). Service selection algorithms for web services with end-to-end qos constraints. Information systems and e-business.
- [32]: Rodriguez-Mier, P. M. (2010). Composition of web services through genetic programming.

- [33]: Moustafa, A. a. (2013). Multi-objective service composition using reinforcement learning. In International Conference on Service-Oriented Computing.
- [34]: Feng, L.-i. O. (2013). Qos optimization for web services composition based on reinforcement. International Journal of Innovative Computing, Information.
- [35]: Li, Y.-Q. a. (2012). An approach of qos-guaranteed web service composition based on a winwin strategy. 19th International Conference on.
- [36]: Lecue. M. (2009). Towards scalability of quality driven semantic web service composition. International Conference, (pp. pages 469–476).
- [37]: Klein, A. I. (2011). Efficient heuristic approach with improved time complexity for qos-aware service composition. IEEE International Conference, (pp. pages 436–443).
- [38]: Ko, J. M.-H. (2008). Quality-of-service oriented web service composition algorithm and planning architecture. journal of Systems and Software .
- [39]: Rosenberg, F. M. (2010). Metaheuristic optimization of larges-scale qos-aware service compositions. Services computing, (pp. pages 97–104.).
- [40]: Hadjila, F. a. (2012). Qos-aware service selection. Alger. Algerie.
- [41]: Wang, J. a. (2008). Optimal web service selection. Based on multi-objective genetic algorithm. (pp. 553–556 IEEE).
- [42]: Ming, C. a.-w. (2007). an approach for web services composition based on qos and discrete particle swarm optimization. Eighth ACIS International Conference on Software Engineering, Artificial Intelligence.
- [43]: Merzoug, M. C. (2014). Qos-aware web service selection based on harmony search. ISKOMaghreb, pages 1–6. IEEE.
- [44] : Souquet Amédée. Algorithmes génétiques. Master's thesis, Université de Nice, 2004.
- [45] : **Michel Van Caneghem**, « Algorithme et Complexité 4 », 2002
- [46] : Gisèle Legault. Un algorithme génétique pour la conception topologique de réseaux téléinformatiques à commutation de paquets.
- [47]: Eric Taillard Patrick Siarry Johann Dréo, Alain Petrowski. Métaheuristiques pour l'optimisation difficile. Eyrolles, Juillet 2003.
- [48] : Jean-Marc Alliot and Nicolas Durand. Algorithmes génétiques. Technico report, Laboratoire d'Optimisation Globale de ENAC, <http://www.recherche.enac.fr/opti/GA/FAG/ag.pdf>, Mars 2005.

- [49] : Sophie Voisin.1 Application des algorithmes génétiques à l'estimation de mouvement par modélisation markovienne. Technico report, Centre National de la Recherche Scientifique (CNRS), <http://imaging.utk.edu/people/svoisin/rapportDEA.pdf>, Juin 2004.
- [50] : Delphi : définition et présentation de ce langage informatique  
journaldunet.fr